

LNCS 6443

Kok Wai Wong
B. Sumudu U. Mendis
Abdesselam Bouzerdoum (Eds.)

Neural Information Processing

Theory and Algorithms

17th International Conference, ICONIP 2010
Sydney, Australia, November 2010
Proceedings, Part I

1
Part I

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 for Informatics, Saarbruecken, Germany

Kok Wai Wong
B. Sumudu U. Mendis
Abdesselam Bouzerdoum (Eds.)

Neural Information Processing

Theory and Algorithms

17th International Conference, ICONIP 2010
Sydney, Australia, November 22-25, 2010
Proceedings, Part I

Volume Editors

Kok Wai Wong
Murdoch University
Murdoch, WA, 6150, Australia
E-mail: k.wong@murdoch.edu.au

B. Sumudu U. Mendis
The Australian National University
Canberra, ACT 0200, Australia
E-mail: sumudu.mendis@anu.edu.au

Abdesselam Bouzerdoum
University of Wollongong
Wollongong, NSW 2522, Australia
E-mail: salim@elec.uow.edu.au

Library of Congress Control Number: 2009939833

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

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

ISSN 0302-9743
ISBN-10 3-642-17536-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-17536-7 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 2010
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

Welcome to the 17th International Conference on Neural Information Processing (ICONIP 2010) held in Sydney, 22–25 November 2010. In this volume you will find papers presented at this conference. ICONIP is the annual conference of the Asia Pacific Neural Network Assembly (APNNA, <http://www.apnna.net>). The aim of the Asia Pacific Neural Network Assembly is to promote the interaction of researchers, scientists, and industry professionals who are working in the neural network and related fields in the Asia Pacific region, primarily via the ICONIP conference. This year's theme was hybrid / human centred neural systems.

ICONIP 2010 received 470 excellent submissions. Of these, 146 regular session and 23 special session papers appear in these proceedings by Springer. Many outstanding papers do not appear here due to space limitations. Each paper was assessed by at least three reviewers. The conference will be followed by two associated workshops, the ICONIP International Workshop on Data Mining for Cybersecurity, held in November at the University of Auckland, New Zealand, and the ICONIP International Workshop on Bio-inspired Computing for Intelligent Environments and Logistic Systems, held in March at the Australian National University in Canberra, Australia.

I am very pleased to acknowledge the support of the conference Advisory Board, the APNNA Governing Board and Past Presidents, who gave their advice, assistance and promotion of ICONIP 2010. I gratefully acknowledge the technical sponsorship of the International Neural Network Society (INNS), the Japanese Neural Network Society (JNNS), the European Neural Network Society (ENNS), and the Australian Research Council Network in Human Communication Science (HCSNet).

A special thanks to Kevin Wong, Sumudu Mendis and Sukanya Manna without whom the conference organisation would have been much less smooth.

For the many reviewers who worked hard on giving thorough, tough but fair referee reports, thank you! Finally I would like to thank all the authors of papers, the speakers and panelists, volunteers and audience. With your support ICONIP 2010 will continue the tradition of being an uplifting, educational and enjoyable conference.

October 2010

Tom Gedeon

Organization

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)
ARC Network in Human Communication Science

Conference Committee

Honorary Chair	Shun-ichi Amari, Japan
General Chair	Tamás (Tom) Gedeon, Australia
Technical Program Chairs	Kok Wai (Kevin) Wong, Australia Tom Gedeon, Australia Salim Bouzerdoum, Australia
Advisory Board	Irwin King, Hong Kong, China Bao-Liang Lu, China Derong Liu, USA Jonathan H. Chan, Thailand Jun Wang, Hong Kong, China Lipo Wang, Singapore Nikhil R. Pal, India Nikola Kasabov, New Zealand Shiro Usui, Japan Soo-Young Lee, South Korea Takeshi Yamakawa, Japan Włodzisław Duch, Poland
Organising Committee Chairs	Kevin Wong, Australia B. Sumudu U. Mendis, Australia

Technical Program Committee

Shigeo Abe	Ryosuke Hosaka	Makoto Ohki
Sabri Arik	Chih-Yu Hsu	Masaaki Ohkita
Hideki Asoh	Xiaolin Hu	Gareth Oliver
Hongliang Bai	Kaizhu Huang	Toshiaki Omori
Sang-Woo Ban	Norhaslinda Kamaruddin	Tetsuya Onoda
Tao Ban	Keisuke Kameyama	Matashige Oyabu
Andre Barczak	Satoru Kato	Seiichi Ozawa
Younès Bennani	Jong Kyoung Kim	Shaoning Pang
Michael Bui	Kyung-Joong Kim	Zoltán Petres
Marek Bundzel	Min Young Kim	Huy Phan
Chee Seng Chan	Sungshin Kim	Santitham Prom-on
Jonathan Chan	Takanori Koga	Shri Rai
Jonathan H. Chan	Markus Koskela	Alexander Rast
Jianhui Chen	Ryosuke Kubota	Napoleon Reyes
Xinyu Chen	Takio Kurita	Ryo Saegusa
Siu-Yeung David Cho	James Kwok	Naoyuki Sato
Sung-Bae Cho	Daewon Lee	Stefan Schliebs
Seungjin Choi	Hyung-Soo Lee	Gourab Sen Gupta
Andrzej Cichocki	Nung Kiong Lee	Atsushi Shimada
Andrew Coward	Sang-Woong Lee	Hayaru Shouno
Justin Dauwels	Soo-Young Lee	John Sum
Zhaohong Deng	Chi Sing Leung	Shiliang Sun
Bin Dong	Chunshien Li	Jaewon Sung
Kenji Doya	Gary C. L. Li	Kenji Suzuki
Hiroshi Dozono	Chee-Peng Lim	Masa Takatsuka
Ke-Lin Du	Dudy Lim	Mieko Tanaka
Fuqing Duan	Heejin Lim	Chan Wai Ting
Kikuo Fujimura	Naiyan Lima	Heizo Tokutaka
Chun Che Fung	Iuon-Chang Lin	Hiroyuki Torikai
Andrew Gilman	Wilfred Lin	Whye Loon Tung
Roland Goecke	Steve Ling	Eiji Uchino
Eikou Gonda	Qingshan Liu	Hiroshi Wakuya
Ping Guo	Weixiang Liu	Liang Wan
Shanqing Guo	Zhiyong Liu	Dianhui Wang
Raj Gupta	Timothy Mann	Jun Wang
Amir Hadad	Sukanya Manna	Rongjie Wang
Hisashi Handa	Nobuo Matsuda	Zhanshan Wang
Xiong Hao	Robert (Bob) McKay	Yoshikazu Washizawa
Pitoyo Hartono	Sadaaki Miyamoto	Kazuho Watanabe
Ken Hawick	Takashi Morie	Bunthit Watanapa
Hanlin He	Mitsuteru Nakamura	Young-Gul Won
Zhaoshui He	Wakako Nakamura	Jiunn-Lin Wu
Akira Hiroes	Mitsuru Nakazawa	Liang-chuan Wu
Kevin Ho	Anto Satriyo Nugroho	Kui Xiang
Keiichi Horio	Chakarida Nukoolkit	Hai Xu

Zenglin Xu	Noha A. Yousri	Sulan Zhang
Nobuhiko Yamaguchi	Yingwei Yu	Yanming Zhang
Dong Yang	Jingling Yuan	Zhancheng Zhang
Haixuan Yang	Jeong Min Yun	Xin Zheng
Zhirong Yang	Xu Zang	Guoqiang Zhong
Qian Yin	Zhigang Zeng	Zhi-Hua Zhou
Xu-Cheng Yin	Qing Zhang	Dingyun Zhu

Table of Contents – Part I

Neurodynamics

Bayesian Interpretation of Border-Ownership Signals in Early Visual Cortex.....	1
<i>Haruo Hosoya</i>	
A Computational Model That Enables Global Amodal Completion Based on V4 Neurons	9
<i>Kazuhiro Sakamoto, Taichi Kumada, and Masafumi Yano</i>	
Quantitative Modeling of Neuronal Dynamics in <i>C. elegans</i>	17
<i>Masahiro Kuramochi and Yuishi Iwasaki</i>	
Human Localization by Fuzzy Spiking Neural Network Based on Informationally Structured Space	25
<i>Dalai Tang and Naoyuki Kubota</i>	
Computational Model of the Cerebral Cortex That Performs Sparse Coding Using a Bayesian Network and Self-Organizing Maps	33
<i>Yuuji Ichisugi and Haruo Hosoya</i>	
Complex Spiking Models: A Role for Diffuse Thalamic Projections in Complex Cortical Activity	41
<i>Peter Stratton and Janet Wiles</i>	
Mutual Information Analyses of Chaotic Neurodynamics Driven by Neuron Selection Methods in Synchronous Exponential Chaotic Tabu Search for Quadratic Assignment Problems	49
<i>Tetsuo Kawamura, Yoshihiko Horio, and Mikio Hasegawa</i>	
A General-Purpose Model Translation System for a Universal Neural Chip	58
<i>Francesco Galluppi, Alexander Rast, Sergio Davies, and Steve Furber</i>	
Realizing Ideal Spatiotemporal Chaotic Searching Dynamics for Optimization Algorithms Using Neural Networks	66
<i>Mikio Hasegawa</i>	
A Multiple Sound Source Recognition System Using Pulsed Neuron Model with Short Term Synaptic Depression	74
<i>Kaname Iwasa, Mauricio Kugler, Susumu Kuroyanagi, and Akira Iwata</i>	

A Model of Path Integration and Navigation Based on Head Direction Cells in Entorhinal Cortex	82
<i>Tanvir Islam and Ryutaro Fukuzaki</i>	
Model Studies on Time-Scaled Phase Response Curves and Synchronization Transition	91
<i>Yasuomi D. Sato, Keiji Okumura, and Masatoshi Shiino</i>	
Roles of Early Vision for the Dynamics of Border-Ownership Selective Neurons	99
<i>Nobuhiko Wagatsuma and Ko Sakai</i>	
Theoretical Analysis of Various Synchronizations in Pulse-Coupled Digital Spiking Neurons.....	107
<i>Hirofumi Ijichi and Hiroyuki Torikai</i>	
Emergence of Highly Nonrandom Functional Synaptic Connectivity Through STDP	116
<i>Hideyuki Kato and Tohru Ikeguchi</i>	
Modulation of Corticofugal Signals by Synaptic Changes in Bat's Auditory System	124
<i>Yoshihiro Nagase and Yoshiki Kashimori</i>	
Efficient Representation by Horizontal Connection in Primary Visual Cortex.....	132
<i>Hiroaki Sasaki, Shunji Satoh, and Shiro Usui</i>	
Stimulation of the Retinal Network in Bionic Vision Devices: From Multi-electrode Arrays to Pixelated Vision	140
<i>Robert G.H. Wilke, Gita Khalili Moghaddam, Socrates Dokos, Gregg Suaning, and Nigel H. Lovell</i>	
Spatial Feature Extraction by Spike Timing Dependent Synaptic Modification	148
<i>Kazuhisa Fujita</i>	
Learning Shapes Bifurcations of Neural Dynamics upon External Stimuli	155
<i>Tomoki Kurikawa and Kunihiko Kaneko</i>	
Towards Spatio-temporal Pattern Recognition Using Evolving Spiking Neural Networks	163
<i>Stefan Schliebs, Nuttapol Nuntalid, and Nikola Kasabov</i>	
Real-Time Simulation of Phosphene Images Evoked by Electrical Stimulation of the Visual Cortex	171
<i>Tamas Fehervari, Masaru Matsuoka, Hirotugu Okuno, and Tetsuya Yagi</i>	

An Effect of Inhibitory Connections on Synchronous Firing Assembly in the Inhibitory Connected Pulse Coupled Neural Network	179
<i>Hiroaki Kurokawa, Masahiro Yoshihara, and Masato Yonekawa</i>	
Array-Enhanced Stochastic Resonance in a Network of Noisy Neuromorphic Circuits.....	188
<i>Gessica Maria Tovar, Tetsuya Asai, and Yoshihito Amemiya</i>	
Computational Neuroscience and Cognitive Science	
Modelling the Interplay of Emotions, Beliefs and Intentions within Collective Decision Making Based on Insights from Social Neuroscience	196
<i>Mark Hoogendoorn, Jan Treur, C. Natalie van der Wal, and Arlette van Wissen</i>	
Visual Selective Attention Model Considering Bottom-Up Saliency and Psychological Distance.....	207
<i>Young-Min Jang, Sang-Woo Ban, and Minho Lee</i>	
Free-Energy Based Reinforcement Learning for Vision-Based Navigation with High-Dimensional Sensory Inputs	215
<i>Stefan Elfwing, Makoto Otsuka, Eiji Uchibe, and Kenji Doya</i>	
Dependence on Memory Pattern in Sensitive Response of Memory Fragments among Three Types of Chaotic Neural Network Models.....	223
<i>Toshiyuki Hamada, Jousuke Kuroiwa, Hisakazu Ogura, Tomohiro Odaka, Haruhiko Shirai, and Izumi Suwa</i>	
A Stimulus-Response Neural Network Model Prepared by Top-Down Signals	231
<i>Osamu Araki</i>	
A Novel Shape-Based Image Classification Method by Featuring Radius Histogram of Dilating Discs Filled into Regular and Irregular Shapes ...	239
<i>Xiaoyu Zhao, Chi Xu, Zheru Chi, and Dagan Feng</i>	
Learning Visual Object Categories and Their Composition Based on a Probabilistic Latent Variable Model	247
<i>Masayasu Atsumi</i>	
Evidence for False Memory Before Deletion in Visual Short-Term Memory	255
<i>Eiichi Hoshino and Ken Mogi</i>	
Novel Alternating Least Squares Algorithm for Nonnegative Matrix and Tensor Factorizations	262
<i>Anh Huy Phan, Andrzej Cichocki, Rafal Zdunek, and Thanh Vu Dinh</i>	

Computational Modeling and Analysis of the Role of Physical Activity in Mood Regulation and Depression	270
<i>Fiemke Both, Mark Hoogendoorn, Michel C.A. Klein, and Jan Treur</i>	

Data and Text Processing

Representation of Hypertext Documents Based on Terms, Links and Text Compressibility	282
<i>Julian Szymański and Włodzisław Duch</i>	
A Heuristic-Based Feature Selection Method for Clustering Spam Emails.....	290
<i>Jungsuk Song, Masashi Eto, Hyung Chan Kim, Daisuke Inoue, and Koji Nakao</i>	
Enhancement of Subjective Logic for Semantic Document Analysis Using Hierarchical Document Signature	298
<i>Sukanya Manna, Tom Gedeon, and B. Sumudu U. Mendis</i>	
Is Comprehension Useful for Mobile Semantic Search Engines?.....	307
<i>Ahmad Ali Iqbal and Aruna Seneviratne</i>	
A Novel Text Classification Approach Based on Deep Belief Network ...	314
<i>Tao Liu</i>	
A Probability Click Tracking Model Analysis of Web Search Results	322
<i>Yujiu Yang, Xinyi Shu, and Wenhua Liu</i>	
Intention Extraction From Text Messages	330
<i>Insu Song and Joachim Diederich</i>	

Adaptive Algorithms

m-SNE: Multiview Stochastic Neighbor Embedding	338
<i>Bo Xie, Yang Mu, and Dacheng Tao</i>	
Learning Parametric Dynamic Movement Primitives from Multiple Demonstrations	347
<i>Takamitsu Matsubara, Sang-Ho Hyon, and Jun Morimoto</i>	
An Algorithm on Multi-View Adaboost	355
<i>Zhijie Xu and Shiliang Sun</i>	
An Analysis of Speaker Recognition Using Bagging CAN2 and Pole Distribution of Speech Signals	363
<i>Shuichi Kurogi, Shota Mineishi, and Seitaro Sato</i>	

Sparse and Low-Rank Estimation of Time-Varying Markov Networks with Alternating Direction Method of Multipliers	371
<i>Jun-ichiro Hirayama, Aapo Hyvärinen, and Shin Ishii</i>	
Nearest Hit-Misses Component Analysis for Supervised Metric Learning	380
<i>Wei Yang, Kuanquan Wang, and Wangmeng Zuo</i>	
Backward-Forward Least Angle Shrinkage for Sparse Quadratic Optimization	388
<i>Tianyi Zhou and Dacheng Tao</i>	
An Enhanced Semi-supervised Recommendation Model Based on Green's Function	397
<i>Dingyan Wang and Irwin King</i>	
Reinforcement Learning by KFM Probabilistic Associative Memory Based on Weights Distribution and Area Neuron Increase and Decrease	405
<i>Takahiro Hada and Yuko Osana</i>	
Extraction of Reward-Related Feature Space Using Correlation-Based and Reward-Based Learning Methods	414
<i>Poramate Manoonpong, Florentin Wörgötter, and Jun Morimoto</i>	
Stationary Subspace Analysis as a Generalized Eigenvalue Problem	422
<i>Satoshi Hara, Yoshinobu Kawahara, Takashi Washio, and Paul von Bünnau</i>	
A Multi-class Object Classifier Using Boosted Gaussian Mixture Model	430
<i>Wono Lee and Minho Lee</i>	
Adaptive Ensemble Based Learning in Non-stationary Environments with Variable Concept Drift	438
<i>Teo Susnjak, Andre L.C. Barczak, and Ken A. Hawick</i>	
High Dimensional Non-linear Modeling with Bayesian Mixture of CCA	446
<i>Tikara Hosino</i>	
The Iso-regularization Descent Algorithm for the LASSO	454
<i>Manuel Loth and Philippe Preux</i>	
Logistic Label Propagation for Semi-supervised Learning	462
<i>Kenji Watanabe, Takumi Kobayashi, and Nobuyuki Otsu</i>	

A New Framework for Small Sample Size Face Recognition Based on Weighted Multiple Decision Templates	470
<i>Mohammad Sajjad Ghaemi, Saeed Masoudnia, and Reza Ebrahimpour</i>	
An Information-Spectrum Approach to Analysis of Return Maximization in Reinforcement Learning.....	478
<i>Kazunori Iwata</i>	
Analytical Approach to Noise Effects on Synchronization in a System of Coupled Excitable Elements	486
<i>Keiji Okumura and Masatoshi Shiino</i>	
Learning ECOC and Dichotomizers Jointly from Data	494
<i>Guoqiang Zhong, Kaizhu Huang, and Cheng-Lin Liu</i>	
Wavelet Entropy Measure Based on Matching Pursuit Decomposition and Its Analysis to Heartbeat Intervals	503
<i>Fausto Lucena, Andre Cavalcante, Yoshinori Takeuchi, Allan Kardec Barros, and Noboru Ohnishi</i>	

Bio-inspired Algorithms

Application Rough Sets Theory to Ordinal Scale Data for Discovering Knowledge	512
<i>Shu-Hsien Liao, Yin-Ju Chen, and Pei-Hui Chu</i>	
Dynamic Population Variation Genetic Programming with Kalman Operator for Power System Load Modeling	520
<i>Yanyun Tao, Minglu Li, and Jian Cao</i>	
A Robust Iris Segmentation with Fuzzy Supports	532
<i>C.C. Teo, H.F. Neo, G.K.O. Michael, C. Tee, and K.S. Sim</i>	
An Adaptive Local Search Based Genetic Algorithm for Solving Multi-objective Facility Layout Problem	540
<i>Kazi Shah Nawaz Ripon, Kyrre Glette, Mats Høvin, and Jim Torresen</i>	
Non-uniform Layered Clustering for Ensemble Classifier Generation and Optimality	551
<i>Ashfaqur Rahman, Brijesh Verma, and Xin Yao</i>	
Membership Enhancement with Exponential Fuzzy Clustering for Collaborative Filtering.....	559
<i>Kiatichai Treerattanapitak and Chuleerat Jaruskulchai</i>	

Real-Valued Multimodal Fitness Landscape Characterization for Evolution	567
<i>P. Caamaño, A. Prieto, J.A. Becerra, F. Bellas, and R.J. Duro</i>	
Reranking for Stacking Ensemble Learning	575
<i>Buzhou Tang, Qingcai Chen, Xuan Wang, and Xiaolong Wang</i>	
A Three-Strategy Based Differential Evolution Algorithm for Constrained Optimization	585
<i>Saber M. Elsayed, Ruhul A. Sarker, and Daryl L. Essam</i>	
A New Expansion of Cooperative Particle Swarm Optimization	593
<i>Hong Zhang</i>	
Adaptive Ensemble Learning Strategy Using an Assistant Classifier for Large-Scale Imbalanced Patent Categorization	601
<i>Qi Kong, Hai Zhao, and Bao-Liang Lu</i>	
Adaptive Decision Making in Ant Colony System by Reinforcement Learning	609
<i>Keiji Kamei and Masumi Ishikawa</i>	
A Cooperative Coevolutionary Algorithm for the Composite SaaS Placement Problem in the Cloud	618
<i>Zeratul Izzah Mohd Yusoh and Maolin Tang</i>	
A Swarm Intelligence Approach to the Quadratic Multiple Knapsack Problem	626
<i>Shyam Sundar and Alok Singh</i>	
Rough-Set-Based Association Rules Applied to Brand Trust Evaluation Model	634
<i>Shu-Hsien Liao, Yin-Ju Chen, and Pei-Hui Chu</i>	
A Genetic Algorithm to Find Pareto-Optimal Solutions for the Dynamic Facility Layout Problem with Multiple Objectives	642
<i>Kazi Shah Nawaz Ripon, Kyrre Glette, Mats Høvin, and Jim Torresen</i>	
Hierarchical Methods	
Topological Hierarchical Tree Using Artificial Ants	652
<i>Mustapha Lebbah and Hanane Azzag</i>	
Bottom-Up Generative Modeling of Tree-Structured Data	660
<i>Davide Baciucc, Alessio Micheli, and Alessandro Sperduti</i>	
Exploit of Online Social Networks with Community-Based Graph Semi-Supervised Learning	669
<i>Mingzhen Mo and Irwin King</i>	

XVIII Table of Contents – Part I

Hierarchical Lossless Image Coding Using Cellular Neural Network	679
<i>Seiya Takenouchi, Hisashi Aomori, Tsuyoshi Otake, Mamoru Tanaka, Ichiro Matsuda, and Susumu Itoh</i>	
Multivariate Decision Tree Function Approximation for Reinforcement Learning	687
<i>Hossein Bashashati Saghezchi and Masoud Asadpour</i>	
Improving Hierarchical Document Signature Performance by Classifier Combination	695
<i>Jieyi Liao, B. Sumudu U. Mendis, and Sukanya Manna</i>	
The Discovery of Hierarchical Cluster Structures Assisted by a Visualization Technique	703
<i>Ke-Bing Zhang, Mehmet A. Orgun, Yanchang Zhao, and Abhaya C. Nayak</i>	
Author Index	713

Table of Contents – Part II

Brain Computer Interface

Utilizing Fuzzy-SVM and a Subject Database to Reduce the Calibration Time of P300-Based BCI.....	1
<i>Sercan Taha Ahi, Natsue Yoshimura, Hiroyuki Kambara, and Yasuharu Koike</i>	
Importance Weighted Extreme Energy Ratio for EEG Classification	9
<i>Wenting Tu and Shiliang Sun</i>	
Toward Automated Electrode Selection in the Electronic Depth Control Strategy for Multi-unit Recordings	17
<i>Gert Van Dijck, Ahmad Jezzini, Stanislav Herwik, Sebastian Kisban, Karsten Seidl, Oliver Paul, Patrick Ruther, Francesca Ugolotti Serventi, Leonardo Fogassi, Marc M. Van Hulle, and Maria Alessandra Umiltà</i>	
Tensor Based Simultaneous Feature Extraction and Sample Weighting for EEG Classification	26
<i>Yoshikazu Washizawa, Hiroshi Higashi, Tomasz Rutkowski, Toshihisa Tanaka, and Andrzej Cichocki</i>	
A Tongue-Machine Interface: Detection of Tongue Positions by Glossokinetic Potentials.....	34
<i>Yunjun Nam, Qibin Zhao, Andrzej Cichocki, and Seungjin Choi</i>	
Practical Surface EMG Pattern Classification by Using a Selective Desensitization Neural Network	42
<i>Hiroshi Kawata, Fumihide Tanaka, Atsuo Suemitsu, and Masahiko Morita</i>	
Reliability-Based Automatic Repeat reQuest with Error Potential-Based Error Correction for Improving P300 Speller Performance	50
<i>Hiromu Takahashi, Tomohiro Yoshikawa, and Takeshi Furuhashi</i>	
An Augmented-Reality Based Brain-Computer Interface for Robot Control	58
<i>Alexander Lenhardt and Helge Ritter</i>	
Brain Computer Interfaces: A Recurrent Neural Network Approach	66
<i>Gareth Oliver and Tom Gedeon</i>	
Research on Relationship between Saccade-Related EEG Signals and Selection of Electrode Position by Independent Component Analysis....	74
<i>Arao Funase, Motoaki Mouri, Andrzej Cichocki, and Ichi Takumi</i>	

Kernel Methods

Application of SVM-Based Filter Using LMS Learning Algorithm for Image Denoising	82
<i>Tzu-Chao Lin, Chien-Ting Yeh, and Mu-Kun Liu</i>	
Tuning N-gram String Kernel SVMs via Meta Learning	91
<i>Nuwan Gunasekara, Shaoning Pang, and Nikola Kasabov</i>	
Bilinear Formulated Multiple Kernel Learning for Multi-class Classification Problem	99
<i>Takumi Kobayashi and Nobuyuki Otsu</i>	
Feature Extraction Using Support Vector Machines	108
<i>Yasuyuki Tajiri, Ryosuke Yabuwaki, Takuya Kitamura, and Shigeo Abe</i>	
Class Information Adapted Kernel for Support Vector Machine	116
<i>Tasadduq Imam and Kevin Tickle</i>	
Gaze Pattern and Reading Comprehension	124
<i>Tan Vo, B. Sumudu U. Mendis, and Tom Gedeon</i>	
A Theoretical Framework for Multi-sphere Support Vector Data Description	132
<i>Trung Le, Dat Tran, Wanli Ma, and Dharmendra Sharma</i>	
Fast Implementation of String-Kernel-Based Support Vector Classifiers by GPU Computing	143
<i>Yongquan Shi, Tao Ban, Shanqing Guo, Qiuliang Xu, and Youki Kadobayashi</i>	

Model Generation and Classification

Classification of Imbalanced Data by Combining the Complementary Neural Network and SMOTE Algorithm	152
<i>Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung</i>	
Generalization Error of Faulty MLPs with Weight Decay Regularizer ...	160
<i>Chi Sing Leung, John Sum, and Shue Kwan Mak</i>	
The Effect of Bottlenecks on Generalisation in Backpropagation Neural Networks	168
<i>Xu Zang</i>	
Lagrange Programming Neural Networks for Compressive Sampling	177
<i>Ping-Man Lam, Chi Sing Leung, John Sum, and A.G. Constantinides</i>	

Input and Output Mapping Sensitive Auto-Associative Multilayer Perceptron for Computer Interface System Based on Image Processing of Laser Pointer Spot	185
<i>Chanwoong Jung, Sang-Woo Ban, Sungmoon Jeong, and Minho Lee</i>	
Improving Recurrent Neural Network Performance Using Transfer Entropy	193
<i>Oliver Obst, Joschka Boedecker, and Minoru Asada</i>	
Design of Artificial Neural Networks Using Differential Evolution Algorithm	201
<i>Beatriz A. Garro, Humberto Sossa, and Roberto A. Vázquez</i>	
ESNs with One Dimensional Topography	209
<i>N. Michael Mayer, Matthew Browne, and Horng Jason Wu</i>	
Computational Advance in Bioinformatics	
iGAPK: Improved GAPK Algorithm for Regulatory DNA Motif Discovery	217
<i>Dianhui Wang and Xi Li</i>	
A Computer-Aided Detection System for Automatic Mammography Mass Identification	226
<i>Hussein Samma, Chee Peng Lim, and Ali Samma</i>	
Exploring Features and Classifiers to Classify MicroRNA Expression Profiles of Human Cancer	234
<i>Kyung-Joong Kim and Sung-Bae Cho</i>	
SOMIX: Motifs Discovery in Gene Regulatory Sequences Using Self-Organizing Maps	242
<i>Nung Kiong Lee and Dianhui Wang</i>	
Microarray-Based Disease Classification Using Pathway Activities with Negatively Correlated Feature Sets	250
<i>Pitak Sootanan, Santitham Prom-on, Asawin Meechai, and Jonathan H. Chan</i>	
Data Mining for Cybersecurity	
A Malware Detection Algorithm Based on Multi-view Fusion	259
<i>Shanqing Guo, Qixia Yuan, Fengbo Lin, Fengyu Wang, and Tao Ban</i>	
A Fast Kernel on Hierarchical Tree Structures and Its Application to Windows Application Behavior Analysis	267
<i>Tao Ban, Ruo Ando, and Youki Kadobayashi</i>	

Evolution of Information Retrieval in Cloud Computing by Redesigning Data Management Architecture from a Scalable Associative Computing Perspective.....	275
<i>Amir H. Basirat and Asad I. Khan</i>	
Factorizing Class Characteristics via Group MEBs Construction	283
<i>Ye Chen, Shaoning Pang, and Nikola Kasabov</i>	
A Hybrid Fuzzy-Genetic Colour Classification System with Best Colour Space Selection under Dynamically-Changing Illumination	291
<i>Heesang Shin, Napoleon H. Reyes, and Andre L. Barczak</i>	
Identifier Based Graph Neuron: A Light Weight Event Classification Scheme for WSN.....	300
<i>Nomica Imran and Asad Khan</i>	
Clustering Categorical Data Using an Extended Modularity Measure ...	310
<i>Lazhar Labiod, Nistor Grozavu, and Younès Bennani</i>	
A Morphological Associative Memory Employing a Reverse Recall	321
<i>Hidetaka Harada and Tsutomu Miki</i>	
Analysis of Packet Traffics and Detection of Abnormal Traffics Using Pareto Learning Self Organizing Maps	329
<i>Hiroshi Dozono, Masanori Nakakuni, Takaru Kabashima, and Shigeomi Hara</i>	
Log Analysis of Exploitation in Cloud Computing Environment Using Automated Reasoning	337
<i>Ruo Ando, Kang Byung, and Youki Kadobayashi</i>	
Self-organizing Maps and Their Applications	
A Multidirectional Associative Memory Based on Self-organizing Incremental Neural Network	344
<i>Hui Yu, Furao Shen, and Osamu Hasegawa</i>	
Range Image Registration Using Particle Filter and Competitive Associative Nets	352
<i>Shuichi Kurogi, Tomokazu Nagi, and Takeshi Nishida</i>	
Rotation Invariant Categorization of Visual Objects Using Radon Transform and Self-Organizing Modules.....	360
<i>Andrew P. Papliński</i>	
Learning Topological Constraints in Self-Organizing Map	367
<i>Guénaël Cabanes and Younès Bennani</i>	

Pseudo-network Growing for Gradual Interpretation of Input Patterns <i>Ryotaro Kamimura</i>	375
The Adaptive Authentication System for Behavior Biometrics Using Pareto Learning Self Organizing Maps <i>Hiroshi Dozono, Masanori Nakakuni, Shinsuke Itou, and Shigeomi Hara</i>	383
Human Action Recognition by SOM Considering the Probability of Spatio-temporal Features <i>Yanli Ji, Atsushi Shimada, and Rin-ichiro Taniguchi</i>	391
On Generalization Error of Self-Organizing Map <i>Fumiaki Saitoh and Sumio Watanabe</i>	399
A Novel Approach for Sound Approaching Detection <i>Hirofumi Tsuzuki, Mauricio Kugler, Susumu Kuroyanagi, and Akira Iwata</i>	407
Ground Penetrating Radar System with Integration of Mutimodal Information Based on Mutual Information among Multiple Self-Organizing Maps <i>Akira Hirose, Ayato Ejiri, and Kunio Kitahara</i>	415
Information-Theoretic Competitive and Cooperative Learning for Self-Organizing Maps <i>Ryotaro Kamimura</i>	423
Early Recognition Based on Co-occurrence of Gesture Patterns <i>Atsushi Shimada, Manabu Kawashima, and Rin-ichiro Taniguchi</i>	431
A Dynamically Reconfigurable Platform for Self-Organizing Neural Network Hardware <i>Hakaru Tamukoh and Masatoshi Sekine</i>	439
Inversion of Many-to-One Mappings Using Self-Organising Maps <i>Anne O. Mus</i>	447
Self-Organizing Hidden Markov Models <i>Nobuhiko Yamaguchi</i>	454
An Image-Aided Diagnosis System for Dementia Classification Based on Multiple Features and Self-Organizing Map <i>Shih-Ting Yang, Jiann-Der Lee, Chung-Hsien Huang, Jiun-Jie Wang, Wen-Chuin Hsu, and Yau-Yau Wai</i>	462
Parallel Batch Training of the Self-Organizing Map Using OpenCL <i>Masahiro Takatsuka and Michael Bui</i>	470

Fast Kohonen Feature Map Associative Memory Using Area Representation for Sequential Analog Patterns	477
<i>Hiroki Midorikawa and Yuko Osana</i>	
Machine Learning Applications to Image Analysis	
Facial Expression Based Automatic Album Creation	485
<i>Abhinav Dhall, Akshay Asthana, and Roland Goecke</i>	
Age Classification Combining Contour and Texture Feature	493
<i>Yan-Ming Tang and Bao-Liang Lu</i>	
A Salient Region Detector for GPU Using a Cellular Automata Architecture	501
<i>David Huw Jones, Adam Powell, Christos-Savvas Bouganis, and Peter Y.K. Cheung</i>	
VG-RAM WNN Approach to Monocular Depth Perception	509
<i>Hélio Perroni Filho and Alberto F. De Souza</i>	
Semi-supervised Classification by Local Coordination	517
<i>Gelan Yang, Xue Xu, Gang Yang, and Jianming Zhang</i>	
RANSAC Based Ellipse Detection with Application to Catadioptric Camera Calibration	525
<i>Fuqing Duan, Liang Wang, and Ping Guo</i>	
Speed Up Image Annotation Based on LVQ Technique with Affinity Propagation Algorithm	533
<i>Song Lin, Yao Yao, and Ping Guo</i>	
Dictionary of Features in a Biologically Inspired Approach to Image Classification	541
<i>Sepehr Jalali, Joo Hwee Lim, Sim Heng Ong, and Jo Yew Tham</i>	
A Highly Robust Approach Face Recognition Using Hausdorff-Trace Transformation	549
<i>Werasak Kurutach, Rerkchai Fooprateepsiri, and Suronapee Phoomvuthisarn</i>	
Blind Image Tamper Detection Based on Multimodal Fusion	557
<i>Girija Chetty, Monica Singh, and Matthew White</i>	
Orientation Dependence of Surround Modulation in the Population Coding of Figure/Ground	565
<i>Keiichi Kondo and Ko Sakai</i>	
Increased Robustness against Background Noise: Pattern Recognition by a Neocognitron	574
<i>Kunihiko Fukushima</i>	

Improving the Performance of Facial Expression Recognition Using Dynamic, Subtle and Regional Features	582
<i>Ligang Zhang and Dian Tjondronegoro</i>	
Identity Retrieval in Biometric Access Control Systems Using Multimedia Fusion	590
<i>Girija Chetty, Renuka Biswas, and Julian Goodwin</i>	
Improvement of Reuse of Classifiers in CBIR Using SVM Active Learning	598
<i>Masaaki Tekawa and Motonobu Hattori</i>	
Realizing Hand-Based Biometrics Based on Visible and Infrared Imagery	606
<i>Goh Kah Ong Michael, Tee Connie, Teo Chuan Chin, Neo Han Foon, and Andrew Teoh Beng Jin</i>	
Visual Object Detection by Specifying the Scale and Rotation Transformations	616
<i>Yasuomi D. Sato, Jenia Jitsev, and Christoph von der Malsburg</i>	
Multi-view Gender Classification Using Hierarchical Classifiers Structure	625
<i>Tian-Xiang Wu and Bao-Liang Lu</i>	
Partial Extraction of Edge Filters by Cumulant-Based ICA under Highly Overcomplete Model	633
<i>Yoshitatsu Matsuda and Kazunori Yamaguchi</i>	
Random Projection Tree and Multiview Embedding for Large-Scale Image Retrieval	641
<i>Bo Xie, Yang Mu, Mingli Song, and Dacheng Tao</i>	
Online Gesture Recognition for User Interface on Accelerometer Built-in Mobile Phones	650
<i>BongWhan Choe, Jun-Ki Min, and Sung-Bae Cho</i>	
Constructing Sparse KFDA Using Pre-image Reconstruction	658
<i>Qing Zhang and Jianwu Li</i>	
Applications	
Learning Basis Representations of Inverse Dynamics Models for Real-Time Adaptive Control	668
<i>Yasuhito Horiguchi, Takamitsu Matsubara, and Masatsugu Kidode</i>	
Feel Like an Insect: A Bio-inspired Tactile Sensor System	676
<i>Sven Hellbach, André Frank Krause, and Volker Dürr</i>	

XXVI Table of Contents – Part II

Spectral Domain Noise Suppression in Dual-Sensor Hyperspectral Imagery Using Gaussian Processes	684
<i>Arman Melkumyan and Richard J. Murphy</i>	
A High Order Neural Network to Solve Crossbar Switch Problem	692
<i>Yuxin Ding, Li Dong, Ling Wang, and Guohua Wu</i>	
Identification of Liquid State of Scrap in Electric Arc Furnace by the Use of Computational Intelligence Methods	700
<i>Marcin Blachnik, Tadeusz Wieczorek, Krystian Mączka, and Grzegorz Kopeć</i>	
Simulating Wheat Yield in New South Wales of Australia Using Interpolation and Neural Networks	708
<i>William W. Guo, Lily D. Li, and Greg Whymark</i>	
Investment Appraisal under Uncertainty – A Fuzzy Real Options Approach	716
<i>Shu-Hsien Liao and Shiu-Hwei Ho</i>	
Developing a Robust Prediction Interval Based Criterion for Neural Network Model Selection	727
<i>Abbas Khosravi, Saeid Nahavandi, and Doug Creighton</i>	
Author Index	735

Bayesian Interpretation of Border-Ownership Signals in Early Visual Cortex

Haruo Hosoya

Department of Computer Science, The University of Tokyo
7th Bld. of Faculty of Science, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan

Abstract. Mammalian visual cortex is known to have various neuronal response properties that depend on stimuli outside classical receptive fields. In this article, we give a probabilistic explanation to one such property called border-ownership signals, by interpreting them as posterior joint probabilities of a low-level edge property and a high-level figure property. We show that such joint probabilities can be found in a hierarchical Bayesian network mimicking visual cortex, and indeed they exhibit simulative responses qualitatively similar to physiological data.

1 Introduction

Border-ownership is the property of a figural border whether it belongs to a figure on one side or the other. In psychology, this notion is considered to be an important clue for the perceptual segregation of figures from backgrounds. Recently, in neurophysiology, closely related neuronal responses have been discovered in early visual cortex of the macaque [13]. For example, suppose that each image in Figure 1A is presented to an animal. In a classical interpretation of visual cortex, an edge-selective neuron (with a receptive field indicated by a circle) should respond equally to images (a) and (b), and equally to (c) and (d), since exactly the same stimulus is presented to the receptive field in either pair of images [3]. However, Zhou et al.'s physiological experiments revealed the existence of such neurons as responding strongly to (a) and (c), and weakly to (b) and (d), which appear to be influenced by the surrounding context whether the square figure is placed on the left hand side or on the right hand side [13].

In this article, we show that these signals can be interpreted as a posterior joint probability of a low-level visual property x (e.g., edge orientation and phase) and a high-level visual feature u (e.g., figure presence), given stimuli to the classical receptive field R and to the surrounding context C :

$$P(x, u|R, C)$$

For example, see Figure 1B and consider a neuron selective to the simultaneous presence of a vertical edge (x) in the classical receptive field (R) and a square shape in whatever color placed within the dotted box (u). Then, this neuron would behave like the hypothetical neuron illustrated in Figure 1A, since it would respond when a square appears in the dotted box (which happens to be

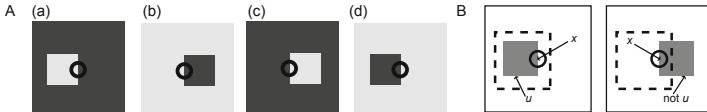


Fig. 1. (A) An image set used for the standard test of border-ownership. The center circle depicts the classical receptive field of a neuron. (B) A model cell $P(x, u | C, R)$ where x represents a vertical edge in the classical receptive field and u represents a square figure placed within the dotted box.

on the left hand side of R) and one of its belonging edges in R (Figure 1B, left), but not when a square exceeds the dotted box even if a belonging edge still appears in R (right).

We will show that, if we construct a Bayesian network [8] resembling the hierarchy of visual cortex, then the posterior joint probabilities mentioned above can readily be found, under certain assumptions, as a part of the circuitry elements composing the network. In this, we will use, as a theoretical basis, Ichisugi's approximate belief propagation algorithm, which can be mapped to cortical six-layer structure and bidirectional connections [4]. We will also present the results of our computer simulation where some model neurons in our network exhibited border-ownership response properties similar to those found experimentally.

2 Bayesian Model of Visual Cortex

In general, a Bayesian network [8] is a directed acyclic graph whose each node represents a random variable. We assume that the range of each variable is finite and fixed at the same time as the structure of the network is given; we call each value of a variable *unit*. We may regard each unit as a functional column and each node as a hypercolumn containing such columns. Thus, a set of nodes may correspond to a cortical area containing several hypercolumns, and a whole Bayesian network to a whole or a part of cerebral cortex. Edges between nodes determine probabilistic dependencies between variables, a more precise explanation given below. The direction of an edge represents the relationship between two cortical areas in the cortical hierarchy, that is, an edge emanates from a node for a higher area to a node for a lower area.

Figure 2 shows an instance of Bayesian network that mimics a part of visual cortex. This network will be used throughout our investigation. It consists of a hierarchy of four layers. Although we do not claim an exact correspondence, one could regard each layer as V1, V2, V4, and IT. The number of nodes are increasingly larger in lower layers. Each node in a layer is connected to several nodes in the layer right below, with an overlap with the child nodes of an adjacent node in the upper layer. Note that the last point makes convergence and divergence between different layers, like in actual visual cortex.

The Bayesian networks framework stipulates that each node X is probabilistically dependent on its parents U_1, \dots, U_n . This is generally formalized by a

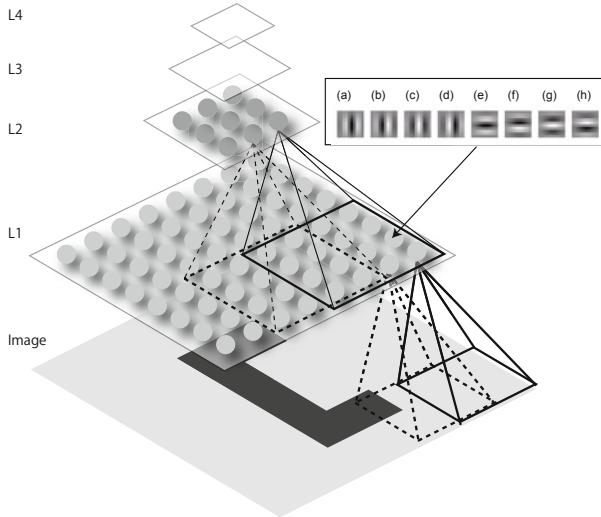


Fig. 2. A Bayesian network mimicking visual cortex that is used in the simulation. Layer L1 has 12×12 nodes each with 8 units, L2 has 11×11 nodes each with 20 units, L3 has 4×4 nodes each with 10 units, and L4 has 3×3 nodes each with 10 units. An input to the network is a gray-scale image of 100×100 pixels. Each unit of an L1 node receives a 10×10 patch of the image processed by one of 8 Gabor filters (with 2 orientations and 4 phases) and normalized to form a probability distribution (a negative value truncated to 0).

conditional probability table $P(X|U_1, \dots, U_n)$. However, since a direct representation of such table in cerebral cortex would require an exponential number of synapses in the number of parent nodes (hypercolumns), a more efficient representation seems biologically more plausible. In this article, we use Ichisugi's assumption that the full conditional probability $P(X|U_1, \dots, U_n)$ is expressed as

$$P(X|U_1, \dots, U_n) \propto \sum_l P(X|U_l) \quad (1)$$

using conditional probability tables $P(X|U_i)$ that could each be represented by the inter-columnar weights between the hypercolumns X and U_i [4]. (For a node X with no parent, we assume $P(X)$ to be uniform.) This assumption can intuitively be justified as follows: in order for a unit of the child node X to fire, it is enough that a unit of *one* of the parent nodes U_1, \dots, U_m fires.

A typical way of working with a Bayesian network is to compute posterior probabilities of each variable given inputs. In this article, we use Pearl's belief propagation, a well-known algorithm for the inference of posterior probabilities [8]. The algorithm works by local message passing of vectors among nodes and attempts to calculate posterior probabilities after several iterations. Because

of this locality of computation, an attractive hypothesis has arisen that the algorithm may be employed by cerebral cortex. Indeed, several theoreticians have proposed neural implementations of belief propagation, with detailed mappings to cortical six-layer structure and forward and backward connections [249].

It can be shown that the posterior joint probabilities mentioned in the introduction can be found as a part of the belief propagation algorithm, under the following assumptions.

1. Belief propagation computes posteriors with sufficient precision despite the presence of undirected cycles (i.e., cycles ignoring edge directions) in our network (cf. [7]).
2. Stimuli to the receptive field are chosen from a distribution approximately uniform and independent from stimuli to the context.

Formally, if we write R_X for stimuli appearing in X 's receptive field and $C_{U \rightarrow X}$ for stimuli appearing in the surrounding context that can influence a parent node U , then the posterior $\text{BEL}(x)$ of node X given stimuli can be expressed as

$$\text{BEL}(x) \approx \alpha \sum_l \sum_{u_l} \sigma(x, u_l) \quad (2)$$

where

$$\sigma(x, u) \propto P(x, u | R_X, C_{U \rightarrow X}). \quad (3)$$

for a constant α . For lack of space, a full derivation is elided here.

3 Simulation

We conducted a simulation in order to see whether the variable $\sigma(x, u)$ introduced in the last section is similar to border-ownership signals observed experimentally. We used a network as shown in Figure 2. We trained the network by using a simple unsupervised learning based on Kohonen's self-organizing maps [6], presenting at each time either a square or a C-shape figure of a random size at a random location (cf. [13]). Whenever we tested the network, we presented a Gabor-filtered input image to layer L1, performed the belief propagation algorithm, and then measured the variable $\sigma(x, u)$ for a chosen unit x of node X and a chosen unit u of one of X 's parent nodes, U .¹ From here on, we call such variable *model cell*. In particular, when we choose nodes X and U from layers L1 and L2, respectively, we refer to the model cells by L1-L2 model cells. We could similarly define L2-L3 and L3-L4 model cells, though the present paper shows only results from L1-L2 cells due to lack of space. Among 77,440 L1 model cells, we picked up 92 for analysis of response properties.²

¹ In the belief propagation, each vector in the algorithm was initialized by a uniform probability distribution except for the input in the bottom layer and propagation was iterated 5 times, which seemed to be sufficient for convergence.

² We chose model cells that had receptive field centers within the central 50×50 pixels area and responded more strongly to a vertical line segment of size 1×11 pixels than a horizontal one (with a response 1.4 times larger than the response to the white blank image).

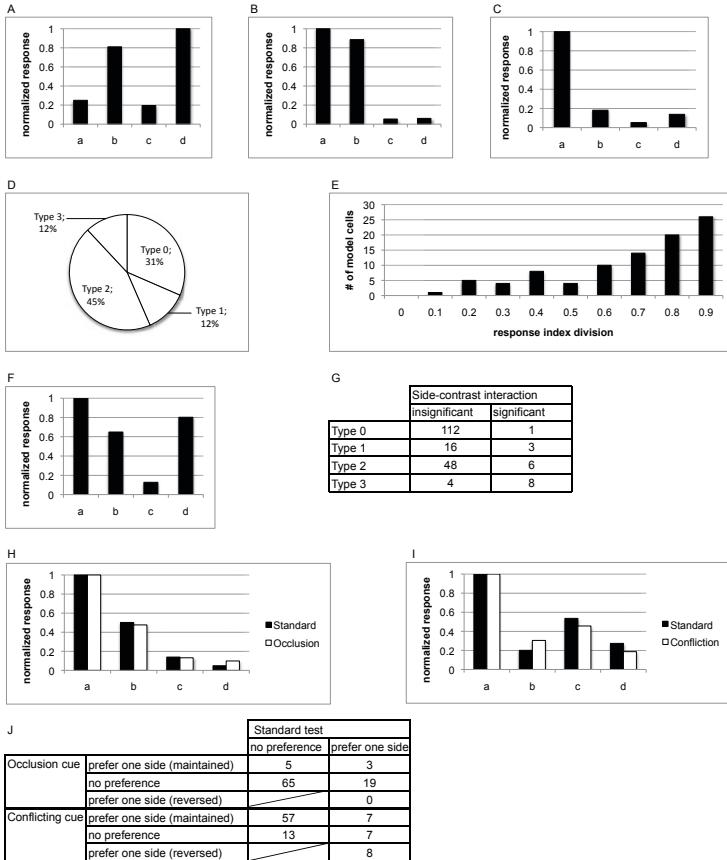


Fig. 3. Simulation results on L1-L2 model cells: (A, B, C) Responses in the standard test for sample model cells of type 1, 2, and 3 (respectively). The responses are normalized by dividing each value by the maximal one. The figure side factor was measured by using the index $\rho_{\text{side}} = \min \{(a + c)/(b + d), (b + d)/(a + c)\}$ and the contrast polarity factor by the index $\rho_{\text{contrast}} = \min \{(a + b)/(c + d), (c + d)/(a + b)\}$, where a, b, c , and d each denote the responses to the four images depicted in Figure **A**. The categorization was done by the threshold $\tau = 0.6$, e.g., type 1 was determined by $\rho_{\text{side}} < \tau$ and $\rho_{\text{contrast}} \geq \tau$. (D) Distribution of the types of all tested model cells. (E) Distribution of response indices for side preference for all tested model cells. (F) Responses in the standard test for a sample model cell showing a side-contrast interaction. The interaction was determined by the index $\rho_{\text{interact}} = \min \{(a + d)/(b + c), (b + c)/(a + d)\}$ with the same threshold τ . (G) Distributions of side-contrast interactions of all tested model cells of each type. (H) Responses of a sample model cell preserving the side preference in the occlusion cue test. (I) Responses of a sample model cell preserving the side preference in the conflicting cue test. (J) Distributions of model cells preserving or changing their side preferences by different sets of test images.

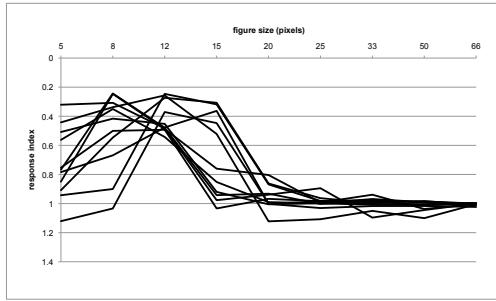


Fig. 4. Simulation results on L1-L2 model cells: Side preference indices of all type 1 and 3 model cells relative to the size of a presented square figure. The response indices were determined by $(a + c)/(b + d)$ if the cell preferred a figure of 12×12 pixels on the left, and $(b + d)/(a + c)$ otherwise.

In the primate experiments [13], the responses of cells to images as in Figure 1A were categorized by (1) whether the response depended on the side on which the square figure appears and (2) whether the response depended on the contrast polarity of the edge appearing in the classical receptive field. Accordingly, there are the following four types.

- Type 1.** The response strongly depends on the figure side but not on the contrast polarity (e.g., responding only to (a) and (c) in Figure 1A).
- Type 2.** The response strongly depends on the contrast polarity but not on the figure side (e.g., responding only to (a) and (b)).
- Type 3.** The response strongly depends on both factors (e.g., responding only to (a)).
- Type 0.** The response barely depends on either factor.

In our simulation, we found L1-L2 model cells of all these types. Figures 3A, B, and C show the normalized responses of typical sample model cells of types 1, 2, and 3, respectively, to the four images like Figure 1A with a square figure of 12×12 pixels (we call this test the standard test). Figure 3D shows the distribution of the types of our model cells, which indicates that, though type 1 and 3 cells existed, type 0 and 2 cells formed a majority, similarly to macaque V1 (type 0, 34%; type 1, 3%; type 2, 48%; type 3, 15%) [13, Figure 16]. Figure 3E shows the histogram of the index for figure side preferences for all the tested model cells, which confirms that there were a noticeable number of cells, though not many, that were clearly sensitive to the figure side ($\rho_{\text{side}} < \tau$). This histogram is also qualitatively similar to the physiological result [13, Figure 15]. As in macaque visual cortex, we found some model cells exhibiting significant interactions between the figure side and the contrast polarity factors. Figure 3F shows the responses of a sample type 2 cell with a significant interaction, which prefers a white square appearing on whichever side. Also, the model cells with side-contrast interaction had various types, as the distribution is summarized in Figure 3G.

Like in experimental data, half model cells had invariance in side preferences with respect to the figure size (Figure 4). Also, a few, though not as many as in the experiments, maintained their side preferences when the figure was obscured by occlusion cues and conflicting cues (Figures 3H, I, and J).

4 Discussion

In this article, we have shown that posterior joint probabilities of low-level and high-level visual features can be found as the variable $\sigma(x, u)$ in a properly structured Bayesian network under several conditions, and exhibit responses resembling to some extent border-ownership signals experimentally observed [13]. We argue that this framework provides a unified view for understanding various properties of these signals.

To see how the four types of model cells emerged in the standard test, first take a unit x corresponding to the Gabor filter (b) in Figure 2 preferring a vertical edge with black on the left, and a parent unit u preferring a square appearing on the left hand side. Then, the variable $\sigma(x, u)$ would represent a type 3 cell responding only to a black square on the left. Second, if we instead take a unit x corresponding to the filter (c) preferring a vertical edge with no contrast polarity, then the variable would represent a type 1 cell responding to either a black or a white square on the left. Finally, take a unit u indifferent to the presented figure; then, the variable would represent a type 2 cell if x corresponds to the filter (b) or (d), and a type 0 cell if x corresponds to the filter (a) or (c).

We can also understand various characteristics of border-ownership signals by assuming certain invariance properties in the parent unit u 's responses. Interaction between the side preference and the contrast polarity mentioned in the last section might be because the parent unit u had invariance in the figure position. For example, if u prefers a white square appearing within a certain range of position and x prefers any vertical edge (i.e., the Gabor filter (c)), then the variable $\sigma(x, u)$ would respond to images (a) and (d) but not to (b) and (c) in Figure 1A, representing a type 0 cell with a side-contrast interaction (Figure 3F). This view may further allow a similar interpretation of some cells' side preferences invariant to the figure size changes, the occlusion cue, and the conflicting cue.

As a final remark, border-ownership has originally been considered as a mechanism for figure-ground segregation and, indeed, Zhou et al.'s experiment as well as many computational models [10, 5, 12, 11, 1] were designed in this context. On the other hand, our present work did not make much connection in this respect. In particular, while our model stipulates that border-ownership signals arise as an interaction of high-level and low-level columnar activations, it does not specify how these signals are used. In fact, one could interpret our model as suggesting that these signals might be a kind of side-effects from figure recognition rather than clues to it.

Acknowledgments. The author is grateful to Yuuji Ichisugi, Hiroki Terashima, and Florian Röhrbein for useful comments and suggestions.

References

1. Craft, E., Schütze, H., Niebur, E., von der Heydt, R.: A neural model of figure-ground organization. *Journal of Neurophysiology* 97, 4310–4326 (2007)
2. George, D., Hawkins, J.: A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In: *Proceedings of International Joint Conference on Neural Networks*, vol. 3, pp. 1812–1817 (2005)
3. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology* 195, 215–243 (1968)
4. Ichisugi, Y.: A cerebral cortex model that self-organizes conditional probability tables and executes belief propagation. In: *International Joint Conference on Neural Networks (IJCNN 2007)*, pp. 178–183 (2007)
5. Kikuchi, M., Akashi, Y.: A model of border-ownership coding in early vision. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) *ICANN 2001. LNCS*, vol. 2130, pp. 1069–1074. Springer, Heidelberg (2001)
6. Kohonen, T.: *Self-Organizing Maps*, 3rd edn. Springer, Heidelberg (2000)
7. Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: *Proceedings of Uncertainty in Artificial Intelligence*, pp. 467–475 (1999)
8. Pearl, J.: *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco (1997)
9. Röhrbein, F., Eggert, J., Körner, E.: A cortex-inspired neural-symbolic network for knowledge representation. In: *Proceedings of the IJCAI Workshop on Neural-Symbolic Learning and Reasoning* (2007)
10. Sajda, P., Finkel, L.H.: Intermediate-level visual representations and the construction of surface perception. *Journal of Cognitive Neuroscience* 7(2), 267–291 (1995)
11. Sakai, K., Nishimura, H.: Surrounding suppression and facilitation in the determination of border ownership. *Journal of Cognitive Neuroscience* 18(4), 562–579 (2006)
12. Zhaoping, L.: Border ownership from intracortical interactions in visual area V2. *Neuron* 47, 143–153 (2005)
13. Zhou, H., Friedman, H.S., von der Heydt, R.: Coding of border ownership in monkey visual cortex. *Journal of Neuroscience* 20, 6594–6611 (2000)

A Computational Model That Enables Global Amodal Completion Based on V4 Neurons

Kazuhiro Sakamoto*, Taichi Kumada, and Masafumi Yano

Research Institute of Electrical Communication, Tohoku University,
2-1-1 Katahira Aoba-ku, Sendai, 980-8577, Japan
sakamoto@riec.tohoku.ac.jp

Abstract. In natural scenes, objects are often partially occluded. Nonetheless, our visual system can readily complete an object shape from available information and perceive it as a whole, a process known as amodal completion. Although implementation of this completion process is an important issue, visual computation for completion, based on both the local continuity of contours and on global regularities, such as symmetry, has received little attention. Here, we show a novel neuro-computational model based on recent physiological findings, in particular those in visual area V4. The model enables amodal completion through the evaluation of a global constellation of features describing a shape's contours.

Keywords: occluded shape, amodal completion, symmetry, area V4.

1 Introduction

In natural scenes, objects are often partially occluded and only incomplete information about their shapes is available. Even from such incomplete information, our visual system can readily complete the contours of a shape and perceive it as a whole. This process is known as amodal completion.

This completion process provides ambiguous but coherent interpretations regarding the occluded shape (Fig. 1), and implementation of this process is an important issue in visual computation that enables object perception in natural scenes. However, most of the computational models for visual completion have been based on the local continuity of shape contour alone [1][2] and have not included global regularities in shape, such as symmetry, in computing visual completion tasks [3]. Only Leeuwenberg and colleagues have tackled this latter problem [4], but they proposed only a psychological model, not an effective computational model.

Object-shape processing in the brain has been attributed to the occipitotemporal pathway in the cerebral cortex. Along this pathway, neurons in area V4 are known to respond well to complex features presented within their receptive fields but not to simple local line-segments [5]. Specifically, Connor and

* This work was supported by the grant from the Ministry of Education, Culture, Sports, Science, and Technology of Japan (22120504 and 22500283).

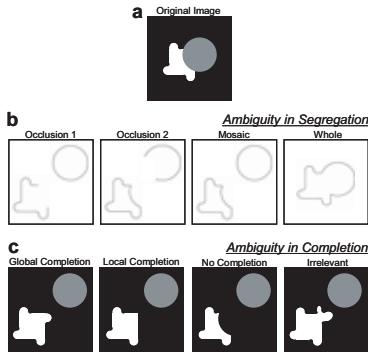


Fig. 1. Amodal completion is ambiguous. For image (a), segregation of the contours (b) and completion of them (c) are computationally ambiguous, although each segregation or completion pattern has different perceptual relevance.

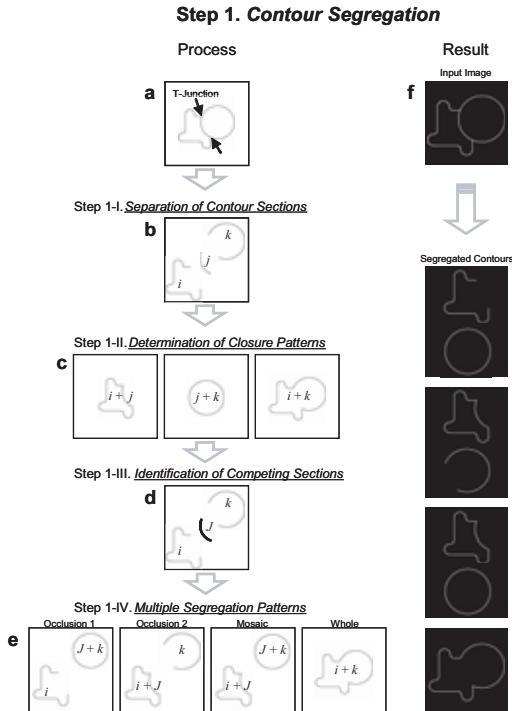


Fig. 2. Schematic view of Step 1, the segregation of contours (a-e), and an example of one possible final result of the process (f)

colleagues showed that each V4 neuron responds to a particular macroscopic feature of curvature, regardless of the whole shape [6], and demonstrated that a certain shape can be represented by the population activity of these curvature neurons [7].

Furthermore, they discovered that the responses of V4 curvature cells were also modulated by adjacent macro features of curvature [6]. Surprisingly, there were two types of modulation. That is, the responses of one type of curvature cells were modulated by curvature features in the clockwise (CW) direction and the responses of the other type of curvature cells were modulated by curvature features in the counterclockwise (CCW) direction. These findings suggest that macroscopic features of an arbitrary shape are represented in a multi-dimensional feature space, including a two-dimensional subspace of curvature and adjacent-curvature.

By taking advantage of this 2D subspace, it seems possible to detect or evaluate the global constellation of features of a shape, especially its symmetry. For example, the detection of rotational symmetry seems possible by evaluating a sort of informational compression of shape representation in the 2D subspace. Additionally, mirror symmetry can be detected as an identical feature map between the CW and CCW types. In support of this idea, area V4 and its vicinity were highly activated by the presentation of symmetrical visual stimuli [8].

Based on the physiological findings above, here, we present a novel neurocomputational model that enables global amodal completion through the evaluation of a constellation of features, each detected as a curvature segment within a shape contour.

2 Method

2.1 Overview of the Model

In amodal completion, two kinds of ambiguity exist (Fig. 1), ambiguity in segregation (Fig. 1b) and ambiguity in completion (Fig. 1c). These ambiguities are reflected in the flow of processes performed in our computational model.

Before Step 1, in the visual system local line-segments of shape contour and their orientation need to be detected. This process corresponds to the task performed by the orientation-selective cells in visual area V1 (V1). For simplicity, the detection of luminance borders was omitted from the model.

In Step 1, the contour sections were segregated through the detection of T junctions (Fig. 2). In Step 2, macro curvature segments along a shape contour were detected by voting each local line segment (Fig. 3). In Step 3 (Fig. 4), the detected curvature segments were mapped onto a full feature space including the 2D curvature- adjacent curvature (C-AC) subspace in the CW and CCW directions. By using the two C-AC subspaces of CW and CCW, global regularities, in particular the rotational or mirror symmetry of a shape, could be used to extrapolate the occluded part of a shapefs contours.

Step 2. Detection of Macro Curvature Segments

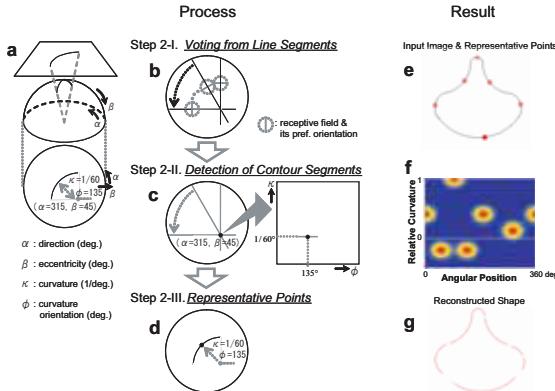


Fig. 3. Step 2, detection of macro curvature segments. (a) The process is executed for a spherically projected image. (b-d) The detection process. (e-g) The detected curvature segments and the shape reconstructed from these segments for a contour similar to the one used in [7].

2.2 Contour Segregation

Figure 2a-e shows a schematic of the sub-steps of the contour-segregation process (Step 1). First, T-junctions within the presented image were detected (Fig. 2a). These T-junctions set the origins of each contour section of the shape (Step 1-I). For the example in Figure 2, the shape was divided into three contour sections, i, j, and k (Fig. 2b). By combining these sections, possible closure patterns or closed areas could be determined (Step 1-II; Fig. 2c).

In Step 1-III, sections that resulted in the closure of multiple tangential areas were demarcated as ambiguous (in Fig. 2d, the ambiguous sections are labeled gJh). Closure was determined by calculating the exterior angle along a contour section. When a contour was closed, the integral was 360 degrees. Competition was determined by summatting the integral of the exterior angles calculated for each direction along a section. If the contour section was a competing one, that is, it was involved in the generation of two closed areas, the accumulative integral of the exterior angles for the section contributed in an opposite way in calculating the accumulative integral for the two closed complete shapes. In the case shown in Figure 2c, section gjh has a negative exterior angle in one direction (shown in the left panel of Fig. 2c), while it has a positive exterior angle in the other direction (shown in the middle panel of Fig. 2c). In contrast, sections gih and ghk (shown in Fig. 2c left-right and middle-right panels, respectively), contributed in the same way in calculating the accumulative integral for the two closed complete shapes.

Finally, possible segregation patterns were determined. For the example shown in Figures 1b and 2e (Step 1-IV), there were four possible segregation patterns, a relevant occlusion pattern (Occlusion 1), a less relevant occlusion pattern (Occlusion 2), a pattern in which two areas have contact (Mosaic), and a pattern

in which the above-mentioned competing section was interpreted as a gcrackh (Whole). That is, the competing section was assigned to a single area in Occlusion 1 and 2, while it was assigned to two areas or no area in the Mosaic and Whole patterns, respectively.

2.3 Detection of Macro Curvature Segments

Macro curvature segments of contour were detected by voting from each local line segment (Step 2, Fig. 3) [9]. This process was done via a spherically-projected image (Fig. 3a) so as to detect macro line segments, namely segments with zero-curvature, in the same way as other curvature segments, by taking advantage of the fact that a line in Cartesian space corresponds to a part of a great circle on a sphere.

Each point of the sphere, (α, β) had two more dimensions, orientation ϕ and curvature κ . Voting was done for each local line-segment along its normal line on the sphere (Step 2-I, Fig. 3b). That is, when a point (α_0, β_0) on the normal line of a local line segment of interest with orientation ϕ_0 was $1/\kappa_0$ away from the local line segment, a vote was casted for $(\alpha_0, \beta_0, \phi_0, \kappa_0)$.

A macro curvature segment was detected as a peak, namely a point that obtained many votes (Step 2-II, Fig. 3c). If the point $(\alpha_0, \beta_0, \kappa_0)$ had a peak in the vote, the orientations of the normal lines were averaged so as to obtain the orientation of the detected macro curvature segment. The orientation of a curvature segment was defined as the orientation of the tangential line at the midpoint of the macro curvature segment. We then correlated the detected curvature and its orientation with the midpoint on the contour as a representative point (Step 2-III, Fig. 3d).

2.4 Contour Completion

A shape was represented by its constituent curvature segments and the spatial relationships of these segments. Namely, detected curvature segments within a shape contour were mapped onto feature spaces (Step 3-I). The model had two feature spaces, a CW one and a CCW one. Each feature space was composed not only of the above-mentioned curvature κ and its orientation ϕ , but also of an adjacent curvature κ , a relative angle between representative points ψ , a length between representative points l and votes n (Fig. 3a). We also obtained the angular position of a contour segment as the phase within a contour (2π corresponds to the full circle of the contour). This angular position was used to show good correspondence with the physiological data. However, for completion by the model, we did not use the angular position but instead used relative angle between representative points ψ and length between representative points l for convenience.

The process of the global completion based on rotational symmetry is schematized in Figure 3b. The rotational symmetry for an incomplete contour was judged by the representational compression in the C-AC subspace (Step 3-IIr). That is, if the ratio between the number of curvature-segment features in the

C-AC subspace and the entire number of the segments within the contour was less than 0.5, the incomplete contour was considered to have rotational symmetry. If an incomplete contour was considered to have rotational symmetry, then the features of the occluded part of the contour were extrapolated. That is, each quantity of a curvature-segment feature was copied appropriately (Step 3-III), so as to complete the contour (Step 3-IV). Completion was assessed easily by integrating the exterior angles of a segment.

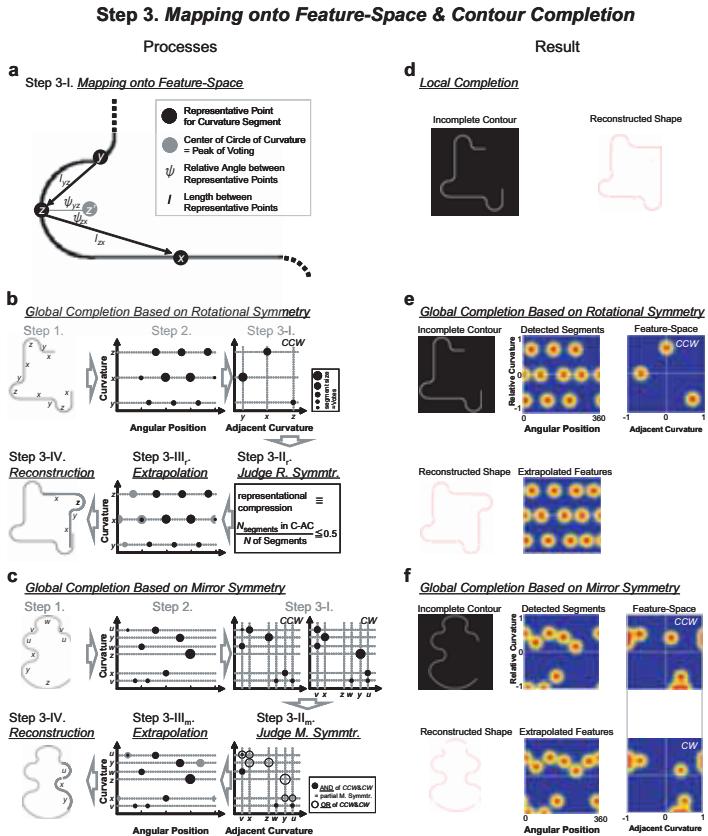


Fig. 4. Step 3, mapping segments onto the feature space and contour completion. (a) Features related to the spatial relationship between contour segments. The global completion processes based on rotational (b) and mirror symmetry (c). Results of the completion processes based on local (d), rotational (e) and mirror symmetry (f).

The process for mirror symmetry is shown in Figure 3c. Mirror symmetry was determined by the existence of common features in the A-AC subspace between the CW and CCW feature spaces (Step 3-II_m). If the incomplete contour was determined to have mirror symmetry, the features in the A-AC subspace of CW

and that of CCW were merged. Then, the occluded features were extrapolated or copied (Step 3-III). Finally, the occluded contour was reconstructed (Step 3-IV).

3 Results

Figure 2f shows an example of the contour segregation process (Step 1). For convenience, input images included line drawings of the shape. This step succeeded in providing multiple segregation patterns (shown in Fig. 2e).

Figures 3e and f show the curvature segments detected (in Step 2) for an input shape similar to the one in Figure 2 of Pasupathy & Connor (2002) [7]. The spots on the contour in Figure 3e indicate representative points of each curvature segment. The relative curvature and angular positions of each segment are indicated in Figure 3f, which shows a good correspondence between the representation of the shape by the model and V4 curvature neurons (presented in Fig. 2b of [7]). As mentioned in the Methods (2.4), the input shape was successfully reconstructed by our model from detected curvatures that were then represented in full feature space (Fig. 3g).

The results of Step 3 are shown in Figures 4d-e. Figure 4d shows an example of local completion. The incomplete contour (left) was successfully completed by the model (right). Failure to complete the input shape could be readily detected by calculating the accumulative integral of the exterior angles for each segment. If input shapes were not completed, completion was forced by elongating the two terminal segments until the integral of the exterior angles was 360 degrees. For the same incomplete contour as that shown in Figure 4d, global completion could also be achieved based on an evaluation of rotational symmetry (Fig. 4e). The rotational symmetry of the incomplete input contour was determined by the comparison of the number of features found in the A-AC subspace with the number of detected curvature segments. This evaluation successfully led to feature extrapolation and shape reconstruction. The ability to achieve global completion based on the evaluation of mirror symmetry was also tested. For the example shown in Figure 4f, there were several identical points in the C-AC subspaces of the CW and CCW feature spaces, as can be recognized by comparison between the two rightmost panels in Figure 4f. The incomplete shape, shown in the top, leftmost panel in Figure 4f, could be completed using mirror symmetry to form the whole shape, shown in the bottom, leftmost panel in Figure 4f.

4 Discussion

We have briefly outlined a computational model that enables global amodal completion, based on the characteristics of neurons in visual area V4. In particular, the model succeeded not only in segregating shape contours containing ambiguous contour sections that could occlude the proper reconstruction of the shape, but also in completing these contours correctly, both locally and globally.

The detection of macro curvature segments by our model was consistent with the physiological data for how V4 neurons respond to macroscopic features of curvature in shape contours [7][8]. Other aspects of our model were implemented in accordance with recent physiological studies. In our model, the evaluation of the representation of detected curvature segments in C-AC subspace detected the presence of symmetry. This success of this methodology seems to correspond to the fact that area V4 and area PIT (posterior inferotemporal), a target of V4 along the occipitotemporal pathway, were activated when symmetrical images were presented [8]. Furthermore, the mirror image confusion in single neurons in the upstream inferotemporal cortex [10] might be easily understood as an inappropriate integration of shape features represented in C-AC subspace between the CW and CCW feature spaces. On the other hand, shapes with occlusion segments were successfully reconstructed both locally and globally in parallel in our model. This aspect of our model is consistent with a human physiological study showing that in area LOC (lateral occipital complex), located upstream of V4, the right LOC was activated by globally completed shapes, whereas the left LOC was activated by locally completed shapes [11].

The computational model in this paper presented a representation and algorithm that enables both local and global amodal completion [12]. The implementation of the principles set forth in this model in a physiologically plausible way is an important future goal.

References

1. Sajda, P., Finkel, L.H.: Intermediate-level visual representations and the construction of surface perception. *J. Cogn. Neurosci.* 7, 267–291 (1995)
2. Fukushima, K.: Neural network model for completing occluded contours. *Neural Networks* 23, 528–540 (2010)
3. Sekuler, A.B., Palmer, S.E., Flynn, C.: Local and global processes in visual completion. *Psychol. Sci.* 5, 260–267 (1994)
4. van Lier, R., van der Helm, P., Leeuwenberg, E.: Integrating global and local aspects of visual occlusion. *Perception* 23, 883–903 (1994)
5. Pasupathy, A., Connor, C.E.: Responses to contour features in macaque area V4. *J. Neurophysiol.* 82, 2490–2502 (1999)
6. Pasupathy, A., Connor, C.E.: Shape representation in area V4: position-specific tuning for boundary conformation. *J. Neurophysiol.* 86, 2505–2519 (2001)
7. Pasupathy, A., Connor, C.E.: Population coding for shape in area V4. *Nat. Neurosci.* 5, 1332–1338 (2002)
8. Sasaki, Y., Vanduffel, W., Knutson, T., Tyler, C., Tootell, R.: Symmetry activates extrastriate visual cortex in human and nonhuman primates. *Natl. Acad. Sci. USA* 102, 3159–3163 (2005)
9. Hashimoto, M.: A research of shape representation in higher visual cortex. Tohoku Univ. master thesis (2009)
10. Rollenhagen, J.E., Olson, C.R.: Mirror image confusion in single neurons of the macaque inferotemporal cortex. *Science* 287, 1506–1508 (2000)
11. de Wit, T.C., Bauer, M., Oostenveld, R., Fries, P., van Lier, R.: Cortical responses to contextual influences in amodal completion. *Neuroimage* 32, 1815–1825 (2006)
12. Marr, D.: Vision. MIT Press, Cambridge (1982)

Quantitative Modeling of Neuronal Dynamics in *C. elegans*

Masahiro Kuramochi and Yuishi Iwasaki

Department of Intelligent System Engineering, Ibaraki University

Hitachi, Ibaraki 316-8511, Japan

09nm917x@hcs.ibaraki.ac.jp, iwasaki@mx.ibaraki.ac.jp

Abstract. We present a mathematical model to quantitatively describe the neuronal dynamics in *Caenorhabditis elegans*. Since calcium imaging is a popular technique to visualize the neuronal activity in *C. elegans*, the model includes the variable of the fluorescence intensity in addition to the membrane potential and the intracellular calcium concentration. The fluorescence intensity is a quantity which is comparable with the experimental data. The parameters in the model are determined to reproduce the neurophysiological experimental data. Our model exhibits good agreement with the data. We apply the model to a neural circuit for chemotaxis and find that the neuronal activity measured by the fluorescence intensity shows quantitatively different behavior from that measured by the membrane potential in some neurons. The difference is discussed from the viewpoint of neuronal mechanisms.

Keywords: *C. elegans*, membrane potential, intracellular calcium concentration, fluorescence intensity, chemotaxis circuit.

1 Introduction

The nematode *Caenorhabditis elegans* is a useful model organism both for neuronal study and for behavioral study. The synaptic connectivity is entirely determined by the anatomical work [1]. Although the nervous system consists of only 302 neurons, *C. elegans* shows various behaviors such as withdrawal reflex, chemotaxis and thermotaxis. To understand these behaviors from the viewpoint of neural circuit, the neuronal activity needs to be measured. Since *C. elegans* is small, about 1 mm in length, electrophysiological measurement of the membrane potential is difficult. Instead of the measurement, calcium imaging techniques using fluorescent proteins have been developed to visualize the neuronal activity. Since *C. elegans* is transparent, optical recording can be done without dissection. Experimental data of the calcium imaging is available to quantitatively describe the neuronal dynamics in *C. elegans* as the worm is alive.

Here, we should pay attention to the experimental data of the calcium imaging. Quantity to be measured in the calcium imaging is not the intracellular calcium concentration itself but the fluorescence intensity [2]. Since a fluorescent probe interacts with intracellular proteins and changes its fluorescence properties, it is difficult to evaluate the exact intracellular calcium concentration from

the fluorescence intensity. In addition to the neuronal dynamics [3], therefore, the dynamics of the fluorescence intensity is considered in our model (section 2.2 for the details). Our model includes variables of the membrane potential, the intracellular calcium concentration and fluorescence intensity. The dynamics of these variables can be comparable with the corresponding experimental data. For a single neuron, the result of our model is compared with that of the electrophysiological experiment [4] in section 3.1.

We apply the model to a neural circuit for chemotaxis in *C. elegans*. The worm moves toward or away from various chemicals. In this work, the attractive response to NaCl (salt) is chosen because of two reasons. One is that the chemosensory neurons for NaCl are identified. The other is that behavior analysis using NaCl is intensively performed to understand how the worm senses and migrates up a gradient of the chemical attractant [5]. From our numerical study on the neural circuit for chemotaxis, we find that the neuronal activity measured by the fluorescence intensity shows quantitatively different behavior from that measured by the membrane potential in some neurons. To understand the difference, we discuss each influences of the ion channels and the synaptic connections on the neuronal dynamics.

2 Modeling

2.1 Neuronal Dynamics

Voltage-gated sodium channels have not been identified in *C. elegans* [4]. Instead of sodium-based classical action potentials, neurons might have calcium-based signal amplification as in the large nematode *Ascaris*. Therefore we consider the neuronal mechanisms shown in Fig. 1. The parameters in the following neuronal model have the same values in the referenced papers or are simply decreased in proportion to the cell size of *C. elegans*. In order to well reproduce the neurophysiological experimental data, furthermore, some parameters are manually fine-tuned from these values (section 3.1 for the details).

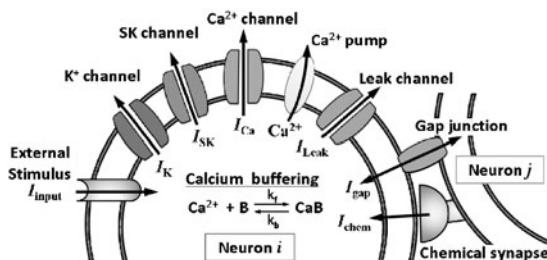


Fig. 1. Schematic illustration of neuronal mechanisms in our model. A neuron is connected to other neurons through chemical synapses and gap junctions.

The membrane potential $V^{(i)}$ of the i th neuron is determined by

$$C_m \frac{dV^{(i)}}{dt} = - \left(I_K^{(i)} + I_{Ca}^{(i)} + I_{SK}^{(i)} + I_{Leak}^{(i)} \right) + I_{chem}^{(i)} + I_{gap}^{(i)} + I_{input}^{(i)}, \quad (1)$$

where the membrane capacitance is $C_m = 38.5 \text{ } \mu\text{F}/\text{cm}^2$ for *C. elegans*. The $I_K^{(i)}$, $I_{Ca}^{(i)}$, $I_{SK}^{(i)}$ and $I_{Leak}^{(i)}$ are the voltage-sensitive potassium current, the voltage-sensitive calcium current, the calcium-sensitive potassium current and the leakage current, respectively. The $I_{chem}^{(i)}$ and $I_{gap}^{(i)}$ are currents from other neurons through chemical synapses and gap junctions, respectively. The $I_{input}^{(i)}$ is an input current for a sensory neuron as an external stimulus.

Currents through the potassium and calcium channels are given by the Hodgkin-Huxley type equations. The voltage-sensitive potassium current is given by $I_K^{(i)} = G_K(m_K^{(i)})^3(V^{(i)} - E_K)$ with $G_K = 9.75 \text{ mS}/\text{cm}^2$ and $E_K = -81 \text{ mV}$. The voltage-sensitive calcium current $I_{Ca}^{(i)}$ is expressed in the following form using the Goldman-Hodgkin-Katz equation.

$$I_{Ca}^{(i)} = \left(m_{Ca}^{(i)} \right)^2 P_{max} V^{(i)} \frac{[Ca^{2+}]_{in}^{(i)} - [Ca^{2+}]_{out} \exp(-2FV^{(i)}/RT)}{1 - \exp(-2FV^{(i)}/RT)}, \quad (2)$$

where $P_{max} = 0.006 \text{ } \mu\text{A}/(\mu\text{M}\cdot\text{mV}\cdot\text{cm}^2)$, $[Ca^{2+}]_{out} = 2.1 \text{ mM}$ and $T = 293.1 \text{ K}$. The F and R are the Faraday constant and the gas constant, respectively. The kinetics for the gating variables $m_K^{(i)}$ and $m_{Ca}^{(i)}$ are described as $dm_x^{(i)}/dt = \alpha_x(V^{(i)})(1-m_x^{(i)}) - \beta_x(V^{(i)})m_x^{(i)}$ where the index $x = \text{K or Ca}$. The rate functions $\alpha_x(\cdot)$ and $\beta_x(\cdot)$ have sigmoidal shape dependencies on the membrane potential.

The small-conductance calcium-gated potassium channel (SK channel) is a fundamental regulator of excitability. The SK channel is activated solely by intracellular calcium ions and therefore may function at rest or at hyperpolarized potentials. When the SK channel exists, a change in intracellular calcium exerts a strong effect on neuronal excitability. The calcium-sensitive potassium current is given by $I_{SK}^{(i)} = G_{SK}m_{SK}^{(i)}(V^{(i)} - E_{SK})$ with $G_{SK} = 0.2125 \text{ mS}/\text{cm}^2$ and $E_{SK} = -81 \text{ mV}$. The kinetics for the gating variables $m_{SK}^{(i)}$ are described as $dm_{SK}^{(i)}/dt = (m_{SK}^{(i)}([Ca^{2+}]_{in}^{(i)}) - m_{SK}^{(i)})/\tau_m^{(i)}([Ca^{2+}]_{in}^{(i)})$ where $m_\infty^{(i)}([Ca^{2+}]_{in}^{(i)}) = [Ca^{2+}]_{in}^{(i)} / ([Ca^{2+}]_{in}^{(i)} + K_{SK})$ and $\tau_m^{(i)}([Ca^{2+}]_{in}^{(i)}) = \psi_{SK} / ([Ca^{2+}]_{in}^{(i)} + K_{SK})$ with $K_{SK} = 0.4 \text{ } \mu\text{M}$ and $\psi_{SK} = 2.8 \text{ } \mu\text{M}\cdot\text{ms}$. The leakage current is given by $I_{Leak}^{(i)} = G_{Leak}(V^{(i)} - E_{Leak})$ with $G_{Leak} = 0.0525 \text{ mS}/\text{cm}^2$ and $E_{Leak} = -74 \text{ mV}$.

For chemical synapse, we assume that synaptic activation and transmission are graded and sigmoidally shaped with presynaptic potential [6].

$$I_{chem}^{(i)} = \sum_j n_{chem}^{(j,i)} \frac{G_{chem} \left(E_{chem}^{(j,i)} - V^{(i)} \right)}{1 + \exp(- (V^{(j)} - V_{shift}) / V_{range})}, \quad (3)$$

where $G_{chem} = 0.3 \text{ mS}/\text{cm}^2$, $V_{range} = -35/2 \log(0.1/0.9) \text{ mV}$ and $V_{shift} = -10 \text{ mV}$. The $n_{chem}^{(j,i)}$ is the total number of chemical synapses from the j th neuron to

the i th neuron. The reversal potential is $E_{\text{chem}}^{(j,i)} = 0$ mV for an excitatory synapse or $E_{\text{chem}}^{(j,i)} = -48$ mV for an inhibitory synapse, respectively. Gap junction is modeled as a ohmic resistance. That is, current flows from j th neuron to i th neuron in proportion to a difference between the membrane potentials.

$$I_{\text{gap}}^{(i)} = \sum_j n_{\text{gap}}^{(j,i)} G_{\text{gap}} \left(V^{(j)} - V^{(i)} \right), \quad (4)$$

where $G_{\text{gap}} = 1.02$ mS/cm². The $n_{\text{gap}}^{(j,i)}$ is the total number of gap junctions between the neurons. In *C. elegans*, $n_{\text{chem}}^{(j,i)}$ and $n_{\text{gap}}^{(j,i)}$ are determined for any pair of the neurons [1].

As shown in Fig. 1, we consider three mechanisms to cause a change in the intracellular calcium concentration. The first mechanism is Ca²⁺ inflow $I_{\text{Ca}}^{(i)}$ from the extracellular fluid through the calcium channel. The second mechanism is a chemical reaction with the calcium binding buffer proteins. In this work, we assume a plenty of the buffer proteins [B] is constant. The third mechanism is the calcium pump which carries Ca²⁺ out of the cell across the membrane. Therefore the intracellular calcium concentration $[\text{Ca}^{2+}]_{\text{in}}^{(i)}$ and the ratio of the buffer molecules occupied by calcium ions $[\text{CaB}]^{(i)}$ are determined by the following equations [2].

$$\begin{aligned} \frac{d[\text{Ca}^{2+}]_{\text{in}}^{(i)}}{dt} &= -\frac{1}{2Fd^{(i)}} I_{\text{Ca}}^{(i)} + k_b[B][\text{CaB}]^{(i)} - k_f[\text{Ca}^{2+}]_{\text{in}}^{(i)}[B] \left(1 - [\text{CaB}]^{(i)} \right) \\ &\quad - \frac{G_{\text{pump}}[\text{Ca}^{2+}]_{\text{in}}^{(i)}}{[\text{Ca}^{2+}]_{\text{in}}^{(i)} + K_{\text{pump}}}, \end{aligned} \quad (5)$$

$$\frac{d[\text{CaB}]^{(i)}}{dt} = -k_b[\text{CaB}]^{(i)} + k_f[\text{Ca}^{2+}]_{\text{in}}^{(i)} \left(1 - [\text{CaB}]^{(i)} \right), \quad (6)$$

where $k_f = 0.1$ (ms·μM)⁻¹, $k_b = 0.3$ ms⁻¹, $G_{\text{pump}} = 3.6$ μM/ms, $K_{\text{pump}} = 0.75$ μM and $[B] = 30$ μM. The $d^{(i)}$ is the Ca²⁺ diffusion length through the membrane.

2.2 Fluorescence Intensity

In this study, an indicator in the calcium imaging is the yellow "cameleon" YC3.60 which is a fluorescent protein often used in *C. elegans*. The YC3.60 indicator reacts with intracellular Ca²⁺ and is luminiferous in a neuron. The fluorescence intensity $Y^{(i)}$ is calculated from the intracellular calcium concentration as follows.

$$Y^{(i)} = \frac{\left([\text{Ca}^{2+}]_{\text{in}}^{(i)} / K_{\text{YC3.60}} \right)^\eta R_{\text{max}} + R_{\text{min}}}{1 + \left([\text{Ca}^{2+}]_{\text{in}}^{(i)} / K_{\text{YC3.60}} \right)^\eta}, \quad (7)$$

where $K_{\text{YC3.60}} = 250$ nM and $\eta = 1.7$. The maximum fluorescence intensity is $R_{\text{max}} = 9.3$ and the minimum fluorescence intensity is $R_{\text{min}} = 1.4$ [2]

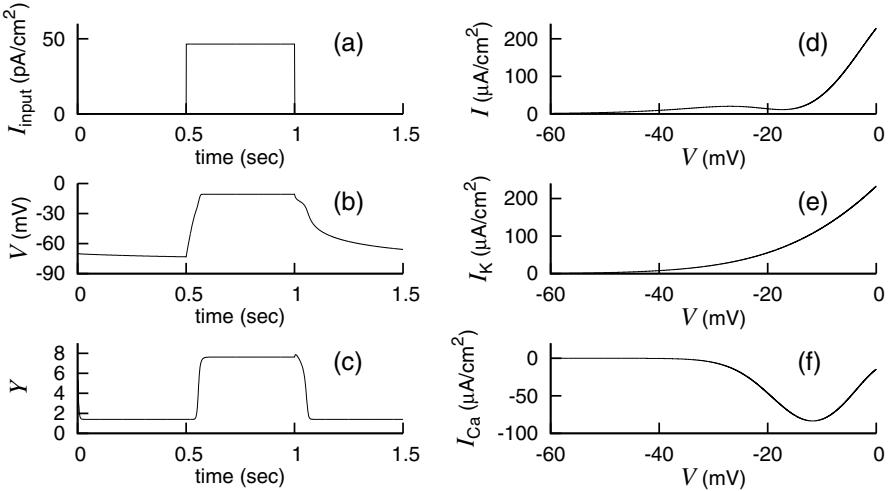


Fig. 2. Neuronal responses of the single neuron to current injections. (a) Input current. (b) Voltage response to current injection. (c) Time course of the fluorescence intensity. (d) Current-voltage relation. (e) The voltage-sensitive potassium current I_K to the voltage response. (f) The voltage-sensitive calcium current I_{Ca} to the voltage response.

The dynamical variables in our model are the physical quantities $V^{(i)}$, $[\text{Ca}^{2+}]_{\text{in}}^{(i)}$, $[\text{CaB}]^{(i)}$, the fluorescence intensities $Y^{(i)}$ and the gating variables $m_K^{(i)}$, $m_{\text{Ca}}^{(i)}$, $m_{\text{SK}}^{(i)}$ in the Hodgkin-Huxley formalism. Especially, $V^{(i)}$ and $Y^{(i)}$ are the observable variables which are quantitatively comparable with the experimental data in *C. elegans*.

3 Results

3.1 Electrophysiology of a Single Neuron

For some neurons in *C. elegans*, the electrophysiological experiment is performed to investigate the electrical property of a single neuron. Unlike in mammals, the neuron in *C. elegans* generates a graded potential in response to current injection. To determine the model parameters shown in the previous section, we use the voltage responses of ASE chemosensory neurons to current injections, whose recordings are measured in patch clamp technique [4]. Results of our model are shown in Fig. 2. The voltage response (Fig. 2b) and the current-voltage relation (Fig. 2d) agree well with the corresponding experimental data in Ref. [4].

Figure 2c indicates that the fluorescence intensity accurately follows stepwise change in the input current than the membrane potential follows. For ASE neurons, there is no corresponding experimental data to I_K (Fig. 2e) and I_{Ca} (Fig. 2f). For the muscle cells in *C. elegans*, however, the current-voltage relations of I_K and I_{Ca} are experimentally obtained [89]. Comparing with these experimental results, Fig. 2e and Fig. 2f show the physiologically adequate relations.

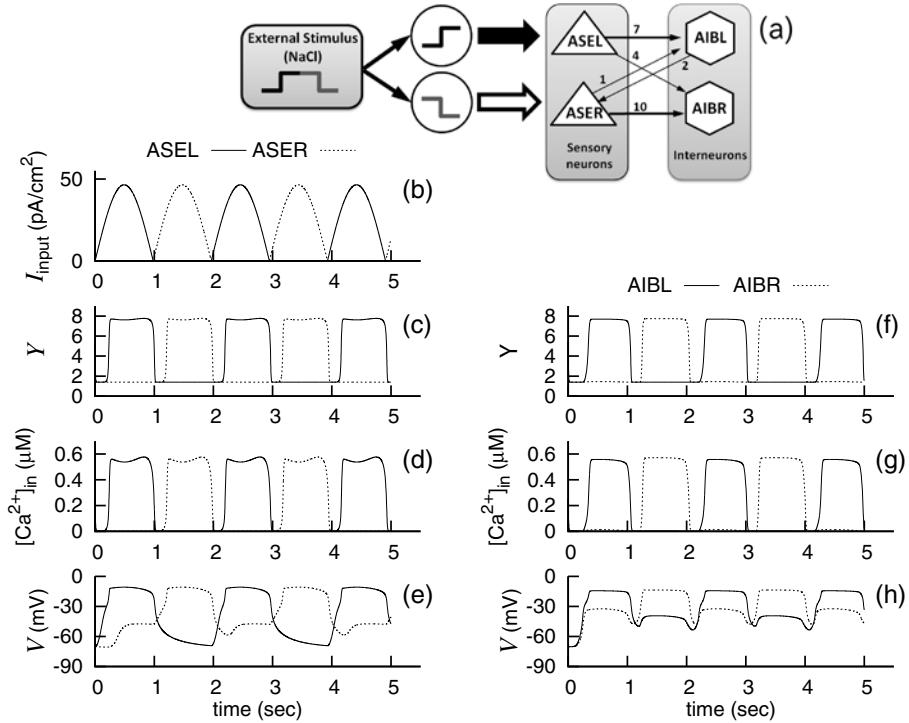


Fig. 3. Neuronal responses in the chemotaxis circuit. (a) Schematic illustration of sensory inputs to the chemotaxis circuit. (b) The current injections to ASEL/R chemosensory neurons. Time course of the fluorescence intensity, the intracellular calcium concentration and the membrane potential of ASEL/R neurons are shown in (c), (d) and (e), respectively. Those of AIBL/R neurons are shown in (f), (g) and (h), respectively.

3.2 Neural Circuit for Chemotaxis

In this section, we apply the model to a neural circuit for NaCl chemotaxis shown in Fig. 3a. In *C. elegans*, the sensory neurons for NaCl are ASEL/R. Here, ASEL/R chemosensory neurons exhibit the left/right asymmetric activity [10]. The ASEL is stimulated by increases in NaCl concentration (ON-sensing cell). On the other hand, ASER is stimulated by decreases in NaCl concentration (OFF-sensing cell). This asymmetric stimulations are taken into account as the input currents for ASEL/R chemosensory neuron. In this work, we assume the periodic $I_{\text{input}}^{(i)}$ shown in Fig. 3b under the following conditions. (i) The worm runs sinusoidally on the surface of an agar plate at the speed of 0.12 mm/s. (ii) A period of head swing of the worm is 1.96 s. (iii) The worm responds to a spatial gradient of NaCl and gradually curve toward higher concentration of NaCl [5]. (iv) The input current $I_{\text{input}}^{(i)}$ is proportional to change in the NaCl gradient. The ASEL/R neurons are connected to other 11 sensory neurons and 10 interneurons through chemical synapses [11]. There is no gap junction in ASEL/R neurons. To

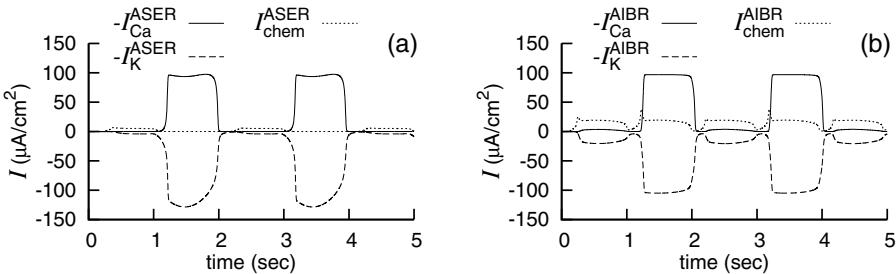


Fig. 4. The voltage-sensitive calcium current, the voltage-sensitive potassium current and the chemical synaptic current. (a) The currents of ASER neuron. (b) The currents of AIBR neuron.

reduce the degree of freedom in neuronal modeling, only AIBL/R interneurons are considered in this work. It is reported that AIBL/R interneurons are associated with chemotaxis behavior [5]. Here, a chemical synapse is characterized by the synaptic "sign"; whether a chemical synapse is functionally excitatory or inhibitory. However, little synaptic sign of *C. elegans* is identified [6][1]. We assumed that chemical synapses of ASELR neurons are excitatory, on the other hand, those of AIBL/R neurons are inhibitory.

The neuronal responses in the chemotaxis circuit is shown in Fig. 3. For the continuous chemical stimuli, we find that $Y^{(i)}$ and $[Ca^{2+}]_{in}^{(i)}$ change stepwisely while $V^{(i)}$ changes gradually. These features arise from the threshold dynamics of $I_{Ca}^{(i)}$. The $[Ca^{2+}]_{in}^{(i)}$ behaves stepwisely since $I_{Ca}^{(i)}$ in the right-hand side of Eqn. (5) is highly sensitive to $V^{(i)}$ within a given narrow range (figure not shown in this paper). The $I_{Ca}^{(i)}$ plays a key role to the neuronal activity [12].

To investigate the influences of the ion channels and the chemical synapses on $V^{(i)}$, the currents $I_{Ca}^{(i)}$, $I_K^{(i)}$ and $I_{chem}^{(i)}$ of ASER and AIBR neurons are shown in Fig. 4. The $I_{SK}^{(i)}$ and $I_{Leak}^{(i)}$ are relatively smaller than these currents. The $I_K^{(i)}$ (hyperpolarization) works properly as a counter term for $I_{Ca}^{(i)}$ (depolarization). In Fig. 4b, I_{chem}^{AIBR} from ASELR and I_{chem}^{AIBR} from ASER are mutually replaced and change periodically (divided plots not shown in this paper). The I_{chem}^{AIBR} from ASER (a change at $t = 1.2$ s in Fig. 4b for example) is slightly higher than I_{chem}^{AIBR} from ASELR (a change at $t = 2.2$ s in Fig. 4b for sample). For V^{AIBR} in Fig. 3h, this small difference in the input current yields only a small difference in the membrane potential. For $[Ca^{2+}]_{in}^{AIBR}$ in Fig. 3g, however, the difference causes a stepwise change ("all" or "none") in the intracellular calcium concentration because of the threshold dynamics of $I_{Ca}^{(i)}$.

4 Discussion

In *C. elegans*, electrophysiological studies at the cellular level have been restricted because of the difficulty to dissect its small body. Therefore little is

known about electrical properties of the neurons. Our neuronal model has the following advantages. (i) The model gives a quantitative relation between the dynamics of the membrane potential and the dynamics of the fluorescence intensity. (ii) Result of the model is comparable with the experimentally observed data and might complement the unobserved data to understand the neuronal mechanism. (iii) Influences of the ion channels and the synaptic connections on the neuronal dynamics are able to be investigated. Since the neural circuit used in this work is small, our results should be confirmed in a large neural circuit. There is no guarantee that the influence of the remaining neurons not shown in Fig. 3a is negligible [11]. Study on the neuronal dynamics using a large neural circuit is left for our future work. It is difficult to predict the synaptic signs in a large neural circuit. Our final goal is to provide a mathematical model which has predictive ability to forecast the neuronal dynamics.

This research was supported by the Grant-in-Aid for Scientific Research on Innovative Areas "Systems Molecular Ethology" (No. 20115004) from MEXT.

References

1. White, J.G., Soughgate, E., Thomson, J.N., Brenner, S.: The structure of the nervous system of the nematode *Caenorhabditis elegans*. Phil. Trans. R. Soc. Lond. Biol. 314, 1–340 (1986)
2. Nagai, T., Yamada, S., Tominaga, T., Ichikawa, M., Miyawaki, A.: Expanded dynamic range of fluorescent indicators for Ca^{2+} by circularly permuted yellow fluorescent proteins. Proc. Natl. Acad. Sci. USA 101, 10554–10559 (2004)
3. Sakata, K., Shingai, R.: Neural network model to generate head swing in locomotion of *Caenorhabditis elegans*. Network: Comput. Neural Syst. 15, 199–216 (2004)
4. Goodman, M.B., Hall, D.H., Avery, L., Lockery, S.R.: Active currents regulate sensitivity and dynamic range in *C. elegans* neurons. Neuron 20, 763–772 (1998)
5. Iino, Y., Yoshida, K.: Parallel use of two behavioral mechanisms for chemotaxis in *Caenorhabditis elegans*. J. Neurosci. 29, 5370–5380 (2009)
6. Wicks, S.R., Roehrig, C.J., Rankin, C.H.: A dynamic network simulation of the nematoda tap withdrawal circuit: predictions concerning synaptic function using behavioral criteria. J. Neurosci. 16, 4017–4031 (1996)
7. Aoyagi, T., Kang, Y., Terada, N., Kaneko, T., Fukai, T.: The role of Ca^{2+} -dependent cationic current in generating gamma frequency rhythmic bursts: modeling study. Neurosci. 115, 1127–1138 (2002)
8. Jospin, M., Mariol, M.-C., Ségalat, L., Allard, B.: Characterization of K^+ currents using an in situ patch clamp technique in body wall muscle cells from *Caenorhabditis elegans*. J. Physiol. 544, 373–384 (2002)
9. Jospin, M., Jacquemond, V., Mariol, M.-C., Ségalat, L., Allard, B.: The L-type voltage-dependent Ca^{2+} channel EGL-19 controls body wall muscle function in *Caenorhabditis elegans*. J. Cell Biol. 159, 337–347 (2002)
10. Suzuki, H., Thiele, T.R., Faumont, S., Ezcurra, M., Lockery, S.R., Schafer, W.R.: Functional asymmetry in *Caenorhabditis elegans* taste neurons and its computational role in chemotaxis. Nature 454, 114–117 (2008)
11. Iwasaki, Y., Gomi, S.: Stochastic formulation for a partial neural circuit of *C. elegans*. Bull. Math. Biol. 66, 727–743 (2004)
12. Mellem, J.E., Brockie, P.J., Madsen, D.M., Maricq, A.V.: Action potentials contribute to neuronal signaling in *C. elegans*. Nat. Neurosci. 11, 865–867 (2008)

Human Localization by Fuzzy Spiking Neural Network Based on Informationally Structured Space

Dalai Tang and Naoyuki Kubota

Dept. of System Design, Tokyo Metropolitan University,
6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan
tang@sd.tmu.ac.jp, kubota@tmu.ac.jp

Abstract. This paper analyzes the performance of the human localization by a spiking neural network in informationally structured space based on sensor networks. First, we discuss the importance of information structuralization. Next, we apply a spiking neural network to extract the human position in a room equipped with sensor network devices. Next, we propose how to update the base value as a method of preprocessing to generate input values to the spiking neurons, and the learning method of the spiking neural network based on the time series of measured data. Finally, we show several experimental results, and discuss the effectiveness of the proposed method.

Keywords: Robot Partners, Neural Networks, Sensor Networks, Human Localization.

1 Introduction

Nowadays, as the progress of network technologies, valuable and useful information can be easily acquired using web intelligence through wireless networks. Web intelligence plays an important role in data mining and structuralization of information based on artificial intelligence (AI) and others. However, such information required for users should be updated often according to the change of real situations such as disasters of earthquakes and fires, weather, traffic jams, climates, and terrorism. It is very important to gather the information on the user without the load of the user. The information service can be done efficiently, if the human location and behaviors can be extracted by environmental systems. Various types of concepts and technologies on ubiquitous computing, sensor networks, ambient intelligence, disappearing computing, intelligent spaces, and others have been proposed and developed to realize the information gathering, life support, safe and secure society [5, 6]. One of the most important things in the concepts and technologies is the structuralization of information. The structuralization of information is to give the qualitative meaning to data and quantitative information in order to improve the accessibility and usability of information. We can obtain huge size of data through sensor networks, but useful, meaningful and valuable information should be extracted from such data. We have proposed the concept of informationally structured space to realize the quick update and access of valuable and useful information for people and robots. The sensing range of both people and robot is limited. If the robot can obtain the exact position of

the robot itself, people, and objects in an environment, the robot does not need some sensors for measuring such information. As a result, the weight and size of a robot can be reduced much, since many sensors can be removed from robots. The received environmental information is more precise because the sensors equipped in the environment are designed suitable to the environmental conditions. Furthermore, if the robot can share the environmental information with people, the communication with people might become very smooth and natural. Therefore, we have proposed the method of human localization [13, 14], map building [15], and 3D visualization [16]. Various types of estimation method of human state have been proposed. We also applied a spiking neural network [12] to localize human position, and to learn pattern of daily life of elderly people. Most of the previous method is based on off-line statistic approaches after sensor data are obtained. In this paper, we analyze the performance of the on-line estimation method of human behaviors in the informationally structured space based on sensor networks through several experimental results, and discuss the effectiveness of the proposed method.

This paper is organized as follows. Section 2 explains the robot partners, data flow in the informationally structured space, spiking neural network for human localization. Section 3 shows experimental results of the proposed method. Section 4 summarizes the paper, and discusses the future vision of robot partners.

2 Information Support System Based on Informationally Structured Space

2.1 Informationally Structured Space

In the introduction we explain the necessity for the informationally structured space. Information resources and the accessibility within an environment are essential for both people and robots. Therefore, the environment surrounding people and robots should have a structured platform for gathering, storing, transforming, and providing information. Such an environment is called informationally structured space [16]. In the informationally structured space should gathering of real environment and cyber space, structuralization, visualization and display of the gathered information, and realizes the quick update and access of valuable and useful information for users. It is very useful for both robots and users to easily access the information on real environments.

The data flow in the developed system based on informationally structured space. The developed system is divided into five components; (1) database server, (2) robot systems; (3) sensor network system, (4) information support system, and (5) Web contents extraction System. The sensor network system is based on a ubiquitous wireless sensor network composed of sensors equipped with wall, floor, ceiling, furniture, and home appliances. These sensors measure the environmental data and human motions. The measured data are transmitted to the database server, and then feature extraction is performed. Each robot can receive the environmental information and Web exacted information from the database server, and serve as a partner to the person. Furthermore, the user interface device is used for the person to access the environmental information.

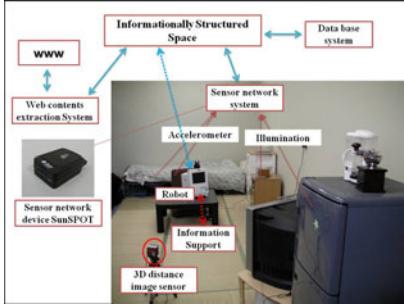


Fig. 1. The concept of Informationally Structured Space

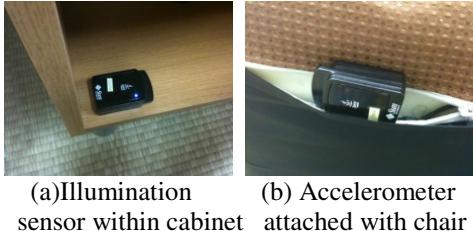


Fig. 2. Sensor network devices attached with furniture

In the sensor network system, we use a wireless sensor device called Sun SPOT (Sun Small Programmable Object Technology). The Sun SPOT include accelerometer, Illumination sensor and temperature sensor (Fig.1 , 2).

2.2 Spiking Neural Network for Human Localization

Memory is a mental function, but we can identify persistent states or patterns as the active or short-term memory. Furthermore, we can associate long-term memory with the connectivity of the network as a whole and the values of the synaptic parameters [20]. Basically, artificial neural networks are classified into pulse-coded neural networks and rate-coded neural networks from the viewpoint of abstraction level [18]. A pulse-coded neural network approximates the dynamics with the ignition phenomenon of a neuron, and the propagation mechanism of the pulse between neurons. An integrate-and-fire model with a first-order linear differential equation is known as a neuron model of a higher abstraction level. A spike response model is slightly more general than the integrate-and-fire model, because the spike response model can choose kernels arbitrarily. One important feature of pulse-coded neural networks is the capability of temporal coding. In fact, various types of spiking neural networks (SNNs) have been applied for memorizing spatial and temporal context. We use a simple spike response model to reduce the computational cost. First of all, the internal state $h_i(t)$ is calculated as follows;

$$h_i(t) = \tanh(h_i^{\text{syn}}(t) + h_i^{\text{ext}}(t) + h_i^{\text{ref}}(t)) \quad (1)$$

Here hyperbolic tangent is used to avoid the bursting of neuronal fires, $h_i^{\text{ext}}(t)$ is the input to the i th neuron from the external environment, and $h_i^{\text{syn}}(t)$ including the output pulses from other neurons is calculated by,

$$h_i^{\text{syn}}(t) = \gamma^{\text{syn}} \cdot h_i(t-1) + \sum_{j=1, j \neq i}^N w_{j,i} \cdot h_j^{\text{PSP}}(t) \quad (2)$$

Furthermore, $h_i^{\text{ref}}(t)$ indicates the refractoriness factor of the neuron; $w_{j,i}$ is the parameter of a weight coefficient from the j th to i th neuron; $h_j^{\text{PSP}}(t)$ is the presynaptic action potential (PSP) approximately transmitted from the j th neuron at the discrete

time t ; N is the number of neurons; r^{syn} is a temporal discount rate. When the internal action potential of the i th neuron is larger than the predefined threshold, a pulse is outputted as follows;

$$p_i(t) = \begin{cases} 1 & \text{if } h_i(t) \geq q_i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where q_i is a threshold for firing. At the time, the position of human position is estimated as the location of the furniture or equipment. Furthermore, R is subtracted from the refractoriness value in the following,

$$h_i^{ref}(t) = \begin{cases} \gamma^{ref} \cdot h_i^{ref}(t-1) - R & \text{if } p_i(t-1) = 1 \\ \gamma^{ref} \cdot h_i^{ref}(t-1) & \text{otherwise} \end{cases} \quad (4)$$

where γ^{ref} is a discount rate and $R > 0$. The presynaptic spike output is transmitted to the connected neuron according to PSP with the weight connection. The PSP is calculated as follows;

$$h_i^{PSP}(t) = \begin{cases} 1 & \text{if } p_i(t) = 1 \\ \gamma^{PSP} \cdot h_i^{PSP}(t-1) & \text{otherwise} \end{cases} \quad (5)$$

where γ^{PSP} is the discount rate ($0 < \gamma^{PSP} < 1.0$). Therefore, the postsynaptic action potential is excitatory if the weight parameter, $w_{j,i}$ is positive. If the condition $h_j^{PSP}(t-1) < h_i^{PSP}(t)$ is satisfied, the weight parameter is trained based on the temporal Hebbian learning rule as follows,

$$w_{j,i} \leftarrow \tanh(\gamma^{wgt} \cdot w_{j,i} + \xi^{wgt} \cdot h_j^{PSP}(t-1) \cdot h_i^{PSP}(t)) \quad (6)$$

where γ^{wgt} is a discount rate and ξ^{wgt} is a learning rate.

We apply the SNN to the human localization based on measured data of the sensor networks. Basically, each furniture or equipment is attached with a sensor. If the measured value of is changed large, then the difference from the base value is used as the inputs to a spiking neuron in the following;

$$h_i^{ext}(t) = \min \left\{ (v_i(t) - V_i)^2, 1 \right\} \quad (7)$$

where $v_i(t)$ is the measured values at t , and V_i is the base value of the i th sensor. Here, the input value to the spiking neuron is fuzzified. Furthermore, the base value is updated in case of accelerometer equipped with the movable objects such as chair or bad,

$$V_i \leftarrow (1 - \xi^{sen}) \cdot V_i + \xi^{sen} \cdot v_i(t) \quad (8)$$

where ξ^{sen} is the learning rate to adjust the base value to the current situation of the movable objects. If the neuron is fired, this means a person uses or moves its corresponding furniture. In this way, the firing pattern indicates the time-series of human position in the room. If the position of each sensor is not localized, the simultaneous firing of sensor neuron and human walking indicates the high possibility that the

person uses the furniture or equipment. Based on this discussion, we can assume the position of the person as that of the i th furniture or equipment, ($X_{i,1}^S$, $X_{i,2}^S$).

$$X_{i,k}^S \leftarrow (1 - \alpha^S) X_{i,k}^S + \alpha^S \cdot \ddot{x}_k \quad \text{if } p_i(t) = 1 \quad (9)$$

where α^S is the update rate; \ddot{x}_k ($k=1,2$) is the human position estimated by the 3D distance image sensor. Furthermore, if the PSP of the estimated human position is less than the predefined threshold and if the temporal difference of human position is also small, then the position of the furniture or equipment nearest with the current position is used as the human location.

3 Experimental Results

This section shows comparison results and analyzes the performance of the proposed method. Figure 2, 3 shows the experimental example of a living room where the illumination sensors are attached with the fridge and cabinet, and the accelerometers are attached with the chair and bed. The number of spiking neurons is 6. These neurons are used for measuring the states of (1) entrance door, (2) chair, (3) bed, (4) cabinet, and (5) fridge. A person entered from the door, and sat in the chair. Next, the person opened the fridge to take a water bottle, and soon, took out medicine from the cabinet. The person went to the bed, and finally went out from the room. Figure 3 Shows snapshots of a person in the living room.

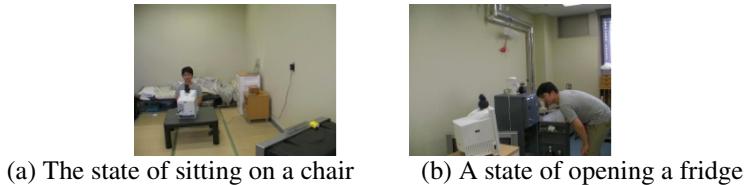


Fig. 3. Snapshots of behaviors in a living room

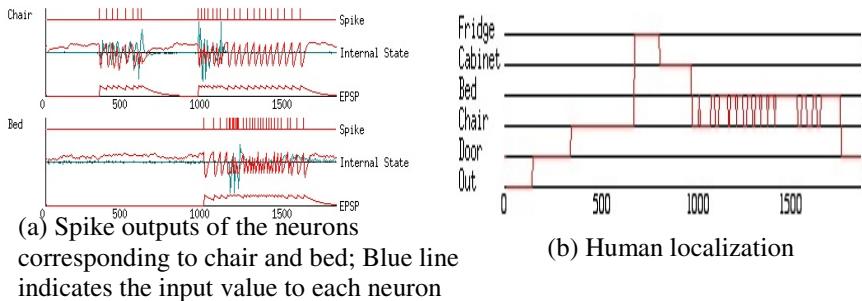


Fig. 4. Experimental results of the proposed method without the update of base values (Case 1)

We recorded the data measured by the sensor network devices in order to analyze the performance of the proposed method. Figure 4(a) shows an experimental result of SNN corresponding to chair and bed without the update of base value (Case 1) where the blue line is the input to each spiking neuron; the red line of the upper figure indicates spike outputs, the red line in the middle of the figure indicates the internal state (presynaptic action potential; PSP), and the red line in the lower figure is the value of PSP from each neuron to measure the state of furniture or equipment. Each neuron fires as the state of internal state increases according to the sensory input calculated by the difference from the base value in each sensor. Figure 4(b) shows the estimated location of the person in the room. The fire of a neuron corresponding to the bed occurs just after that of the chair, and the weight connection between them is updated strongly. As a result, both of neurons are fired continuously each other, and this causes the bursting of two neurons. Since the posture of movable objects is changing easily, the base values for measuring the motion of furniture should be updated step by step.

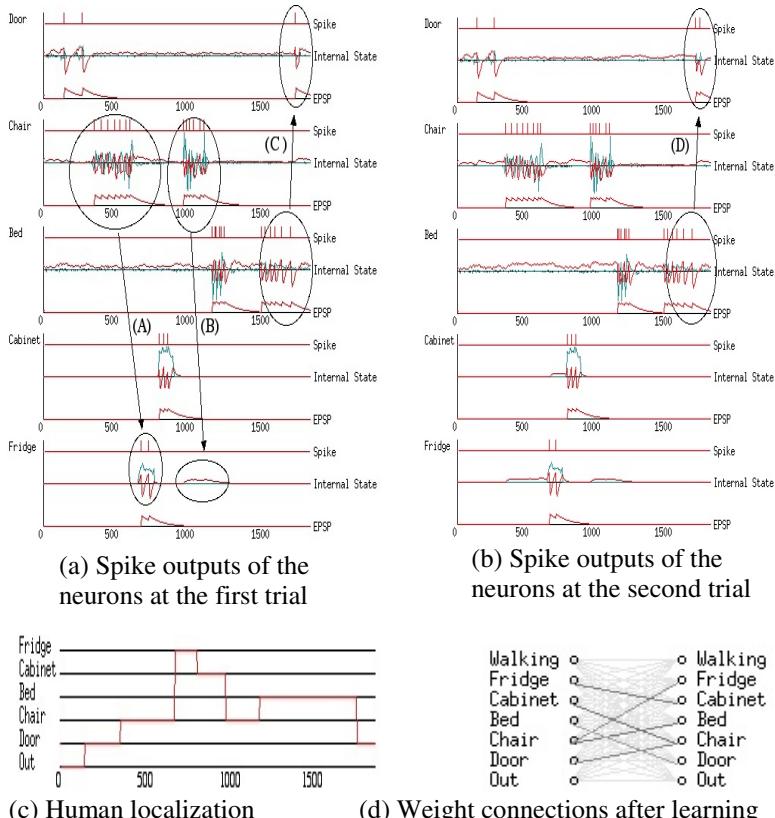


Fig. 5. Experimental results of the proposed method (Case 2)

Next, we conduct the simulation of the proposed method using the recorded data in Case 1. Figure 5 shows the simulation result of SNN where (a) shows the behavior of SNN at the first trial after the weight values are initialized; (b) the behavior of SNN at the second trial without initializing the weight values after the first trial; (c) the estimated location of the person in the room; and (d) weight connections after the learning. The weight value between the chair and fridge is strengthened because the person moved from the chair to the fridge ((A) in Fig.5 (a)). As a result, the PSP is transmitted from the neuron corresponding to the chair to that of fridge, and the internal state of the neuron corresponding to the fridge is increasing without the external input ((B) in Fig.5(a)). Furthermore, the weigh connection of neurons from the bed to door is also strengthened ((C) in Fig.5 (a)). As a result, the times of fires of the neuron corresponding to the door at the second trial are increased from one to two ((D) in Fig.5 (b)). This means the learning of weight parameters can enhance the performance of human localization. Figure.5(d) shows weight parameters between neurons of SNN based on Hebbian learning where the degree of gray indicates the strength of weight connection between two neurons. The strength between two neurons shows the temporal transition of human location in the room. By using the connection strength, the robot partner can predict the human behavior, and perform the natural communication.

4 Summary

This paper analyzed the performance of the human localization by spiking neural network in informationally Structured Space based on sensor network. First, we applied a spiking neural network to extract the human position in a living room. Next, we proposed how to update the base value as a method of preprocessing to generate input values to spiking neurons, and the learning method of spiking neural network based on the time series of measured data. Experimental results show the effectiveness of the proposed methods. The obtained results are available for the observation of human location in a living room. However, there are trade-off or balance between the update of base values and the enhancement of human localization by weight connection between neurons. The update of base values can inhibit incorrect fires by external noise, while that might inhibit suitable fires according to small change of measured date. The weight connection can enhance the suitable fires based on the prediction as a result of the temporal learning by time-series of measured data, while that might cause incorrect fires by external noise. Therefore, we intend to propose a learning method to realize the suitable balance of enhancement and inhibition as a future work.

As other future works, we will develop information support system based on text and audio-visual information by the robot partner, and an extraction method of human preference through interaction with people.

References

1. Khemapech, I., Duncan, I., Miller, A.: A Survey of Wireless Sensor Networks Technology. In: Proc. of The 6th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting (2005)

2. Ishiguro, H., Shiomi, M., Kanda, T., Eaton, D., Hagita, N.: Field Experiment in a Science Museum with communication robots and a ubiquitous sensor network. In: Proc. of Workshop on Network Robot System at ICRA 2005 (2005)
3. Satoh, I.: Location-based Services in Ubiquitous Computing Environments. International Journal of Digital Libraries (2006)
4. Remagnino, P., Hagras, H., Monekosso, N., Velastin, S.: Ambient Intelligence: A Gentle Introduction. In: Remagnino, P., Foresti, G., Ellis, T. (eds.) The book entitled “Ambient Intelligence A Novel Paradigm”, pp. 1–15. Springer, Heidelberg (2005)
5. Callaghan, V., Colley, M., Hagras, H., Chin, J., Doctor, F., Clarke, G.: Programming iSpaces: A Tale of Two Paradigms. In: Steventon, A., Wright, S. (eds.) The book entitled “Intelligent Spaces: The Application of Pervasive ICT”. Ch. 24, pp. 389–421. Springer, Heidelberg (2005)
6. Morioka, K., Hashimoto, H.: Appearance Based Object Identification for Distributed Vision Sensors in Intelligent Space. In: Proc. of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), vol. 1, pp. 199–204 (2004)
7. Kemmotsu, K., Tomonaka, T., Shiotani, S., Koketsu, Y., Ichara, M.: Recognizing Human Behaviors with Vision Sensors in Network Robot Systems. In: Proc. of The 1st Japan-Korea Joint Symposium on Network Robot Systems (JK-NRS 2005) (2005)
8. Khemapech, I., Duncan, I., Miller, A.: A Survey of Wireless Sensor Networks Technology. In: Proc. of The 6th Annual Post Graduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting (2005)
9. Corderio, C., Agrawal, D.: Ad Hoc & Sensor Networks. World Scientific, Singapore (2006)
10. Kubota, N., Nishida, K.: Cooperative Perceptual Systems for Partner Robots Based on Sensor Network. International Journal of Computer Science and Network Security 6(11), 19–28 (2006)
11. Kubota, N., Shimomura, Y.: Human-Friendly Networked Partner Robots toward Sophisticated Services for A Community. In: Proc. of SICE-ICCAS 2006, pp. 4861–4866 (2006)
12. Kubota, N., Obo, T., Fukuda, T.: An Intelligent Monitoring System based on Emotional Model in Sensor Networks. In: The 18th IEEE International Symposium on Robot and Human Interactive Communication, pp. 346–351 (2009)
13. Kubota, N., Tang, D., Obo, T., Wakisaka, S.: Localization of Human Based on Fuzzy Spiking Neural Network in Informationally Structured Space. In: Proc. of 2010 IEEE World Congress on Computational Intelligence (WCCI 2010), Barcelona, Spain, July 18–23, pp. 2209–2214 (2010)
14. Obo, T., Kubota, N., Lee, B.H.: Localization of Human in Informationally Structured Space Based on Sensor Networks. In: Proc. of 2010 IEEE World Congress on Computational Intelligence (WCCI 2010), Barcelona, Spain, July 18–23, pp. 2215–2221 (2010)
15. Kubota, N., Yuki, K., Baba, N.: Integration of Intelligent Technologies for Simultaneous Localization and Mapping. In: Proc. of ICROS-SICE International Joint Conference 2009 (ICCAS-SICE 2009), Fukuoka, Japan, August 18–21 (2009)
16. Satomi, M., Masuta, H., Kubota, N.: Hierarchical Growing Neural Gas for Information Structured Space. IEEE Symposium Series on Computational Intelligence (2009)
17. Anderson, J.A., Rosenfeld, E.: Neurocomputing. The MIT Press, Cambridge (1988)
18. Gerstner, A.: Spiking Neurons. In: Maass, W., Bishop, C.M. (eds.) Pulsed Neural Networks. ch. 1, pp. 3–53. MIT Press, Cambridge (1999)
19. Maass, W., Bishop, C.M. (eds.): Pulsed Neural rural Networks. MIT Press, Cambridge (1999)
20. Little, W.A., Shaw, G.L.: A statistical theory of short and long term memory. Behavioral biology 14, 115

Computational Model of the Cerebral Cortex That Performs Sparse Coding Using a Bayesian Network and Self-Organizing Maps

Yuuji Ichisugi¹ and Haruo Hosoya²

¹ National Institute of Advanced Industrial Science and Technology(AIST)
Tsukuba Central 2, Tsukuba, Ibaraki 305-8568, Japan

y-ichisugi@aist.go.jp

² 7th Bld. of Faculty of Science, The University of Tokyo,
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan
hahosoya@is.s.u-tokyo.ac.jp

Abstract. The authors have proposed a computational model of the cerebral cortex, called the BESOM model, that combines a Bayesian network and Self-Organizing Maps. In this paper, we add another model of the cerebral cortex, called sparse coding, into our model in a biologically plausible way. In the BESOM model, hyper-columns in the cerebral cortex are interpreted as random variables in a Bayesian network. We extend our model so that random variables can become “inactive.” In addition, we apply bias at the time of recognition so that almost all of the random variables may become inactive. This mechanism realizes sparse coding without breaking the theoretical framework of the model based on the Bayesian networks.

Keywords: Sparse coding, Bayesian network, Self-organizing maps, cerebral cortex, BESOM.

1 Background

Some computational neuroscientists have begun to understand that the *Bayesian network* [2] is the essential mechanism of the cerebral cortex [6][8][9][11][12][13][14][15]. Bayesian networks are a technology for knowledge representation that can efficiently express the causal relationships among many random variables. Models based on Bayesian networks can successfully explain the fundamental mechanism of the cerebral cortex, namely, robust pattern recognition using prediction based on context[8]. Furthermore, previous studies strongly suggest that the cerebral cortex is a Bayesian network, according to models that reproduce electrophysiological phenomena[9][15] and models that explain the roles of major anatomical characteristics of the cerebral cortex[11][12]. Moreover, the mechanism of the motor area is explained by using a combination of a Bayesian network and a reinforcement learning mechanism[13]. Another study realizes probabilistic reasoning at the Markov Random Field (a model similar to Bayesian networks) with a biologically plausible neural circuit of spiking neurons[14].

Although previous models explain only some parts of the mechanism of the cerebral cortex functions, we believe that, in the not too distant future, these models (maybe including [5]) will become integrated into one universal model based on Bayesian networks.

The authors have also proposed a model of the cerebral cortex using a Bayesian network, called the *BESOM model* [11][13]. This model inherits the basic structure of Neocognitron and its successors[1], which are macroscopic models of the visual area. Moreover, the mechanism of the *Self-Organizing Maps* (SOM) [3], whose origin is the model of the orientation columns of the primary visual area, is adopted as a learning algorithm. BESOM can also be regarded as a novel machine-learning algorithm that uses multiple SOMs, like [10] and [17].

On the other hand, there is another model, called the *sparse coding model* [4], of an aspect of the cerebral cortex. Sparse coding is a kind of unsupervised learning whose goal is to acquire a basis on which to express an input vector by a linear summation of a smaller number of basis vectors. It has been suggested that sparse coding is performed in the primary visual area[4] and primary auditory area[16] of the cerebral cortex. Sparse coding efficiently compresses information and is supposed to conserve both the energy spent by neurons and the cost of maintaining synapses.

In this paper, we propose a biologically plausible computational model that unifies three mechanisms, a Bayesian network, SOM, and sparse coding. The model is an extension of the previous BESOM model.

2 Overview of the Model

2.1 Bayesian Network and MPE

A Bayesian network[2] is a model of knowledge representation that expresses causal relationships between random variables using a directed acyclic graph. Random variables are expressed as *nodes*, and relationships between random variables are expressed as *edges*. Each node has a table of conditional probability, which denotes the degree to which nodes are related to the set of its parent nodes.

In a Bayesian network, an *MPE* (*most probable explanation*) is the set of values of nodes that most likely explains given observed data. Let \mathbf{i} be a set of values of observed random variables and \mathbf{h} be a set of values of hidden variables (unobserved random variables). MPE $\hat{\mathbf{h}}$ is defined by the following equations.

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmax}} P(\mathbf{h}|\mathbf{i}) = \underset{\mathbf{h}}{\operatorname{argmax}} P(\mathbf{h}, \mathbf{i}) \quad (1)$$

where $P(\mathbf{h}, \mathbf{i})$ is the joint probability of \mathbf{h} and \mathbf{i} , which can be calculated by the following formula if a Bayesian network is given:

$$P(\mathbf{h}, \mathbf{i}) = \prod_{x \in \mathbf{h} \cup \mathbf{i}} P(x|parents(x)) \quad (2)$$

where $parents(x)$ denotes the set of values of parent nodes of node X .

2.2 The Structure of Two-Layered BESOM

Figure 1 shows the neural circuit of the two-layered BESOM network used in this paper. The ellipses are *nodes* (random variables), the small circles are *units* (values that random variables can take), and the straight lines are the *connections* (conditional probabilities) between units.

Although every pair of units contained in the two layers (the hidden layer and the input layer) has a connection, most connections are omitted in Fig. 1. There is no connection between units in the same layer.

At the time of learning, each node in the hidden layer plays the role of a competitive layer of SOM, learns the *weights of the connections* (conditional probabilities) between units, and compresses the input from its child nodes.

At the time of recognition, all nodes play the role of random variables in a Bayesian network. At this time, all edges of the Bayesian network are from all nodes of the hidden layer to all nodes of the input layer. There is no edge between nodes in the same layer.

The number of nodes, the number of units, and the network structure of nodes are given first, and are not changed by learning.

When BESOM is used as a cerebral cortex model, each node is a hyper-column, each unit is a column (minicolumn), and each connection weight between a pair of units is the weight of a synapse.

The correspondence of components in BESOM, SOM, a Bayesian network and a cerebral cortex is summarized in Table 1.

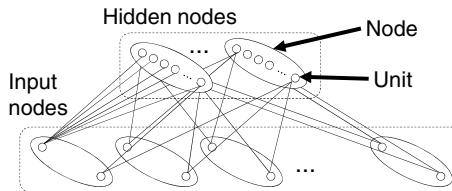


Fig. 1. The structure of two-layered BESOM network

Table 1. The correspondence of components in BESOM, SOM, a Bayesian network and a cerebral cortex

BESOM	SOM	Bayesian network	Cerebral cortex
node	competitive layer	random variable	hyper-column
unit	element of input vector, unit of competitive layer	value of random variables	column
parent node	competitive layer	parent node (cause)	upper area
child node	input layer	child node (result)	lower area
output of unit	the winner of competition	MPE	response of column
weight of connection	element of reference vector	conditional probability	weight of synapse

2.3 Recognition Steps and Learning Steps

When a set of values of input nodes (observed data) is given, BESOM executes a recognition step and a learning step. By repeating this cycle, BESOM acquires an approximated generative model of the outer world.

Let $\mathbf{i}(t)$ be the set of values of input nodes at time t . We assume that each input is generated from i.i.d. (independent identical distribution). The objective of the learning is to maximize the likelihood of the parameter θ (the vector of all the elements of all conditional probability tables).

$$\theta^* = \operatorname{argmax}_{\theta} \left(\prod_{i=1}^t P(\mathbf{i}(i)|\theta) \right) P(\theta) \quad (3)$$

$$= \operatorname{argmax}_{\theta} \left(\prod_{i=1}^t \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(i)|\theta) \right) P(\theta) \quad (4)$$

Although the algorithm described in this paper is somewhat complex, its essence can be simply expressed by the following two equations.

At the recognition steps, the estimated values of hidden variables $\hat{\mathbf{h}}(t)$ (MPE) based on the current parameter $\theta(t)$ are calculated by the following equation.

$$\hat{\mathbf{h}}(t) = \operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(t)|\theta(t)) \quad (5)$$

At the following learning step, the parameter $\theta(t+1)$ is estimated by Eq. (4), with an approximation that replaces the marginalization of \mathbf{h} with the estimated value $\hat{\mathbf{h}}(t)$.

$$\theta(t+1) = \operatorname{argmax}_{\theta} \left(\prod_{i=1}^t P(\hat{\mathbf{h}}(i), \mathbf{i}(i)|\theta) \right) P(\theta) \quad (6)$$

The exact calculation of Eq. (5) and Eq. (6) requires enormous amounts of computation. On the other hand, the actual brain should execute both recognition and learning very efficiently with some clever approximation. In addition, the brain should perform online learning; that is, $\theta(t+1)$ should be calculated only by using $\mathbf{i}(t)$, $\hat{\mathbf{h}}(t)$ and $\theta(t)$. We think the algorithm described in this paper satisfies these restrictions and is thus a plausible brain model.

2.4 Basic Idea of Sparse Coding for BESOM

We add the mechanism of sparse coding to BESOM by introducing an “inactive state” into each random variable (i.e., a hyper-column). If the inactive state is introduced to each random variable and large numbers of nodes become inactive at each recognition step, sparse coding will be realized in BESOM. (See Fig. 2) Actually, the way in which inactive states are introduced into a Bayesian network is not obvious. Moreover, in order for the mechanism to become an appropriate model of the cerebral cortex, it should be implemented in a biologically plausible way. Detailed recognition and learning algorithms that realize the idea of inactive states are described in the next section.

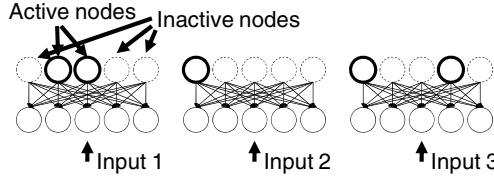


Fig. 2. Basic idea of sparse coding using BESOM. Different sets of hidden nodes become inactive depending on inputs.

3 Algorithm

3.1 Characteristics of the Acquired Bayesian Network

We restrict the Bayesian network acquired by the proposed algorithm to satisfy the following conditions.

1. Let X be a node (a random variable). X 's value should be one of the following $s + 1$ values.

$$X \in \{x_\phi, x_1, x_2, \dots, x_{s-1}, x_s\} \quad (7)$$

We call the value x_ϕ the “ ϕ -value” and the values other than x_ϕ “non- ϕ -values.” The value x_ϕ means the node X is inactive.

2. A conditional probability table $P(x_i|u_1, \dots, u_m)$ ($i = \phi, 1, \dots, s$) should satisfy the following equation:

$$P(x_i|u_1, \dots, u_m) = \frac{1}{m} \sum_{k=1}^m P(x_i|u_k) \quad (8)$$

The constant $1/m$ normalizes so that $\sum_i P(x_i|u_1, \dots, u_m)$ becomes 1.

3. Let U be a node and X be a child node of U . The ϕ -value of U , u_ϕ , should not have a causal relation to X . That is, the following equation should hold.

$$P(x_i|u_\phi) = P(x_i) \quad (i = \phi, 1, \dots, s) \quad (9)$$

This condition will be satisfied when the learning converges because of the learning rules described in Section 3.3.

3.2 Recognition Step

In the original sparse coding algorithm [4], sparseness is realized by adding the penalty term for activeness to the objective function. We apply this idea to BESOM.

Eq. (2), which calculates the joint probability of an MPE candidate \mathbf{h} and an input \mathbf{i} , is modified as follows, so that a penalty is given depending on the number of activity nodes:

$$P(\mathbf{h}, \mathbf{i}) = e^{-\beta A(\mathbf{h})} \prod_{x \in \mathbf{h} \cup \mathbf{i}} P(x|parents(x)) \quad (10)$$

where β is the parameter that controls the sparseness, $A(\mathbf{h})$ is the number of active nodes in \mathbf{h} .

The formula for this joint probability is used at the recognition steps that calculate MPE. We do not calculate strict MPE in the present simulation. The approximate MPE is calculated by the hill-climbing method. In the actual brain, a variation of the belief revision algorithm [2], which seems biologically more plausible [14], may be used.

If MPE is used as a learning step as it is, the learning will be likely to fall into a local minimum. Then, to avoid the local minimums, we add a moderate amount of noise to the recognition results in the early stages of the learning.

3.3 Learning Step

At each a learning step, the weights of connections between units will be updated according to the MPE calculated at the preceding recognition step. In the nodes at the hidden layer, the units that correspond to the values of MPE are regarded as winners for competitive learning.

In this section, we clarify our explanation by explaining a simplified version of the learning rule, which does not include neighborhood-learning rules.

Let us call the units corresponding to the value of MPE *winner units*. In the input layer, winner units represent observed data. In the hidden layer, winner units represent estimated values of hidden variables. Let X be a node at the hidden layer and Y_l ($l = 1, \dots, n$) be a child node of X at the input layer.

The connection weight w_{ij}^l between the winner unit x_i and unit y_j^l is updated by the following formulas:

- When $i = \phi$:

$$w_{\phi j}^l = \begin{cases} \Phi_{Y_l} & (j = \phi) \\ (1 - \Phi_{Y_l})/s & (j \neq \phi) \end{cases} \quad (11)$$

where Φ_{Y_l} is the frequency that Y_l becomes a ϕ -value in MPE; these Φ_{Y_l} are learned separately.

- When $i \neq \phi, j \neq \phi$:

$$w_{ij}^l \leftarrow w_{ij}^l + \alpha(v_j^l - w_{ij}^l) \quad (12)$$

where α is the learning rate, v_j^l is an input from the child node Y_l , whose value is 1 if unit y_j^l is the winner unit, and is 0 otherwise.

- When $i \neq \phi, j = \phi$:

$$w_{i\phi}^l = 1 - \sum_{j=1}^s w_{ij}^l \quad (13)$$

The connection weight w_{ij}^l is learned by the above algorithm. When the estimated values of the hidden variables are regarded as true observed values and the learning rate α is appropriately scheduled, the connection weight w_{ij}^l becomes a maximum likelihood estimator of the conditional probability $P(Y_l = y_j^l | X = x_i)$ [11].

The obtained conditional probabilities are used at the next recognition step to calculate joint probability, defined as Eq. (10), assuming the constraints of Eq. (8).

4 Experiment: Sparse Coding of Natural Images

We used images provided by Olshausen (the images filtered with whitening/low-pass as described in [7]) and clipped to the range [0, 1]. The images are used as input to a two-layered BESOM, with 4 nodes in the hidden layer and 49 in the input layer. At each step, we extracted a image patch with $7 \times 7 = 49$ pixels from a random position. Then, we gave the pixel intensities in the image patch to the binary input nodes. The value of each input node is set to 1 according to the pixel intensity taken as a probability. For example, for intensity 0.2, the value was set to 1 with probability 0.2. The parameter that determines the sparseness of node activity is set to $\beta = 8$.

Figure 3(a) shows a learning result. It shows the values of the conditional probability tables of the ϕ -value units $P(y_1^l|x_\phi)$ and non- ϕ -value unit $P(y_1^l|x_i)$ ($i = 1, \dots, 9, l = 1, \dots, 49$) as the brightness of 7×7 pixels.

It is shown that the non- ϕ -value unit of each node obtained the conditional probability table with orientation selectivity like V1 simple cells [4]. The ϕ -value unit of each node learned the mean of input images. In this experiment, we found that 0-3 nodes are activated according to the inputs.

Figure 3(b) shows a learning result with the sparseness parameter $\beta = 0$. In this case, every base image is close to the mean image of input because all nodes tend to be active. The result shows weak orientation selectivity.

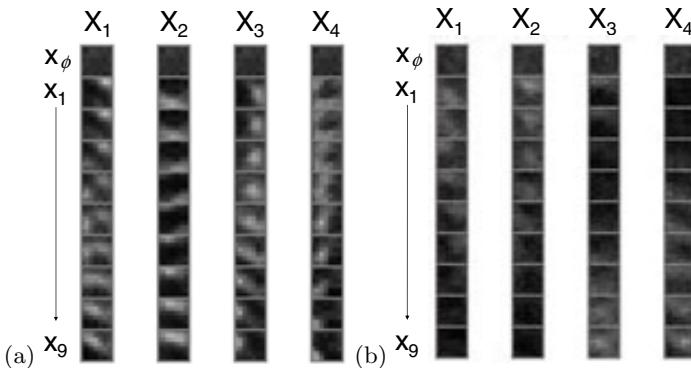


Fig. 3. Learning result of sparse coding of natural images. The sparseness parameter values are $\beta = 8$ (a) and $\beta = 0$ (b).

References

1. Fukushima, K.: Neural network model for selective attention in visual-pattern recognition and associative recall. *Applied Optics* 26(23), 4985–4992 (1987)
2. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
3. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (1995)
4. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381(6583), 607–609 (1996)

5. Coward, L.A.: The Recommendation Architecture: lessons from the design of large scale electronic systems for cognitive science. *Journal of Cognitive Systems Research* 2(2), 111–156 (2001)
6. Lee, T.S., Mumford, D.: Hierarchical Bayesian inference in the visual cortex. *Journal of Optical Society of America, A.* 20(7), 1434–1448 (2003)
7. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research* 37(23), 3311–3325 (2003)
8. George, D., Hawkins, J.: A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In: Proc. of IJCNN 2005, vol. 3, pp. 1812–1817 (2005)
9. Rao, R.: Bayesian inference and attention in the visual cortex. *Neuroreport* 16(16), 1843–1848 (2005)
10. Oshiro, N., Kurata, K.: Separating visual information into position and direction by two inhibitory-connected SOMs. *Artif. Life and Robotics* 9(2), 86–89 (2005)
11. Ichisugi, Y.: The cerebral cortex model that self-organizes conditional probability tables and executes belief propagation. In: Proc. of International Joint Conference on Neural Networks (IJCNN 2007), pp. 1065–1070 (August 2007)
12. Roehrbein, F., Eggert, J., Koerner, E.: Bayesian Columnar Networks for Grounded Cognitive Systems. In: Proc. of the 30th Annual Conference of the Cognitive Science Society, pp. 1423–1428 (2008)
13. Hosoya, H.: A motor learning neural model based on Bayesian network and reinforcement learning. In: Proceedings of International Joint Conference on Neural Networks (2009)
14. Litvak, S., Ullman, S.: Cortical Circuitry Implementing Graphical Models. *Neural Computation* 21, 3010–3056 (2009)
15. Chikkerur, S., Serre, T., Tan, C., Poggio, T.: What and Where: A Bayesian Inference Theory of Attention, *Vision Research* (2010)
16. Terashima, H., Hosoya, H.: Sparse codes of harmonic natural sounds and their modulatory interactions. *Network: Computation in Neural Systems* 20(4), 253–267 (2009)
17. Furukawa, T.: SOM of SOMs. *Neural Networks* 22(4), 463–478 (2009)

Complex Spiking Models: A Role for Diffuse Thalamic Projections in Complex Cortical Activity

Peter Stratton¹ and Janet Wiles²

¹ School of Information Technology and Electrical Engineering, The University of Queensland,
Brisbane, Australia

² Queensland Brain Institute, The University of Queensland, Brisbane, Australia
`{stratton,wiles}@itee.uq.edu.au`

Abstract. Cortical activity exhibits complex, persistent self-sustained dynamics, which is hypothesised to support the brain's sophisticated processing capabilities. Prior studies have shown how complex activity can be sustained for some time in spiking neural network models, but network activity in these models resembled high firing rate seizure which would eventually fail, leading to indefinite quiescence. We present a spiking network model of cortex innervated by diffuse thalamic projections, called the Complex Spiking Model (CSM). The model exhibits persistent, self-sustained, non-periodic, complex dynamics at low firing rates. Multiple network configurations were tested, systematically varying diffuse excitation from the thalamus, strength of the local cortical inhibition and excitation, neighbourhood diameters, synaptic efficacies and synaptic current time constants. Complex activity in all the network configurations depended strongly upon the strength of the diffuse excitation from the thalamus. We propose that diffuse thalamic projections to cortex facilitate complex cortical dynamics and are likely to be an important factor in the support of cognitive functions.

Keywords: Spiking neural networks, complex cortical activity, diffuse thalamic projections, thalamo-cortical model, Izhikevich neurons.

1 Introduction

Cortical activity exhibits persistent ongoing complex dynamics [1, 2], with oscillatory activity which approximates a power law [3]. As activity shifts continuously throughout the brain, brain regions constantly couple (integrate) and de-couple (segregate) across multiple spatial and temporal scales [4-6]. This complex ongoing dynamics is hypothesised to support the brain's flexibility and sophisticated processing capabilities [1, 7, 8], including memory retrieval, planning and problem solving [9, 10]. During times when the brain is not actively processing sensory stimuli or task-related events, and as such is in a state known as the 'resting' or 'default-mode' state, brain activity is concentrated in a well-defined sub-network including regions of frontal and association cortices [11]. These are regions usually associated with higher-level cortical processing.

The thalamus has strong reciprocal connections with the cortex. A set of connections known as the *specific* thalamo-cortical connections originate in distinct thalamic

nuclei and terminate in distinct cortical regions; one of the primary roles of these connections is to convey perceptual stimuli to the cortex. The *diffuse* connections, on the other hand, project non-specifically from the thalamus to many cortical regions, notably frontal and association cortices; brain regions that are known to be important for cognitive functions such as memory retrieval, planning and problem solving. There is a significant degree of overlap between these regions and those identified as belonging to the default-mode network. Activation of any of the regions of the thalamus from which these diffuse connections arise results in the recruitment and co-activation of many or all of the other regions, and consequent propagation of activity to all connected regions of the cortex [12]. Thus the diffuse thalamo-cortical projection system is implicated in general cortical arousal and control of gross cortical activity.

Previous studies have shown how complex activity can be sustained for some time in spiking neural network models with connection delays [4, 13]. These models utilised small-world connectivity as the structural facilitator of dynamical integration and segregation. The sparse long-range connections encouraged propagation of activity in self-perpetuating loops, provided that the connection densities and propagation delays were appropriately matched and within well-defined ranges. However, activity in these models was characterised by repetitive population- or cluster-wide oscillations; the complexity manifested only as unpredictability or ‘jitter’ in individual neuron spike times while population activity remained semi-synchronous and seizure-like. Such activity is not observed in healthy, functioning cortex. Additionally, the activity in these models was not indefinitely self-sustained; it would always eventually fail and return the networks to the quiescent state.

Recent work has shown how complex, non-periodic, low firing-rate, and indefinitely self-sustained activity can be supported in a cortical network model [14]. The cortical network combined local neighbourhood connectivity with a number of long-range connections and all connections were subject to activity-dependent facilitation and depression. The network activity did not enter seizure or quiescence, and activity patterns did not repeat periodically despite network dynamics being deterministic and network input being constant. However, the network incorporated a single artificial globally-connected inhibitory neuron and no local inhibition. In the current study, we show that complex, self-sustained activity can be achieved with a cortical model incorporating realistic thalamic connectivity and local inhibition, and that the thalamic connectivity, analogous to the global inhibition of the previous model, is in fact required to support complex activity in the cortical neurons.

2 Complex Spiking Model (CSM)

We present a spiking network model of cortex innervated by diffuse thalamic projections, called the Complex Spiking Model (CSM) (see Fig. 1). The connectivity from the diffuse thalamic nucleus (*Diffuse*), to the cortex, to the reticular nucleus (*RN*) with inhibition back to the diffuse nucleus, is assumed to have a role in cortical activation control in brains, since if cortical activity rises, inhibition to the diffuse nucleus increases (through the reticular nucleus), which decreases the excitation being sent to the cortex.

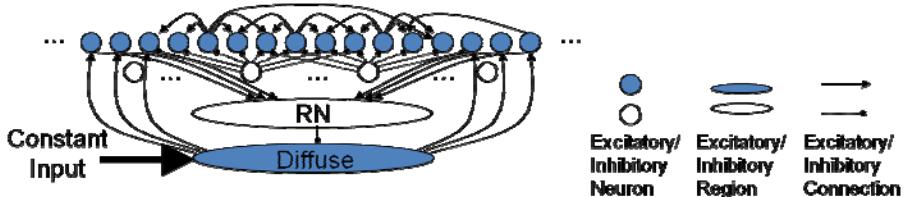


Fig. 1. The Complex Spiking Model (CSM) used for this research (not all connections shown). The model is structured as follows: Excitatory cortical neurons (small shaded circles) are innervated by diffuse thalamic projections (from shaded oval). The thalamic nucleus from which the diffuse connections arise receives continuous steady (not time-varying) input current. Cortical neurons project back to the thalamic reticular nucleus which inhibits the diffuse nucleus, completing the diffuse thalamocortical loop. Cortical neurons excite each other within a small neighbourhood radius, inhibit each other through local interneurons within a larger inhibitory radius, and project a small number of random long range excitatory synapses to other cortical neurons. All cortical excitatory synapses are subject to activity-dependent facilitation and depression. For details of the cortical neuron and synapse implementation see [14]. RN – thalamic reticular nucleus; Diffuse – diffuse (intralaminar) thalamic nucleus.

The CSM was constructed using Izhikevich model cortical neurons [15], and the reticular and diffuse thalamic nuclei were each implemented as a single analog neuron with output equal to the sum of its input currents. Synapses were modeled as postsynaptic currents, initiated by presynaptic spikes, that exponentially decayed with characteristic time constants of 35 to 50 milliseconds to emulate the long-duration synaptic currents generated by NMDA receptors [16]. To implement short-term synaptic dynamics, a combination of depression and facilitation was used [17]. For synaptic depression, all of the currently-available proportion of neurotransmitter p at each synapse was released at the occurrence of each spike; p recovered back to unity with time constant d . The rate of change of p for neuron i , \dot{p}_i , was given by:

$$\dot{p}_i = -\delta(t - t_i)p_i + \frac{(1 - p_i)}{d} \quad \dots(1)$$

where δ is the Dirac delta function, t_i is the time of the last spike from presynaptic neuron i . With dynamic synapses the calculation for all the synaptic currents into a postsynaptic neuron, $I_{dynamic}$, becomes:

$$I_{dynamic} = \sum_i w_i p_i \quad \dots(2)$$

where w_i is the synaptic weight from neuron i .

3 Methods

To measure the complexity of the spiking activity generated by the network, the interspike intervals (ISI) for each neuron were individually determined, then the standard deviations (SD) for each neuron were calculated, and finally all standard deviations were averaged to calculate the mean interspike interval standard deviation (ISI-SD) for

the network. Networks exhibiting either random uncorrelated activity or seizure-like dynamics both have low ISI-SD, since the lack of any segregation in network activity reduces the spread of interspike intervals. In contrast, complex activity is characterised by a high ISI variability, since neurons fire rapidly when integrated into an active neural assembly, then can remain quiet for extended periods while segregated.

Initially the network was set up as in Fig. 1, then network activity was simulated for 10,000 seconds to check that neither seizure nor activity failure eventuated, and that complex self-sustained activity persisted for the entire simulation. Subsequently, multiple network configurations were tested with different values of network parameters. Parameters which were varied included strength of the local cortical excitation and inhibition, neighborhood diameters of excitation and inhibition, and synaptic current time constants, along with the strength of the diffuse excitation from the thalamus. For each network configuration, the mean network firing rate and ISI-SD measures were calculated. These measures were plotted on two dimensional graphs called *heat maps*, where each point represents a different network configuration and is shaded according to the rate or measure at that point. Heat maps provide a succinct visualisation of large numbers of simulations showing the range of network dynamics in different regions of parameter space. The heat maps in these studies use one network per pixel, and 100x100 networks per map.

4 Results and Discussion

The CSM exhibits persistent, self-sustained, non-periodic and complex dynamics at low firing rates (see Fig. 2). Local assemblies of neurons spontaneously coalesce and disintegrate, mediated by the neighbourhood excitation. Multiple assemblies are active simultaneously across the network, mediated by the long range connections, and firing of these assemblies correlates and decorrelates over time as their phase relationships shift. One assembly may be active several times over the course of network activity, but never with an identical set of other assemblies simultaneously active elsewhere in the network. Activity patterns never repeat and there is no periodicity in the observed network dynamics (see Fig. 2 top-left), yet there are clear oscillations in network assemblies which correlate and decorrelate over time (see Fig. 2 bottom-left). These dynamics lead to a power law-like power spectrum (see Fig. 2 right). All these results hold for simulations of activity of up to 10,000 seconds duration, despite network dynamics being deterministic and network input being constant. The network activity shown in Fig. 2 has a high ISI-SD of 0.85 sec, despite neurons firing at 40 Hz or more when part of an active assembly, since outside of this assembly activity, each neuron typically falls silent for seconds or even tens of seconds. These results replicate those obtained in earlier work [14], but eliminates the artificial globally-connected inhibitory neuron, replacing it with a model of local inhibition and thalamic connectivity.

To show that the thalamo-cortical loop in the CSM facilitates the generation of complex activity, we tested multiple network configurations. Two network parameters were varied – the diffuse excitation from the thalamus, and the strength of the local cortical inhibition. The mean network firing rates and ISI-SD measures were plotted on *heat maps* to facilitate visualisation of the network dynamics as parameters were

varied (see Methods). The systematic mapping of the parameter spaces shows that complex activity is dependent upon the strength of the diffuse excitation from the thalamus (see Fig. 3). Provided that the diffuse excitation is in a defined range (0.015 to 0.06 synaptic conductance), then a broad range of local inhibition strength suffices for low firing rate, high complexity dynamics to ensue, and the strength of the local inhibition within this broad range has little effect on the dynamical complexity. For diffuse excitation outside the identified range however, complex activity occurs only for a very narrow range of local inhibition strength around 0.05.

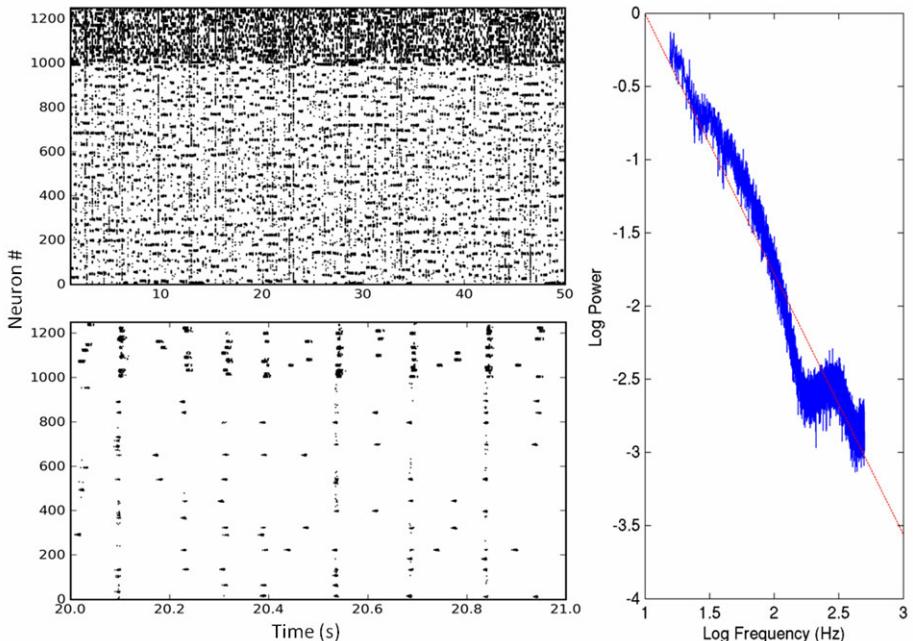


Fig. 2. Typical dynamics of the CSM. Activation of assemblies of neurons can be seen to correlate and decorrelate over time, indicating complex activity. Top-left: Raster plot showing spiking behaviour of 1000 excitatory neurons and 250 inhibitory interneurons (Y axis) over 50 seconds (X axis). Dots show spike times; many spikes appearing together indicate activation of a cell assembly and appear as horizontal lines. The network dynamics is complex, activity patterns are not seen to repeat, and the network does not enter seizure or quiescence. Bottom-left: Zooming in on one second of activity from the raster plot on the top-left shows oscillatory dynamics and sparse spiking patterns giving rise to low sustained firing rates (mean rate = 1.6 Hz). Right: As indicated by the straight line fit (dotted) on the log-log graph, the power spectrum of the activity shown on the left approximates a power law, similar to spectral activity seen in the brain [3].

Simulation studies with an extensive range of other network parameters were conducted, varying strength of the local cortical excitation, neighbourhood diameters of excitation and inhibition, and synaptic current time constants along with the diffuse excitation from the thalamus (results not shown). All the networks simulated yielded comparable findings to that shown in Fig. 3: without the global control of network activity afforded by the thalamo-cortical loop, complex dynamics did not arise.

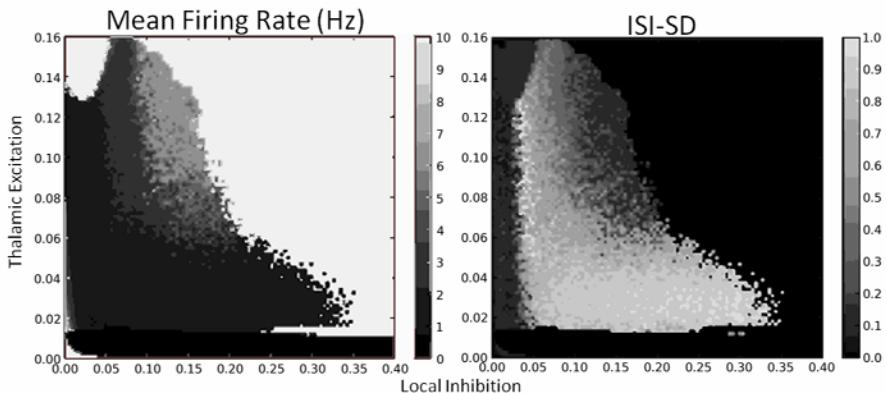


Fig. 3. Heat maps showing CSM dynamics for a range of thalamic excitation and local inhibition strengths. Networks were simulated with varying thalamic excitation (0–0.16 synaptic conductance, in 100 steps) and local inhibition (0–0.4 synaptic conductance, in 100 steps) resulting in 10,000 simulations in total (with 1 simulation per parameter combination). Low firing rate, high complexity dynamics can be seen to occur robustly for diffuse thalamic excitation within a defined range. Left: Heat map showing mean firing rate, calculated as the total number of spikes fired by the excitatory neurons in the network divided by both the number of neurons and the simulation time in seconds. Mean firing rate depends on the initial parameters set for the network (black – no firing; white – high firing rate seizure). For high values of local inhibition and thalamic excitation (top-right corner), the network enters seizure, shown by firing rates at or above 10 Hz (white). Right: Heat map showing interspike interval standard deviation (ISI-SD, see text for definition). Complex activity, indicated by high ISI-SD (light colours), occurs for a broad range of local inhibition as long as diffuse thalamic excitation is within a specific range. By comparing both heat maps, it can be seen that in the complex activity region (light colours in the heat map on the right), mean firing rate is low (see corresponding dark grey regions in the heat map on the left).

One way of understanding the results is as follows: When global network control is absent, network activity reduces to numerous interconnected pockets of activity that can be mutually excitatory due to activity propagating through the long range connections, but cannot be mutually inhibitory (via the thalamo-cortical loop). Lack of mutual inhibition causes activity to spread rapidly through the network, resulting in simple seizure-like dynamics. By contrast, with the thalamo-cortical loop functionally active, unchecked spread of excitatory influences in the network cannot occur, since any increase in total network activity immediately exerts an influence across the entire network by reduction of diffuse thalamic input.

The CSM presented in this paper demonstrates how complex ongoing activity can result from the interplay between the varying levels of excitation arising from the diffuse thalamic nucleus, the long range cortical connectivity, and the excitability of any given neuron or neural assembly as dictated by its own history of prior activity. If a neuron or assembly has recently been firing, then its synaptic connections with its local neighbors and its long-range partners will be reduced in efficacy due to synaptic depression [17]. Therefore, even if it is receiving strong excitation from its long-range partners and strong input from the diffuse nucleus, it will be unlikely to fire. Other

neurons or assemblies which might not be as strongly innervated by excitatory influences may nevertheless fire instead, explaining why the same assemblies don't repeatedly fire in unison despite being strongly connected. Because there are a large number of connections between multiple potential assemblies in the network at any one time, and each assembly has its own unique firing history and hence excitability state, the combinatorial explosion of possible active assemblies, critically mediated by excitation from the diffuse thalamic nucleus, ensures that complex activity is persistent and never gives way to either seizure (runaway activity) or quiescence (activity failure).

5 Conclusion

This study is the first to demonstrate computationally that complex, non-periodic, self-sustained cortical activity can be facilitated by diffuse thalamic connectivity. Global activation control afforded by the diffuse thalamic nucleus facilitates complex dynamics in the Complex Spiking Model (CSM) because global summation of activity can reliably cause segregation of activity across the entire network. Without global control, network activity reduces to numerous interconnected pockets of activity that can mutually integrate due to activity propagating through the long range connections, but cannot mutually segregate. As is the case in seizure or random activity states that lack segregated activity, spike train complexity is greatly reduced. Based on the simulations presented in this study, we propose that diffuse thalamic projections to cortex facilitate complex cortical dynamics and are likely to be an important factor in the support of cognitive functions.

Acknowledgments. This work was supported by an Australian Research Council (ARC) and National Health and Medical Research Council (NHMRC) Thinking Systems Grant (Special Research Initiative: Thinking Systems; Grant number: TS0669699).

References

1. Breakspear, M., Terry, J.R., Friston, K.J.: Modulation of excitatory synaptic coupling facilitates synchronization and complex dynamics in a nonlinear model of neuronal dynamics. *Network* 52, 151–158 (2003)
2. Honey, C.J., Kötter, R., Breakspear, M., Sporns, O.: Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proceedings of the National Academy of Sciences* 104, 10240–10245 (2007)
3. Kitzbichler, M.G., Smith, M.L., Christensen, S.R., Bullmore, E.: Broadband criticality of human brain network synchronization. *PLoS Computational Biology* 5 (2009)
4. Shanahan, M.: Dynamical complexity in small-world networks of spiking neurons. *Physical Review E* 78, 41924 (2008)
5. Sporns, O., Tononi, G., Edelman, G.M.: Connectivity and complexity: the relationship between neuroanatomy and brain dynamics. *Neural Networks* 13, 909–922 (2000)
6. Varela, F., Lachaux, J.P., Rodriguez, E., Martinerie, J.: The brainweb: phase synchronization and large-scale integration. *Nature Reviews Neuroscience* 2, 229–239 (2001)

7. Fries, P.: A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. *Trends in Cognitive Sciences* 9, 474–480 (2005)
8. Buzsaki, G., Draguhn, A.: Neuronal oscillations in cortical networks. *Science* 304, 1926–1929 (2004)
9. Mazoyer, B., Zago, L., Mellet, E., Bricogne, S., Etard, O., Houde, O., Crivello, F., Joliot, M., Petit, L., Tzourio-Mazoyer, N.: Cortical networks for working memory and executive functions sustain the conscious resting state in man. *Brain Research Bulletin* 54, 287–298 (2001)
10. Binder, J.R., Frost, J.A., Hammeke, T.A., Bellgowan, P.S.F., Rao, S.M., Cox, R.W.: Conceptual processing during the conscious resting state: A functional MRI study. *Journal of Cognitive Neuroscience* 11, 80–93 (1999)
11. Greicius, M.D., Krasnow, B., Reiss, A.L., Menon, V.: Functional connectivity in the resting brain: a network analysis of the default mode hypothesis. *Proceedings of the National Academy of Sciences* 100, 253–258 (2003)
12. Starzl, T.E., Whitlock, D.G.: Diffuse thalamic projection system in monkey. *J. Neurophysiol.* 15, 449–468 (1952)
13. Riecke, H., Roxin, A., Madruga, S., Solla, S.A.: Multiple attractors, long chaotic transients, and failure in small-world networks of excitable neurons. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 17, 26110 (2007)
14. Stratton, P., Wiles, J.: Self-sustained non-periodic activity in networks of spiking neurons: the contribution of local and long-range connections and dynamic synapses. *NeuroImage* 52, 1070–1079 (2010)
15. Izhikevich, E.M.: Simple model of spiking neurons. *IEEE Transactions on Neural Networks* 14, 1569–1572 (2003)
16. Wang, X.J.: Synaptic Basis of Cortical Persistent Activity: the Importance of NMDA Receptors to Working Memory. *Journal of Neuroscience* 19, 9587–9603 (1999)
17. Markram, H., Wang, Y., Tsodyks, M.: Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences* 95, 5323–5328 (1998)

Mutual Information Analyses of Chaotic Neurodynamics Driven by Neuron Selection Methods in Synchronous Exponential Chaotic Tabu Search for Quadratic Assignment Problems

Tetsuo Kawamura¹, Yoshihiko Horio¹, and Mikio Hasegawa²

¹ Graduate School of Engineering, Tokyo Denki University,

2-2 Kanda-Nishiki-cho, Chiyoda-ku, Tokyo 101-8457, Japan

² Department of Electrical Engineering, Tokyo University of Science, 1-14-6
Kudankita, Chiyoda-ku, Tokyo, 102-0073, Japan

09kme17@ms.dendai.ac.jp, horio@eee.dendai.ac.jp, hasegawa@ee.kagu.tus.ac.jp

Abstract. The exponentially decaying tabu search, which exhibits high performance in solving quadratic assignment problems (QAPs), has been implemented on a neural network with chaotic neurodynamics. To exploit the inherent parallel processing capability of analog hardware systems, a synchronous updating scheme, in which all neurons in the network are updated simultaneously, has also been proposed. However, several neurons may fire simultaneously with the synchronous updating. As a result, we cannot determine only one candidate for the 2-opt exchange from among the many fired neurons. To solve this problem, several neuron selection methods, which select a specific neuron from among the fired neurons, have been devised. These neuron selection methods improved the performance of the synchronous updating scheme; however, the dynamics of the chaotic neural network driven by these heuristic algorithms cannot be intuitively understood. In this paper, we analyze the dynamics of a chaotic neural network driven by the neuron selection methods by considering the spatial and temporal mutual information.

Keywords: mutual information, high-dimensional chaotic dynamics, chaotic neural network, QAP, tabu search.

1 Introduction

The tabu search [1] has been implemented on a chaotic neural network [2–4], where exponentially decaying tabu effects are naturally realized by the refractoriness of the chaotic neuron model. In this approach, the 2-opt algorithm drives chaotic neurodynamics; that is, 2-opt exchanges are carried out according to the firings of the corresponding neurons. This chaotic exponential tabu search exhibits high performance in solving the traveling salesman problem and quadratic assignment problems (QAPs) [2,4,5].

We have constructed a hardware system with a mixed analog/digital architecture to implement the chaotic exponential tabu search [6]. Then, experimental results confirmed the high performance of the hardware system.

In order to exploit inherent parallel processing capability of the analog hardware system, we have proposed a synchronous updating scheme, in which all the neurons in the network are updated simultaneously [78]. The original chaotic tabu search algorithm sequentially updates the neurons. Consequently, substantial time is required for large-scale QAPs. In contrast, with the synchronous updating scheme, it is possible to increase the processing speed; furthermore, the execution time does not depend on the size of the QAP.

In the chaotic exponential tabu search, we use the 2-opt algorithm as a local search method. The exchange of elements in the 2-opt algorithm is governed by the firing states of corresponding neurons: if the (i, j) th neuron in the network fires, the (i, j) th element is exchanged, as described in Section 3. However, through the synchronous updating, several neurons may fire simultaneously. As a result, we cannot determine only one candidate for the 2-opt exchange from among the many fired neurons. To solve this problem, we have proposed several neuron selection methods. Furthermore, experimental results confirmed the high performance of the neuron selection methods [9].

However, the algorithm of the best neuron selection method cannot be understood intuitively or logically. Therefore, we consider the high problem-solving performance to involve high-dimensional chaotic dynamics. However, it is not yet clear what aspect of the chaotic neurodynamics contributes to the effectiveness.

Therefore, in the present paper, we analyze the dynamics of a chaotic neural network driven by neuron selection methods by considering spatial and temporal mutual information. The large-scale chaotic neural network has high-dimensional chaotic dynamics, which suffer from large external effects from the neuron selection methods and 2-opt exchanges. Thus, it is difficult to use evaluation methods for low-dimensional dynamical systems such as Lyapunov spectrum and fractal dimension analyses. Therefore, we use mutual information, with which the spatio-temporal information transfers can be evaluated quantitatively, even for high-dimensional dynamical systems. In addition, the calculation of mutual information requires only binary variables, thus avoiding high-precision multi-channel measurements of analog variables from the hardware.

2 Quadratic Assignment Problem

The QAP is an NP-hard combinatorial optimization problem [5]. A size- N QAP is defined by two $N \times N$ matrixes, \mathbf{A} and \mathbf{B} [5]. The matrix \mathbf{A} denotes mutual distances among N indices, while \mathbf{B} denotes relationships among N units. We employ a standard formulation to express a possible solution of the QAP using the permutation \mathbf{p} , which expresses a set of assignments of the N units to the N indices, as shown in Fig. 1 [10]. The cost function $F(\mathbf{p})$ of the QAP, which we seek to minimize, is defined as

$$F(\mathbf{p}) = \sum_{i=1}^N \sum_{j=1}^N a_{ij} b_{p(i)p(j)}, \quad (1)$$

$$\text{index: } 1 \dots j \dots q(i) \dots n \quad \text{index: } 1 \dots j \dots q(i) \dots n$$

$$\mathbf{p}:(p(1), \dots, p(j), \underbrace{\dots, i, \dots, p(n)}) \rightarrow \mathbf{p}:(p(1), \dots, i, \dots, p(j), \dots, p(n))$$

Fig. 1. The (i, j) -assignment and the $(p(j), q(i))$ -assignment resulting from the firing of the (i, j) th neuron

where a_{ij} is the (i, j) th element of \mathbf{A} , and $b_{p(i)p(j)}$ is the $(p(i), p(j))$ th element of \mathbf{B} .

3 Synchronous Exponential Tabu Search

For a size- N QAP, we use a neural network composed of $N \times N$ chaotic neurons. If the (i, j) th neuron in the network fires, the element i of \mathbf{p} is assigned to the index j , as shown in Fig. 1. At the same time, the element $p(j)$ is assigned to the index $q(i)$. These exchanges are referred to as the (i, j) -assignment and $(p(j), q(i))$ -assignment, respectively [3]. In the sequential updating, the internal state of each neuron is updated one by one from the $(1, 1)$ th neuron in the network to the (N, N) th neuron [23]. Therefore, while one neuron is updating, other neurons cannot be updated. Thus, sequential updating is not suitable for parallel processing. In order to fully exploit the inherent advantage of analog hardware systems, a synchronous updating scheme has been proposed [8]. Furthermore, the time required for synchronous updating does not depend on the size of the QAP.

The chaotic dynamics of the (i, j) th neuron in the network is defined as follows [7]:

$$\xi_{ij}(t+1) = \beta(F_1(t) - F_{ij}(t)), \quad (2)$$

$$\eta_{ij}(t+1) = k_f \eta_{ij}(t) - \alpha_\eta X_{p(j)q(i)}(t) + R + \epsilon_1, \quad (3)$$

$$\zeta_{ij}(t+1) = k_r \zeta_{ij}(t) - \alpha_\zeta x_{ij}(t) + R + \epsilon_2, \quad (4)$$

$$y_{ij}(t+1) = \xi_{ij}(t+1) + \eta_{ij}(t+1) + \zeta_{ij}(t+1), \quad (5)$$

$$x_{ij}(t+1) = f(y_{ij}(t+1)), \quad (6)$$

where the internal state $\xi_{ij}(t+1)$ gives a gain in the cost at discrete time $t+1$, $F_1(t)$ is the current cost, $F_{ij}(t)$ is the cost after the (i, j) -assignment, β is a scaling parameter, the internal state $\eta_{ij}(t+1)$ expresses the tabu effect for the $(p(j), q(i))$ -assignment, the internal state $\zeta_{ij}(t+1)$ represents the tabu effect for the (i, j) -assignment, α_η and α_ζ are scaling parameters for tabu effects, k_r and k_f are decay parameters for tabu effects, R is an external bias, ϵ_1 and ϵ_2 are terms corresponding to dynamical noise in the hardware, $y_{ij}(t+1)$ is the total internal state, $x_{ij}(t+1)$ is the neuronal output, and $f(\cdot)$ is a monotonically increasing nonlinear output function of the neuron. Furthermore, $X_{p(j)q(i)}(t)$ is a binary variable for hardware implementation. When the $(p(j), q(i))$ -assignment is carried out, $X_{p(j)q(i)}(t)$ is set to “1,”, otherwise “0.”

4 Neuron Selection Methods

In the synchronous updating scheme, $y_{ij}(t)$ for all i and j are updated simultaneously. We define the (i, j) th neuron to fire if $y_{ij}(t) \geq 0$. Then, the 2-opt exchange corresponding to the (i, j) th neuron is carried out. However, more than one neuron may fire simultaneously with the synchronous updating. Therefore, we cannot determine a candidate for the 2-opt exchange. To solve this problem, neuron selection methods (Method-A and Method-B), which select a specific neuron among the many fired neurons, were proposed [7][11]. Method-A and Method-B are briefly reviewed in the following subsections. In addition, Method-C, which is used in the simulations in section 6, is described.

4.1 Neuron Selection Method-A

We refer to the neuron selection method proposed in [7] as “Method-A.” The algorithm of Method-A is as follows:

- a) The internal states of all neurons in the network are simultaneously updated.
- b) Fired neurons are sorted according to the values of $y_{ij}(t)$ in descending order. From among these, the first U neurons are selected, where U is an integer parameter.
- c) The current permutation $\mathbf{p}_{current}$ is tentatively updated using the neuron with the largest internal state, that is, $U = 1$. We denote the resulting tentative permutation as $\mathbf{p1}$. We also calculate the cost $F(\mathbf{p1})$ for $\mathbf{p1}$. Note that this tentative update of the permutation does not actually alter $\mathbf{p}_{current}$ and the internal states of the neurons.
- d) Next, the current permutation $\mathbf{p}_{current}$ is tentatively updated using the neuron with the second-largest internal state, that is, $U = 2$. The resulting tentative permutation is denoted as $\mathbf{p2}$. Moreover, we calculate $F(\mathbf{p2})$ for $\mathbf{p2}$.
- e) The above process is repeated for U neurons.
- f) From step b) to e), U different costs, $F(\mathbf{p1}), F(\mathbf{p2}), \dots, F(\mathbf{p}U)$, corresponding to U tentative permutations $\mathbf{p1}, \mathbf{p2}, \dots, \mathbf{p}U$, respectively, are obtained. Among the U neurons, the neuron that incurs the smallest cost is chosen. Finally, the current permutation $\mathbf{p}_{current}$ is “actually” updated using this neuron.

4.2 Neuron Selection Method-B

In [11], Method-A was modified to improve performance. This method is referred to as “Method-B” in this paper. This method exhibits high performance in solving QAPs [9]. However, it is difficult to intuitively understand why this ad hoc heuristic method performs well.

- a)**b)and c)** Steps a) to c) are the same as in Method-A.
- d) Next, the “permutation $\mathbf{p1}$ ” is updated using the neuron with the second-largest internal state. The resulting permutation is denoted as $\tilde{\mathbf{p}}2$. Moreover, we calculate $F(\tilde{\mathbf{p}}2)$ for $\tilde{\mathbf{p}}2$.

- e) The above process is repeated for U neurons.
- f) From steps b) to e), we obtain U different costs, $F(\mathbf{p1})$, $F(\tilde{\mathbf{p2}})$, ..., $F(\tilde{\mathbf{p}}U)$, corresponding to U tentative permutations, $\mathbf{p1}$, $\tilde{\mathbf{p2}}$, ..., $\tilde{\mathbf{p}}U$, respectively. Among U neurons, the neuron that incurs the smallest cost is chosen. Finally, the “current permutation $\mathbf{p}_{current}$ ” is actually updated using this neuron.

4.3 Neuron Selection Method-C

For reference, we use a straightforward neuron selection method, Method-C, the mechanism of which can be intuitively understood.

- a)b)c)d)and e) Steps a) to e) are the same as in Method-B.
- f) From steps b) to e), U different costs, $F(\mathbf{p1})$, $F(\tilde{\mathbf{p2}})$, ..., $F(\tilde{\mathbf{p}}U)$, corresponding to U tentative permutations, $\mathbf{p1}$, $\tilde{\mathbf{p2}}$, ..., $\tilde{\mathbf{p}}U$, respectively, are obtained. The permutation $\tilde{\mathbf{p}}i (1 \leq i \leq U)$ that incurs the smallest cost replaces $\mathbf{p}_{current}$.

In Method-A and Method-B, the 2-opt exchange is applied only once to $\mathbf{p}_{current}$ throughout steps a) to f). In contrast, in Method-C, $\tilde{\mathbf{p}}i$ will be the final permutation in step f). That is, 2-opt exchanges are applied i times to $\mathbf{p}_{current}$. Therefore, the resulting permutation can differ greatly from $\mathbf{p}_{current}$ for large i , and Method-C can be used to explore the solution space. Accordingly, we use Method-C as a reference in the simulations.

5 Mutual Information for Chaotic Neural Network

Dynamics of the chaotic neural networks driven by the three above-mentioned neuron selection methods are evaluated by considering the spatial and temporal mutual information defined below.

5.1 Spatial Mutual Information

The average spatial mutual information MI_{sp} is given as

$$MI_{sp} = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N MI_{ij}, \quad (7)$$

$$\begin{aligned} MI_{ij} &= H(x_i(t_n)) - H(x_i(t_n) \setminus x_j(t_{n-1})), \\ &= \sum_{k=1}^2 \sum_{l=1}^2 P(x_i(t_n)_k, x_j(t_{n-1})_l) \times \log_2 \frac{P(x_i(t_n)_k, x_j(t_{n-1})_l)}{P(x_i(t_n)_k)P(x_j(t_{n-1})_l)}, \end{aligned} \quad (8)$$

where N is the number of neurons, MI_{ij} is the spatial mutual information of the i th neuron obtained from the j th neuron, $H(x_i(t_n))$ is the entropy at time t_n , $H(x_i(t_n) \setminus x_j(t_{n-1}))$ is the conditional entropy, $P(x_i(t_n)_k)$ is occurrence probability of the k th event of the i th neuron, and $P(x_i(t_n)_k, x_j(t_{n-1})_l)$ is the co-occurrence probability. The neuron has two states: firing ($k = 1$) and non-firing ($k = 2$).

5.2 Temporal Mutual Information

The average temporal mutual information MI_{tmp} [12] is given as

$$MI_{tmp} = \frac{1}{N} \sum_{i=1}^N MI_i, \quad (9)$$

$$\begin{aligned} MI_i &= H(x_i(t_n)) - H(x_i(t_n) \setminus x_i(t_{n-1})), \\ &= \sum_{k=1}^2 \sum_{l=1}^2 P(x_i(t_n)_k, x_j(t_{n-1})_l) \times \log_2 \frac{P(x_i(t_n)_k, x_j(t_{n-1})_l)}{P(x_i(t_n)_k)P(x_j(t_{n-1})_l)}, \end{aligned} \quad (10)$$

where MI_i is the temporal mutual information of the i th neuron.

6 Simulation Results

Using numerical simulations, we solve several QAP benchmarks to determine what dynamics contribute to efficient solution searching, considering spatial and temporal mutual information. We use the benchmark problems Had20, Lipa20b, and Nug30 from the QAP library [5]. In the simulations, we fixed $\alpha_\zeta = 1.0$ and $R = 0.1$. On the other hand, α_η , β , k_r , and k_f , are selected from $\{0.1, 0.3, 0.4, 0.5, 0.65, 0.7, 0.75, 0.9, 0.95, 1.0\}$. Then, we derived 20 parameter sets, with which the arrival rate for the optimal solution and the average gap from the optimal solution show good values. These parameter sets were used to evaluate MI_{sp} and MI_{tmp} . The arrival rate for the optimal solution, AR , averaged over 100 trials with different initial conditions is defined as

$$AR = \frac{ON}{100} \times 100 [\%], \quad (11)$$

where ON is the number of optimal solutions obtained throughout the trials. On the other hand, the average gap from the optimal solution, AG , is given by

$$AG = \frac{\sum_{n=1}^{100} EM_n}{100} \times 100, [\%] \quad (12)$$

where EM_n is the gap from the optimal solution for the n th trial.

We use various values of U : 1, 2, 3, 4, and 5. Note here that all methods select the neuron with the largest internal state only when $U = 1$. In other words, when $U = 1$, all methods are equivalent to the method in [7]. We hence denote methods with $U = 1$ as Method-1 hereafter. Moreover, ϵ_1 and ϵ_2 in Eqs. (3) and (4) represent independent random noise with maximum amplitude of ± 0.04 . The total number of iterations in one trial is 10,000. The iteration number is incremented when an actual exchange is done after a synchronous updating.

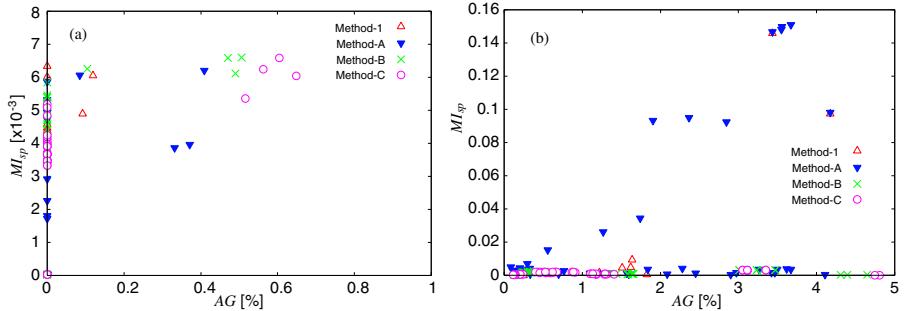
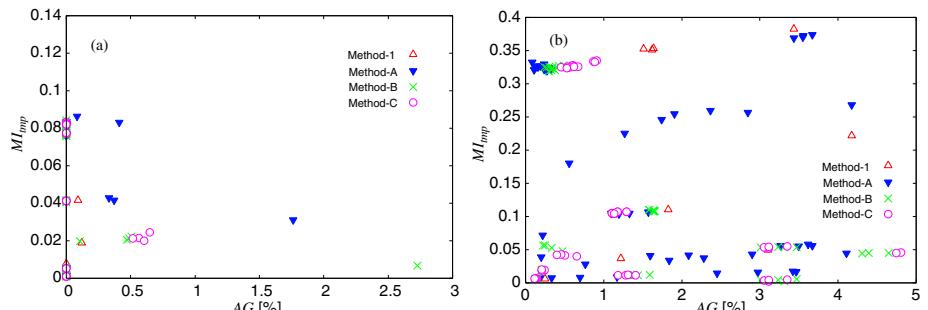
In the first simulation, we examine the solving performance of each neuron selection method. We fix $U = 5$ except for Method-1, and use the network parameters listed in Table 1. AR and AG are shown in Table 2 for each method. As shown in Table 2, Method-B and Method-C always perform better than

Table 1. Values of network parameters used in the simulations

Instance	α_η	β	k_r	k_f
Had20	0.1	0.7	0.7	0.7
Lipa20b	0.1	0.3	0.9	0.9
Nug30	0.1	0.7	0.7	0.7

Table 2. Arrival rate for the optimal solution AR , and average gap from the optimal solution AG

Instance	AR				AG			
	Method-1	Method-A	Method-B	Method-C	Method-1	Method-A	Method-B	Method-C
Had20	9	41	50	50	0.326	0.222	0.189	0.194
Lipa20b	100	100	100	100	0.000	0.000	0.000	0.000
Nug30	14	0	53	56	0.248	0.174	0.119	0.114

**Fig. 2.** Average spatial mutual information, MI_{sp} , versus average gap from the optimal solution, AG , for (a) Lipa20b and (b) Nug30**Fig. 3.** Average temporal mutual information, MI_{tmp} , versus average gap from the optimal solution, AG , for (a) Lipa20b and (b) Nug30

Method-1 and Method-A. Figure 2 shows the relationship between MI_{sp} and AG for the Lipa20b and Nug30 problems. As shown in Fig. 2, for small AG , MI_{sp} is also small. Moreover, when AG values are large, the MI_{sp} values for Method-A are distributed from small to large, especially in Fig. 2(b). In addition, the MI_{sp} values for Methods-B and -C have similar distributions.

Figure 3 shows the relationship between MI_{tmp} and AG for each method used to solve the Lipa20b and Nug30 problems. As shown in Fig. 3, MI_{tmp} takes particular values when the optimal solution is obtained, that is, when $AG = 0$. This result suggests that when the network performs well, MI_{tmp} takes particular values depending on the problem. For example, in Fig. 3(a), the MI_{tmp} values are concentrated around 0.01, 0.04, and 0.08 when $AG = 0$. Furthermore, in Fig. 3, the MI_{tmp} values for Methods-B and -C show similar trends for MI_{sp} . On the other hand, MI_{tmp} in the case of Method-A is distributed over a wide range when AG is large, as shown in Fig. 3(b).

In summary, the problem-solving performance of Method-A is worse than that of Method-B or Method-C. In addition, MI_{sp} and MI_{tmp} for Method-A are distributed over a wide range. In contrast, the MI_{sp} values for Method-B and Method-C are low when the problem-solving performance is high. Furthermore, for Methods-B and -C, MI_{tmp} takes particular values. In addition, MI_{sp} and MI_{tmp} of Method-B exhibit similar trends and distributions to those of Method-C. As a consequence, we conclude that the dynamics of Method-B is similar to that of Method-C. Therefore, the chaotic neural network driven by Method-B has the same search dynamics with global jumping in the solution space as Method-C, which contributes to the high problem-solving performance.

7 Conclusions

We analyzed the dynamics of the synchronous exponential chaotic tabu search driven by neuron selection methods by considering spatial and temporal mutual information. From the results, we qualitatively confirmed that, for high problem-solving performance, the average spatial and temporal mutual information should be low, and the average temporal mutual information should take particular values depending on the problem. In addition, the high performance of Method-B can be attributed to the global searching dynamics which explore the solution space by jumping.

This research was supported by JSPS through its Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program). and Kakenhi (20300085).

References

1. Taillard, E.: Robust tabu search for the quadratic assignment problems. *Parallel Computing* 17, 443–455 (1991)
2. Hasegawa, M., Ikeguchi, T., Aihara, K.: A novel chaotic search for quadratic assignment problems. *European Journal of Operational Research* 139, 543–556 (2002)

3. Hasegawa, M., Ikeguchi, T., Aihara, K.: Exponential and chaotic neurodynamical tabu searches for quadratic assignment problems. *Control and Cybernetics* 29(3), 773–788 (2000)
4. Hasegawa, M., Ikeguchi, T., Aihara, K.: Solving large scale traveling salesman problems by chaotic neurodynamics. *Neural Networks* 15, 271–283 (2002)
5. Burkard, R.E., Karisch, S.E., Rendl, F.: QAPLIB - A quadratic assignment problem library, <http://www.opt.math.tu-graz.ac.at/qaplib/>
6. Horio, Y., Aihara, K.: Analog computation through high-dimensional physical chaotic neuro-dynamics. *Physica D* 237, 1215–1225 (2008)
7. Yokota, N., Horio, Y., Aihara, K.: A synchronous exponential chaotic tabu search for quadratic assignment problems. In: Proc. Int. Workshop on Nonlinear Circuit and Signal Processing, CD-ROM (March 2008)
8. Yokota, N., Horio, Y., Aihara, K.: A modified synchronous exponential chaotic tabu search for quadratic assignment problems. IEICE Technical Report NLP107(561), 49–54 (2008) (in Japanese)
9. Kawamura, T., Horio, Y.: Evaluation of neuron selection techniques for synchronous exponential chaotic tabu search for quadratic assignment problems. IEICE Technical Report NLP109(269), 67–71 (2009) (in Japanese)
10. Hillier, F., Michael, M.: Quadratic assignment problem algorithms and the location of indivisible facilities. *Management Science, Series A, Sciences* 13(1), 42–57 (September 1966)
11. Shirakuma, A., Horio, Y., Aihara, K.: An improved neuron selection technique in an exponential chaotic tabu search hardware system for large-scale quadratic assignment problems. In: Proc. Int. Workshop on Nonlinear Circuit and Signal Processing, CD-ROM (March 2009)
12. Masanao, O., Yuba, K.: Associative memory and amount of mutual information in a chaotic neural network introducing function typed synaptic weights. In: Proc. of the 9th Int. Conf. on Neural Info. Processing, vol. 4, pp. 1617–1622 (November 2002)

A General-Purpose Model Translation System for a Universal Neural Chip

Francesco Galluppi*, Alexander Rast, Sergio Davies, and Steve Furber

School of Computer Science, University of Manchester, Kilburn Building,
Oxford Road, Manchester, M13 9PL, United Kingdom

{galluppf,rasta,daviess}@cs.man.ac.uk, steve.furber@manchester.ac.uk
<http://www.cs.manchester.ac.uk/>
<http://intranet.cs.man.ac.uk/apt/projects/SpiNNaker/>

Abstract. This paper describes how an emerging standard neural network modelling language can be used to configure a general-purpose neural multi-chip system by describing the process of writing and loading neural network models on the SpiNNaker neuromimetic hardware. It focuses on the implementation of a SpiNNaker module for PyNN, a simulator-independent language for neural networks modelling. We successfully extend PyNN to deal with different non-standard (eg. Izhikevich) cell types, rapidly switch between them and load applications on a parallel hardware by orchestrating the software layers below it, so that they will be abstracted to the final user. Finally we run some simulations in PyNN and compare them against other simulators, successfully reproducing single neuron and network dynamics and validating the implementation.

Keywords: neuromimetic, neuromorphic, hardware, PyNN.

1 Introduction

Simulation of large systems composed of spiking neurons can follow two distinct paths: simulation on standard computers, using proprietary neural code or standard neural network simulators (such as NEST [1], Brian [2] etc.); or simulation on dedicated hardware platforms. The advantages of using standard simulators are in interoperability and ease of use, since they abstract all the network instantiation and programming management from end users. This comes at the cost of efficiency [3]. Simulations on dedicated hardware on the other hand are more efficient but, due to the proprietary nature of dedicated hardware [4], writing a network model and running it becomes a task accessible only to the system creators, given that a normal user will not be willing to spend the time to learn the architecture and configuration modalities of a dedicated hardware system [5]. To simplify this process we develop a PyNN [6] plugin for the SpiNNaker system [7], a massively parallel dedicated chip for real time neural network simulations. We

* All the authors are with the APT group, School of Computer Science, The University of Manchester.

abstract the hardware from the end-user point of view by implementing a module for PyNN, a simulator-independent language for building neural networks. The system represents an attempt towards a universal neural network simulation system. The rest of the paper is structured as follows: Section 2 presents the application loading workflow for the SpiNNaker system, and the role of the PyNN module, Section 3 describes the implementation while section 4 presents some results and comparisons with other simulators. A discussion section presents current implications and future work, followed finally by some conclusions.

2 Loading Models on the SpiNNaker

SpiNNaker System. The SpiNNaker system presents a general-purpose scalable multi-chip multi-core platform for real-time parallel simulation of large-scale spiking neural systems [8]. Each SpiNNaker chip contains 18 identical ARM968 subsystems responsible for modelling various types of neural and synaptic dynamics, up to ~ 1000 neurons per core. Chips are interconnected using 6 asynchronous links. Spikes propagate in the system using a multicast routing mechanism through a packet-switched link fabric. Synaptic information is kept in an SDRAM chip in order to store synaptic information and simulation results.

While the system architecture offers a low-power, scalable, fully configurable platform for modelling neural networks, its innate parallelism and distribution of information raises challenges about application mapping, loading and running [9]. Information is distributed through different components and different chips of the SpiNNaker system. The network model needs to be translated into SpiNNaker data structures that will then be used to configure different parts, like the Tightly-coupled memory (TCM) of the ARM cores, the SDRAM and the Multicast Router of each chip.

Workflow. The workflow proposed to load neural network models on the SpiNNaker system is presented in figure 1. The module builds the network and extracts the information regarding the network structure and parameters. This is compiled by Initload, a mapping tool which translates networks and maps into SpiNNaker binary data structures [10]. Finally, the application is loaded and executed on the system. The PyNN.SpiNNaker module is also responsible for

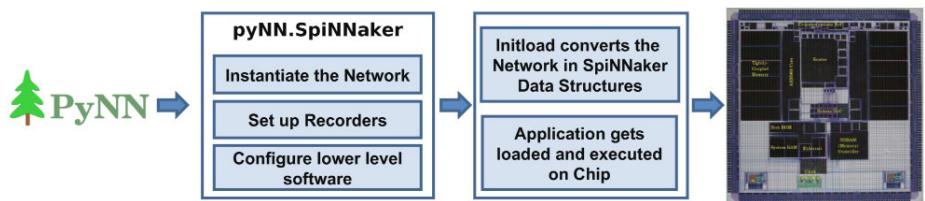


Fig. 1. a) Write a network model in PyNN (b) PyNN.SpiNNaker plugin builds the network and converts it into neuron and connection lists (c) Initload converts this into SpiNNaker data structures (d) application is loaded and started on chip

scripting the execution of low-level tools, in order to automate and abstract those configuration steps from the final user. Those steps include generating extra files about the system architecture and SpiNNaker specific parameters, dynamically calling low level compilers, interacting with the ethernet interface in order to load files on the chip, starting the simulation and retrieving simulation results.

3 Implementing the PyNN.SpiNNaker Module

Populations. SpiNNaker is a universal platform for spiking neurons in the sense that no neural model is embedded in the hardware, but applications that take advantage of the parallel distributed architecture and communication infrastructure can be developed, integrated and switched rapidly. Two neural models are currently available for SpiNNaker: an Izhikevich [11] and a Leaky Integrate-and-Fire (LIF) [12] module.

The PyNN module iterates through populations and extracts the parameters needed. Every neuron of a population is linearly assigned a unique ID, which the mapper will use to allocate and interconnect neurons in the system. The neural application along with state variables and parameters are placed in the TCM of each core, since the neural application continuously needs them in order to update neuron equations, process new inputs, plasticity etc. We use PyNN to model neurons as parameter place-holders, since the simulation will run on-chip. In this way, customization of PyNN with non-standard models can be very rapid, and PyNN can be used as a configuration interface for dedicated hardware.

Connections. Connections are configured in two different places into the SpiNNaker system. Routing tables route the spikes from the pre- to the post-synaptic neuron, while synaptic information (weight, delay, etc.) is stored in the SDRAM of the post-synaptic neuron(s) [8]. Routing tables and memory maps for each chip are generated by initLoad starting from a list of connections and their parameters extracted from the PyNN network.

Inputs. We consider two kinds of input: current sources and spike generator groups. Current sources are extracted from the network, compiled, loaded into the SDRAM of each SpiNNaker chip, and then executed at runtime. Spike generator groups can be modelled in different ways: by loading a list of spikes in the SpiNNaker chip, by using an external component which will inject spikes at runtime or by building routines that will generate spikes internally. The SpiNNaker system runs simulation at real-time speed, making it possible to implement interactive applications within Python using the on-board ethernet or AER interfaces to inject spikes at real-time.

SpiNNaker Parameters. We use a PyNN setup call in order to pass additional system parameters to SpiNNaker. Such parameters could include the number of x/y chips in the machine, the SpiNNaker chip IP address, number of neurons assigned for each core, etc.

4 Simulations and Results

4.1 LIF Neuron

We tested the implementation of the LIF SpiNNaker module described in [12] by modelling a single LIF neuron. Figure 2 displays the results from injecting it with a DC current source strong enough to make it exceed the threshold potential (-55 mV) and fire 3 times in the interval considered. We simulated the same script with other two simulators supported by PyNN (Brian and NEST) in order to compare results. As shown in figure, membrane potential dynamics are very close in all three simulations; the difference in timing after a spike can be considered an effect of the fact that refractory periods are not explicitly modelled on the SpiNNaker LIF module. This can also be observed in figure 2(b), where activity rate is plotted against amplitude of the input current (normalized to the membrane time constant). Number of spikes generated per second is comparable between NEST and SpiNNaker, with SpiNNaker producing spikes more frequently because of the absence of a refractory period.

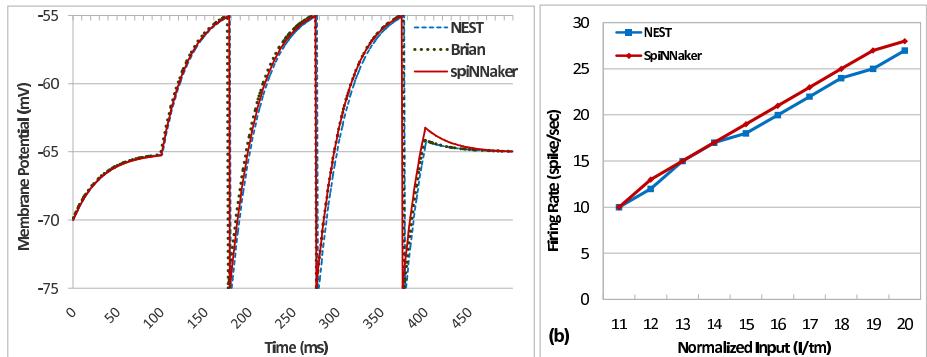


Fig. 2. (a) Results of a single LIF neuron compared with PyNN.nest and PyNN.brian
 (b) Comparison between firing rates

4.2 Izhikevich Neuron

Simulation of the Izhikevich neuron required extending PyNN to support a new neuron model - Izhikevich's. Since the simulation runs on hardware, no further specification/modelling is required at this stage in PyNN, which is used purely as a configuration tool for lower level hardware. This approach could easily be extended to any other neural model running on hardware that relies on parameters configuration. Since the Izhikevich neuron is not a standard cell type in PyNN, we used a Brian script to test our implementation. Figure 3 that our fixed-point implementation of the Izhikevich neuron follows the dynamics obtained with Brian. Results of the simulation show that membrane potential dynamics of a single Izhikevich RS neuron injected with a DC source current are similar despite

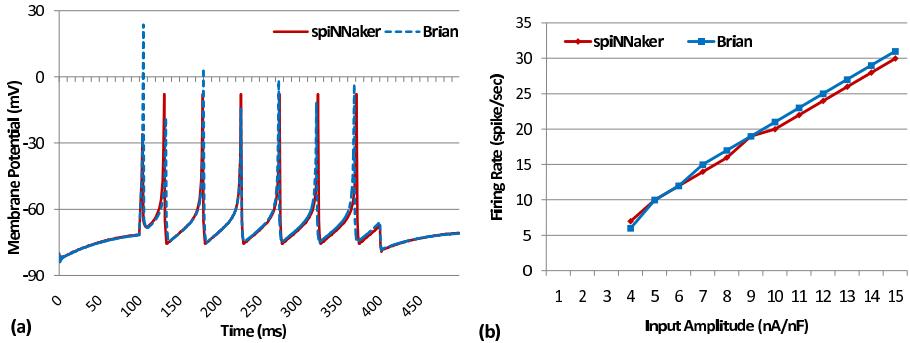


Fig. 3. Results of a single neuron compared with Brian

the fixed point implementation used in the SpiNNaker model, and firing rates are qualitatively the same.

4.3 Oscillatory Network Dynamics

In order to test network dynamics we implemented a network inspired by the fourth benchmark proposed in [3], using a PyNN script from ModelDB [13] in order to verify consistency of the results and compare them between different simulators. The network is modelled by 500 randomly interconnected excitatory and inhibitory (ratio 4:1) LIF neurons which interact through voltage deflections [14]. In order to compare results against other simulators we set synaptic time constants to be very short, so as to mimic the voltage-jump synapses used by the SpiNNaker LIF module. We run the simulation compared the results of the simulation with the one produced by the PyNN.nest module.

```

n_exc = 400      # number of excitatory cells
n_inh = 100      # number of inhibitory cells
cell_params = {
    'tau_m'      : 32,      'cm'          : 1,      'v_init'       : -60,
    'v_rest'     : -49,     'v_reset'     : -60,     'v_thresh'    : -50,
    'tau_syn_E'  : 1,       'tau_syn_I'   : 1,      'tau_refrac'  : 1
}

```

Neurons are modelled as above. They are interconnected with a 2% probability. Other network parameters are inspired by [3] to compare results (weights 0.25 mV and -2.25 for excitatory and inhibitory synapses, connection delay randomly distributed in the interval [1,14] msec).

The results of the simulation are presented in figure 4. As neurons cross the threshold potential they start spiking, building a background activity through random connections. We inject a current in a subset of the population to increase its activity at 1/3 of the simulation and double it at 2/3 (observed as the patch of neurons with high activity in figure). Despite choosing a standard cell in PyNN (exponential current-based LIF neuron with first-order synapse kinetics) whose

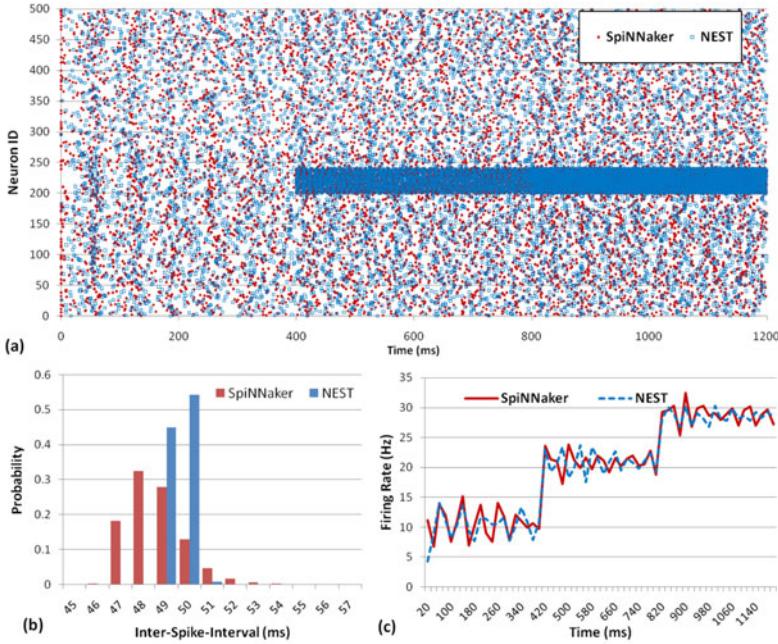


Fig. 4. Simulation with 500 neurons: comparison between NEST and SpiNNaker. (a) Raster Plot for the entire simulation (b) Inter-Spike-Interval (ISI) Distribution (c) Mean activity rate.

synapse dynamics SpiNNaker does not model, a comparison of raster plots and mean population frequency rates usign the PyNN.nest module shows that the qualitative behaviour of the network is maintained; the same number of spikes are produced (difference in the order of 2%) and the inter-spike interval distributions have their peaks close (48-50 msec) with their distance comparable to the 1 ms sampling rate, although the SpiNNaker distribution has a wider spread that that obtained with NEST.

5 Discussion

We have developed a PyNN interface to load models on the SpiNNaker system and shown how it can be used to handle network translation in specific hardware data files. Likewise it can manage software and data loading and distribution on a massively parallel machine, even for non-standard models like Izhikevich's. Finally it has been used to validate the neural implementation on SpiNNaker. Future work includes implementation and integration of synaptic dynamics with PyNN. One goal is to implement standard neural models and verify them against standard simulators. Another is to be able to create new

models which can easily be configured with PyNN. In contrast to previous systems [4], the SpiNNaker/PyNN combination lets the user configure software at both ends. As new computational units are modelled inside the system, PyNN can be used to organize their relationships and dynamically configure them on board. Users will have the ability to work in a standard environment, producing files that can easily be interpreted and exchanged within the community.

6 Conclusions

The continuous introduction of new neural, network, connectivity and plasticity models, while giving a wide range of research possibilities, poses challenges on how to share and standardize work between different groups and areas. A universal hardware simulator must be able to absorb such developments. On the other hand PyNN is emerging as a standard neural network modelling language which can encourage sharing of models with the possibility to run them on hardware. In this respect we have shown the advantages of using and extending a standard neural modelling tool such as PyNN to configure a universal neural network hardware system such as SpiNNaker. SpiNNaker, with its innate parallelism and asynchronous packet-based communication system offers an efficient platform for large neural network modelling. It is a universal platform in the sense that it imposes no particular neural model or neural network topology, but is fully configurable by the end user. The advantage of using general-purpose hardware coupled with PyNN is that as new models emerge, both the configuration interface and the simulation code run on the hardware can be extended to accommodate them, increasing the longevity and re-usability of the hardware system and exploiting the extensibility of an open-source standard modelling language.

Acknowledgements. We would like to thank the Engineering and Physical Sciences Research Council (EPSRC), Silistix, and ARM for support of this research.

References

1. Plessner, H.E., Eppler, J.M., Morrison, A., Diesmann, M., Gewaltig, M.O.: Efficient Parallel Simulation of Large-Scale Neuronal Networks on Clusters of Multi-processor Computers. In: Proc. 13th Int'l Euro-Par Conf. on Parallel Processing (Euro-Par 2007), pp. 672–681 (2007)
2. Goodman, D.: Brian: a simulator for spiking neural networks in python. Frontiers in Neuroinformatics 2 (2008)
3. Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J.M., Diesmann, M., Morrison, A., Goodman, P.H., Harris, F.C., Zirpe, M., Natschlger, T., Pecevski, D., Ermentrout, B., Djurfeldt, M., Lansner, A., Rochel, O., Vieville, T., Muller, E., Davison, A.P., Boustanli, S.E., Destexhe, A.: Simulation of networks of spiking neurons: a review of tools and strategies. J. Comput. Neurosci. 23(3), 349–398 (2007)

4. Brüderle, D., Müller, E., Davison, A., Muller, E., Schemmel, J., Meier, K.: Establishing a Novel Modeling Tool: A Python-based Interface for a Neuromorphic Hardware System. *Frontiers in Neuroinformatics* 3(17), 1–10 (2009)
5. Steinkraus, D., Buck, I., Simard, P.Y.: Using GPUs for machine learning algorithms. In: Proc. 8th Int'l Conf. Document Analysis and Recognition, pp. 1115–1120 (2005)
6. Davison, A.P., Brüderle, D., Eppler, J., Kremkow, J., Muller, E., Pecevski, D., Perrinet, L., Yger, P.: Pynn: A common interface for neuronal network simulators. *Front Neuroinformatics* 2, 11 (2008)
7. Furber, S., Brown, A.: Biologically-inspired massively-parallel architectures - computing beyond a million processors. In: ACSD 2009: Proceedings of the 2009 Ninth International Conference on Application of Concurrency to System Design, Washington, DC, USA, pp. 3–12. IEEE Computer Society, Los Alamitos (2009)
8. Rast, A.D., Jin, X., Galluppi, F., Plana, L.A., Patterson, C., Furber, S.: Scalable event-driven native parallel processing: the spinnaker neuromimetic system. In: CF 2010: Proceedings of the 7th ACM international conference on Computing frontiers, pp. 21–30. ACM, New York (2010)
9. Khan, M.M., Rast, A.D., Navaridas, J., Jin, X., Plana, L., Luján, M., Temple, S., Patterson, C., Richards, D., Woods, J.V., Miguel-Alonso, J., Furber, S.B.: Event-Driven Configuration of a Neural Network CMP System over an Homogeneous Interconnect Fabric. *Parallel Computing* 30(7), 435–444 (2010)
10. Jin, X., Galluppi, F., Patterson, C., Rast, A., Davies, S., Temple, S., Furber, S. B.: Algorithm and software for simulation of spiking neural networks on the multi-chip spinnaker system. In: IEEE International Joint Conference on, Neural Networks, IJCNN 2008 (IEEE World Congress on Computational Intelligence) (September 2010)
11. Jin, X., Furber, S.B., Woods, J.V.: Efficient modelling of spiking neural networks on a scalable chip multiprocessor. In: IEEE International Joint Conference on, Neural Networks, IJCNN 2008 (IEEE World Congress on Computational Intelligence), pp. 2812–2819 (September 2008)
12. Rast, A., Galluppi, F., Jin, X., Furber, S.B.: The leaky integrate-and-fire neuron: A platform for synaptic model exploration on the spinnaker chip. In: IEEE International Joint Conference on, Neural Networks, IJCNN 2008 (IEEE World Congress on Computational Intelligence) (September 2010)
13. Migliore, M., Morse, T., Davison, A., Marenco, L., Shepherd, G., Hines, M.: ModelDB: making models publicly accessible to support computational neuroscience. *Neuroinformatics* 1(1), 135–139 (2003)
14. Vogels, T.P., Abbott, L.F.: Signal propagation and logic gating in networks of integrate-and-fire neurons. *J. Neurosci.* 25(46), 10786–10795 (2005)

Realizing Ideal Spatiotemporal Chaotic Searching Dynamics for Optimization Algorithms Using Neural Networks

Mikio Hasegawa

Tokyo University of Science, Tokyo 102-0073, Japan

Abstract. This paper proposes an optimization algorithm, which utilizes ideal spatiotemporal chaotic dynamics for solution search in a high dimensional solution space. Such chaotic dynamics is generated by the Lebesgue spectrum filter, which has been applied to the chaotic CDMA in previous researches to minimize the cross-correlation among the sequences. In the proposed method, such a filter is applied to the output functions of optimization neural networks to realize an ideal chaotic search, which generates ideally complicated searching dynamics. The proposed scheme is applied to two combinatorial optimization approaches, the Hopfield-Tank neural network with additive noise and a heuristic algorithm driven by neural networks, which solve the Traveling Salesman Problems and the Quadratic Assignment Problems. The simulation results show that the proposed approach using the ideal chaotic dynamics simply improves the performance of the chaotic algorithms without searching appropriate parameter values even for large-scale problems.

1 Introduction

Chaotic dynamics have been shown effective for combinatorial optimization problems by many researches [1][2][3][4][5][6][7][8]. There are two major approaches using chaotic dynamics to avoid trapping at undesirable local minimum solutions. The first one introduces the chaotic fluctuation to the Hopfield-Tank neural network [9], and the second one drives local search heuristics by the chaotic neural networks [10]. Although the first approach is applicable only to very small toy problems, the second approach can solve much more difficult and large-scale problems by introducing simple neighboring solution search heuristics and has been shown more effective than the conventional heuristic algorithms, such as the stochastic searches and the tabu searches [3][4][5].

In the previous researches, effectiveness of such chaotic dynamics for combinatorial optimization has been analyzed, and several important features of the chaotic searches have been found [6][8]. In the approach that adds chaotic sequences to each neuron in the Hopfield-Tank neural network solving optimization problems [7], it has been clarified that the most important factor for high performance of the chaotic noise is specific autocorrelation function of the chaotic dynamics, by an analysis based on the method of surrogate data [8]. This effective

chaotic noise has the autocorrelation with a negative value in lag 1 and damped oscillation. Such chaotic dynamics with negative autocorrelation has been also utilized in the chaotic CDMA researches [11][12]. The chaotic dynamics, whose autocorrelation of the sequences becomes $C(\tau) \approx C \times r^\tau$, $r = -(2 - \sqrt{3})$, is used because the cross-correlation among the sequences becomes smallest by such autocorrelation. In Ref. [13], effects of such sequences, which has negative autocorrelation with damped oscillation, has been applied also to the combinatorial optimization problems, and it has been shown that the noise sequences having negative autocorrelation improves the performance of asynchronously updated neural networks to the same level as those with the chaotic noise. Since such negative autocorrelation noise minimizes cross-correlation as shown in chaotic CDMA research, lower cross correlation may be also important for realizing complex spatiotemporal solution search in combinatorial optimization problems.

In the chaotic CDMA [12], such ideal chaotic noise, whose autocorrelation is $C(\tau) \approx C \times r^\tau$, $r = -(2 - \sqrt{3})$ has been generated by a FIR filter called Lebesgue spectrum filter. It has been shown that such a filter make lower cross-correlation among the sequence and improves the bit error rate of the CDMA communication system for various code sequences, such as stochastic sequences and deterministic or chaotic sequences.

In this paper, the Lebesgue spectrum filter making negative autocorrelation is applied to the combinatorial optimization methods, for realizing ideal spatiotemporal searching dynamics. The proposed algorithm minimizes cross-correlation among the solution update dynamics on each axis in a high dimensional solution space, and realizes ideally complex spatiotemporal searching dynamics. The proposed scheme is applied to two optimization approaches, the Hopfield-Tank neural networks with additive noise [9] and the 2-opt heuristic method driven by a neural network [3], solving the Traveling Salesman Problems (TSPs) and the Quadratic Assignment Problems (QAPs). Effectiveness of the proposed approach is investigated also for the large-scale problems up to 1173-city TSP.

2 Optimization Performance of Mutually Connected Neural Networks with Additive Chaotic Dynamics

First, the effectiveness of the ideal chaotic dynamics is evaluated in the Hopfield-Tank neural network approach. This approach is based on the minimization of the energy function of the neural networks by asynchronous update of each neuron. However, since the original Hopfield-Tank neural networks stop search at a local minimum, the chaotic noise and other stochastic dynamics has been added to the neurons to avoid trapping at such undesirable states and to achieve much higher performance [7][8].

In this paper, such a neuronal update function with additive noise is defined as follows,

$$x_{ik}(t+1) = f \left[\sum_{j=1}^N \sum_{l=1}^N w_{ikjl} x_{jl}(t) + \theta_{ik} + \beta z_{ik}(t) \right], \quad (1)$$

where $x_{ik}(t)$ is the output of the (i, k) th neuron at time t , w_{ikjl} is the connection weight between the (i, k) th and (j, l) th neurons, θ_{ik} is the threshold of the (i, k) th neuron, N is the number of cities, $z_{ij}(t)$ is a noise sequence added to the (i, j) th neuron, β is the amplitude of the noise, and f is the sigmoidal output function, $f(y) = 1/(1 + \exp(-y/\epsilon))$. The noise sequence used for $z_{ij}(t)$ is normalized to zero mean and unit variance.

To apply this neural network to the TSP, the connection weights w_{ijkl} and the thresholds θ_{ij} are set as the follows,

$$w_{ijkl} = -A\{\delta_{ij}(1 - \delta_{kl}) + \delta_{kl}(1 - \delta_{ij})\} - Bd_{ij}(\delta_{lk+1} + \delta_{lk-1}), \quad (2)$$

$$\theta_{ij} = 2A, \quad (3)$$

where d_{ij} is the distance between the cities i and j , A and B are the weight of the constraint term (formation of a closed tour) and the objective term (minimization of total tour length), and δ_{ij} is the Kronecker delta, respectively.

For solving the QAPs, whose objective function is

$$F(\mathbf{p}) = \sum_{i=1}^N \sum_{j=1}^N a_{ij} b_{p(i)p(j)}, \quad (4)$$

By the Hopfield-Tank neural networks, the connection weights and the thresholds have to be set as follows,

$$w_{ijkl} = -A\{\delta_{ij}(1 - \delta_{kl}) + \delta_{kl}(1 - \delta_{ij})\} - Ba_{ij}b_{kl}, \quad (5)$$

$$\theta_{ij} = 2A. \quad (6)$$

Fig. 1 shows the solvable performances of the above neural networks applied to the TSP and the QAP, with various noise sequence for $z_{ij}(t)$, such as the white Gaussian noise, the Chebyshev map chaos with the same dimension for

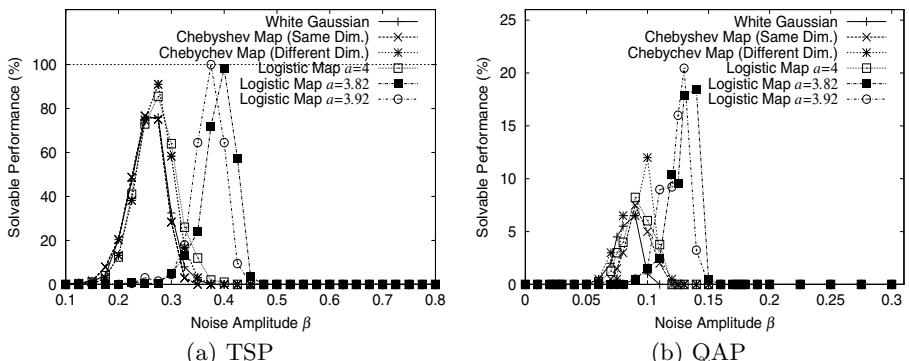


Fig. 1. Solvable performance of the neural networks with the stochastic noise and the chaotic noise on a 20-city TSP and a QAP with 12 nodes (Nug12)

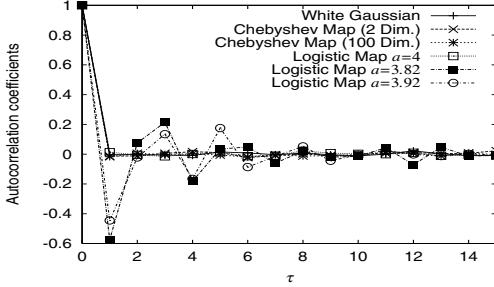


Fig. 2. Autocorrelation of the noise sequence applied to the optimization neural networks

all of neurons, that with different dimension for each neuron, and the logistic map chaos, $z_{ij}^c(t+1) = az_{ij}^c(t)(1 - z_{ij}^c(t))$, with $a = 3.82$, $a = 3.92$ and $a = 4$. The abscissa axis is an amplitude of the noise β in Eq. (II). The solvable performance on the ordinate is the percentage of successful runs obtaining the optimum solution in 1000 runs with different initial conditions. The successful run obtaining of the optimum solution means that the optimum solution state is found at least once in a fixed iteration. The cutoff iterations for each run are set to 1024 for TSP and to 4096 for QAP, respectively. The parameters of the neural networks are $A = 1$, $B = 1$, $\epsilon = 0.3$ for the TSP, and $A = 0.35$, $B = 0.2$, $\epsilon = 0.075$ for the QAP, respectively. The problems introduced in Figs. II (a) and (b) are a 20-city TSP in [8] and a QAP with 12 nodes, Nug12 in QAPLIB [14], respectively.

The results in Fig. II show that the neural networks with the logistic map with $a = 3.82$ and $a = 3.92$ perform better than other noise, for both problems. From Fig. 2 showing autocorrelation coefficients of each noise, the sequences having the better performance, the logistic map with $a = 3.82$ and $a = 3.92$, have negative autocorrelation. On the other hand, autocorrelation of the others are almost zero. From these results, it is clear that the negative autocorrelation is very important for improving the performance of the combinatorial optimization algorithm based on the Hopfield-Tank neural networks.

3 Realizing Ideal Spatiotemporal Searching Dynamics for High-Dimensional Solution Space

In order to realize ideal searching dynamics which has negative autocorrelation, this paper proposes a novel scheme to apply the Lebesgue spectrum filter to the output function of the optimization algorithms using the neural networks. In the followings, first such a filter is applied to the Hopfield-Tank neural network approach, and then it is also applied to the 2-opt heuristics for the large-scale TSPs.

3.1 Optimization Performance of Ideal Spatiotemporal Searching Dynamics for the Mutually Connected Neural Networks

In order to make the outputs of the neurons having lower cross-correlation and to realizing ideally complex searching dynamics, the following FIR filter, called Lebesgue spectrum filter, is introduced to make outputs having negative autocorrelation, which have been applied to the chaotic CDMA in Ref. [2],

$$\hat{f}(t) = \sum_{u=0}^M r^u f(t-u). \quad (7)$$

By setting $r = -(2 - \sqrt{3})$, ideal sequences to minimize the cross correlation have been generated for the chaotic CDMA. This paper introduces such a filter to the output function of each neurons, which can be described as follows, by replacing the output function f in Eq. (1) with the following equation,

$$f(y(t)) = 1/(1 + \exp(-\sum_{u=0}^M r^u y(t-u)/\epsilon)). \quad (8)$$

In the following experiments, $M = 8$ for each noise.

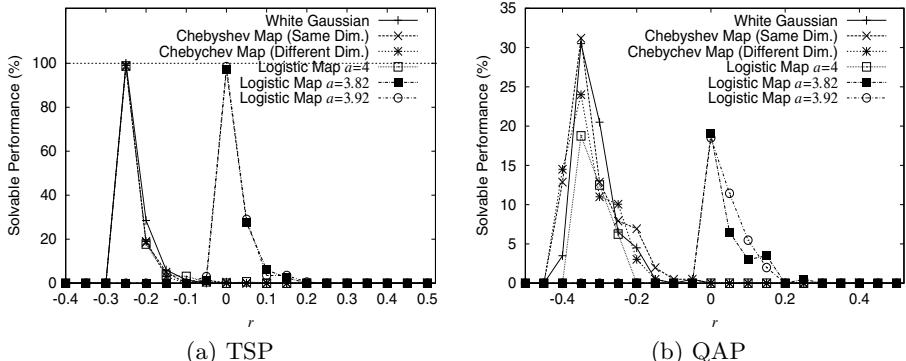


Fig. 3. Solvable performance of a neural network with the Lebesgue spectrum filter, on a 20-city TSP and a QAP with 12 nodes (Nug12)

The solvable performance of the neural networks with the novel output function generating the ideal spatiotemporal chaotic sequence is shown in Fig. 3. From these results, we can see that the performances of the neural networks with the noise, whose autocorrelation is almost zero, could be much improved for all of the cases. In Fig. 3 (b) for the QAP, the performance becomes even better than the logistic map chaos with $a = 3.82$ or $a = 3.92$, that have the best performance when the filter is not applied shown in Fig. 3 (b).

From Fig. 3 for the noise sequences which already have negative autocorrelation, the logistic map with $a = 3.82$ and $a = 3.92$, $r = 0$ is the best. On

the other hand, for other noise sequences having zero autocorrelation, the best value of autocorrelation parameter r for the Lebesgue spectrum filter is around $-(2 - \sqrt{3})$, which is the same value as that used to generate optimal sequences to minimize the cross correlation in the chaotic CDMA researches. From this result, such negative autocorrelation may ideally minimize the cross correlation in this asynchronously updated neural network. Minimization of the cross correlation among the neurons makes ideally complex search in the searching space and the performance is much improved.

3.2 Optimization Performance of Ideal Spatiotemporal Searching Dynamics for the 2-opt Heuristic Method

The Lebesgue spectrum filter generating ideal spatiotemporal searching dynamics is also applied to the heuristic algorithms. This paper introduces the 2-opt method for the TSP, which is very simple but possible to be applied to very large scale TSPs.

In order to apply the Lebesgue spectrum filter described above, the 2-opt method is implemented on a neural network in a similar way to Ref. [3]. The neuron in this neural network has two dimensional output function, $x_{ij}(t+1) = \Delta_{ij}(t)$, where $\Delta_{ij}(t)$ is the improvement of the total tour length by the 2-opt exchange connecting the city i and j . In the original 2-opt method, $x_{ij}(t+1)$ is asynchronously calculated, and the solution is really updated when $x_{ij}(t+1) > 0$. The Lebesgue spectrum filter generating negative autocorrelation is introduced to the 2-opt by using the following output function for each neuron,

$$x_{ij}(t+1) = \sum_{u=0}^M r^u x_{ij}(t-u). \quad (9)$$

When $x_{ij}(t+1) > 0$, the solution is really updated by the 2-opt exchange connecting the city i and j .

The results of the 2-opt with the Lebesgue spectrum filter on a 100-city, 200-city, 318-city, 442-city and 1173-city TSPs, KroA100, KroA200, Lin318, Pcb442, Pcb1173

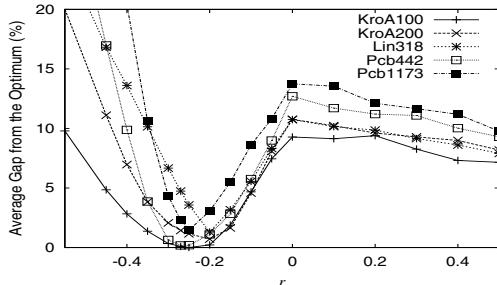


Fig. 4. Solvable performance of the 2-opt with the Lebesgue spectrum filter, on the large TSPs

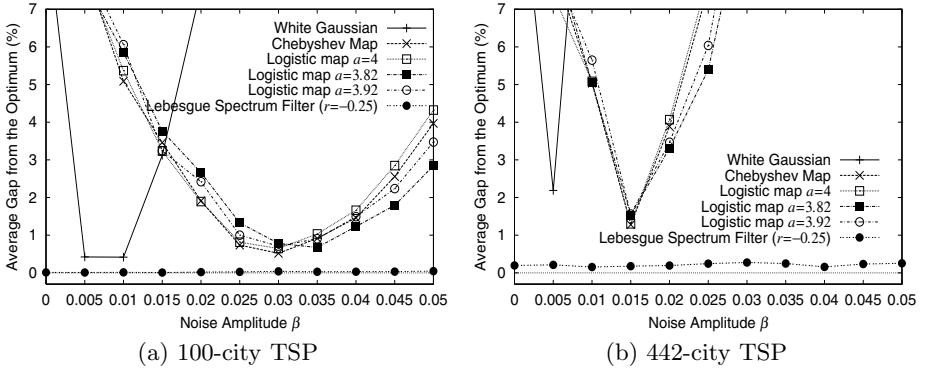


Fig. 5. Solvable performance of the 2-opt methods with additive stochastic noise, chaotic noise and with the Lebesgue spectrum filter which realizes ideal spatiotemporal searching dynamics, on a 100-city TSP, KroA100, and a 442-city TSP, Pcb442

Pcb1173 [14], are shown in Fig. 4, with changing the parameter r in the Lebesgue spectrum filter. The best performance is the case with negative r around $-(2 - \sqrt{3})$, which minimizes cross correlation among $s_{ij}(t)$. From these results, it is clarified that the Lebesgue spectrum filter, which realizes ideal spatiotemporal searching dynamics, is effective also for the heuristic methods.

In Fig. 5, the results of the 2-opt with the Lebesgue spectrum filter are compared with the 2-opt method with five kinds of additive noise sequences, which was introduced in the previous section, on a 100-city and a 442-city TSPs, KroA100 and Pcb442, respectively. From both results, the logistic map noises with negative autocorrelation, $a = 3.82$ and $a = 3.92$ do not provide better solutions than other noise. On the other hand, the proposed filtered search is effective also for the heuristic methods, and improves the performance of the algorithm much better than other algorithms.

4 Conclusion

This paper proposes an optimization approach using ideal chaotic searching dynamics realized by the Lebesgue spectrum filter on the output functions for the neural networks solving combinatorial optimization problems. The proposed scheme realizes ideal complex spatiotemporal searching dynamics, by minimizing the cross correlation among the decision functions using negative autocorrelation. The conventional researches on the chaotic CDMA [2] have shown that such filtered sequences having autocorrelation $C(\tau) \approx C \times r^\tau$ with r around $-(2 - \sqrt{3})$, has the lowest cross-correlation. In the proposed algorithm, such lower cross-correlation makes very complicated searching dynamics, and the performance of the algorithms with such filter can be improved.

The proposed algorithm is very simple and no need to find the appropriate parameter values carefully. Moreover, it is possible to be applied to various

algorithms. Therefore, it should be an interesting future work to apply this proposed approach to various problems and various algorithms.

Acknowledgement

This research is supported by the Japan Society for the Promotion of Science (JSPS) through its “Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program).”

References

1. Nozawa, H.: A neural network model as a globally coupled map and applications based on chaos. *Chaos* 2, 377–386 (1992)
2. Chen, L., Aihara, K.: Chaotic Simulated Annealing by a Neural Network Model with Transient Chaos. *Neural Networks* 8(6), 915–930 (1995)
3. Hasegawa, M., et al.: Combination of Chaotic Neurodynamics with the 2-opt Algorithm to Solve Traveling Salesman Problems. *Physical Review Letters* 79(12), 2344–2347 (1997)
4. Hasegawa, M., et al.: A Novel Chaotic Search for Quadratic Assignment Problems. *European Journal of Operational Research* 139, 543–556 (2002)
5. Hasegawa, M., et al.: Solving Large Scale Traveling Salesman Problems by Chaotic Neurodynamics. *Neural Networks* 15, 271–283 (2002)
6. Hasegawa, M., et al.: Solving Combinatorial Optimization Problems by Nonlinear Neural Dynamics. In: Proc. of IEEE ICNN, pp. 3140–3145 (1995)
7. Hayakawa, Y., et al.: Effects of the Chaotic Noise on the Performance of a Neural Network Model for Optimization Problems. *Physical Review E* 51(4), 2693–2696 (1995)
8. Hasegawa, M., et al.: An Analysis on Additive Effects of Nonlinear Dynamics for Combinatorial Optimization. *IEICE Trans. Fundamentals* E80-A (1), 206–212 (1997)
9. Hopfield, J., Tank, D.: Neural Computation of Decision in Optimization Problems. *Biological Cybernetics* 52, 141–152 (1985)
10. Aihara, K., et al.: Chaotic neural networks. *Phys. Lett. A* 144, 333–340 (1990); Aihara, K.: Chaotic neural networks. In: Kawakami, H. (ed.) *Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems*, pp. 143–161. World Scientific, Singapore (1990)
11. Mazzini, G., et al.: Interference minimization by autocorrelation shaping in asynchronous DS-CDMA systems: chaos-based spreading is nearly optimal. *Electronics Letters* 35(13), 1054–1055 (1999)
12. Umeno, K., Yamaguchi, A.: Construction of Optimal Chaotic Spreading Sequence Using Lebesgue Spectrum Filter. *IEICE Trans. Fundamentals* E85-A (4), 849–852 (2002)
13. Hasegawa, M., Umeno, K.: Solvable Performances of Optimization Neural Networks with Chaotic Noise and Stochastic Noise with Negative Autocorrelation. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) *ICONIP 2007, Part I. LNCS*, vol. 4984, pp. 693–702. Springer, Heidelberg (2008)
14. Burkard, R., et al.: QAPLIB-A Quadratic Assignment Problem Library. *Journal of Global Optimization* 10, 391–403 (1997), <http://www.seas.upenn.edu/qplib>

A Multiple Sound Source Recognition System Using Pulsed Neuron Model with Short Term Synaptic Depression

Kaname Iwasa, Mauricio Kugler,
Susumu Kuroyanagi, and Akira Iwata

Department of Scientific and Engineering Simulation,
Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan
`kaname@nitech.ac.jp, mauricio@kugler.com, {bw,iwata}@nitech.ac.jp`

Abstract. Many applications would emerge from the development of artificial systems able to accurately localize and identify sound sources. However, one of the main difficulties of such kind of system is the natural presence of mixed sound sources in real environments. This paper proposes a pulsed neural network based system for extraction and recognition of objective sound sources from background sound source. The system uses the short term depression, that implements by the weight's decay in the output layer and changing the weight by frequency component in the competitive learning network. Experimental results show that objective sounds could be successfully extracted and recognized.

1 Introduction

By the information provided from the hearing system, the human being can identify any kind of sound(sound recognition) and where it comes from(sound localization)[\[1\]](#). If this ability could be reproduced by artificial devices, many applications would emerge, from support devices for people with hearing loss to safety devices. With the aim of developing such kind of device, a sound localization and recognition system using Pulsed Neuron (PN) model[\[2\]](#) have been proposed in [\[3\]](#). PN models deal with input signals on the form of pulse trains, using an internal membrane potential as a reference for generating pulses on its output. PN models can directly deal with temporal data, avoiding unnatural windowing processes, However, that system cannot recognize the multiple sound sources, as the competitive learning network in the system choose only the most suitable neuron using winner-takes-all rule. For example, when two types of sound are mixed, with a objective sound and another background sound, the objective sound is mistaken with the background sound or other sound. In many researches, Independent Components analysis(ICA) is used to separate and recognize the multiple sound sources[\[4\]](#). ICA has a high performance by mathematical grounds but its calculation is complex. PN model is simple structure and can be more easily implemented in hardware[\[5\]](#).

In the human brain, by construct the objective sound can be extracted and recognized by the high-level processing like the cocktail party effect. Selective

attention for short term depression [6][7][8] is used in this work in order to solve this problem. The short term depression is a synaptic operation, in which a potential is decreased after being increased to a peak. This paper proposes a PN based system for extraction and recognition of mixed sound sources using the short term depression, with the purpose of extracting the objective sound source from the multiple sound and recognize the objective sound.

2 Proposed System

The proposed system consists of three main blocks, the filtering and frequency-pulse converter, the competitive learning network and the output layer. The competitive learning network and the output layer blocks are based on PN.

2.1 Filtering and Frequency-Pulse Converter

In order to enable pulsed neuron based modules to process the sound data, the analog input signal must be divided on its frequency components and converted to pulses. A bank of band-pass filters decomposes the signal, and each frequency channel is independently converted to a pulse train, which rate is proportional to the amplitude of the correspondent signal. The filters' center frequencies were determined in order to divide the input range (100 Hz to 7 kHz) in 43 channels equally spaced in a logarithm scale.

2.2 Pulsed Neuron Model

When processing temporal data (e.g. sound), it is important to consider the time relations and to have computationally inexpensive calculation procedures to enable real-time processing, requirements fulfilled by the PN model.

Figure 1(a) shows the structure of the PN model. When an input pulse $IN_n(t)$ reaches the n^{th} synapse, the local membrane potential $p_n(t)$ is increased by the value of the weight w_n . The local membrane potentials decay exponentially with a time constant τ_n across time. The neuron's output $o(t)$ is given by:

$$o(t) = H(I(t) - \theta) \quad (1)$$

$$I(t) = \sum_{n=1}^N p_n(t) \quad (2)$$

$$p_n(t) = w_n IN_n(t) + p_n(t-1) \exp\left(-\frac{t}{\tau}\right) \quad (3)$$

where N is the total number of inputs, $I(t)$ is the inner potential, θ is the threshold and $H(\cdot)$ is the unit step function. The PN model also has a refractory period t_{ndti} , during which the neuron is unable to fire, independently of the membrane potential.

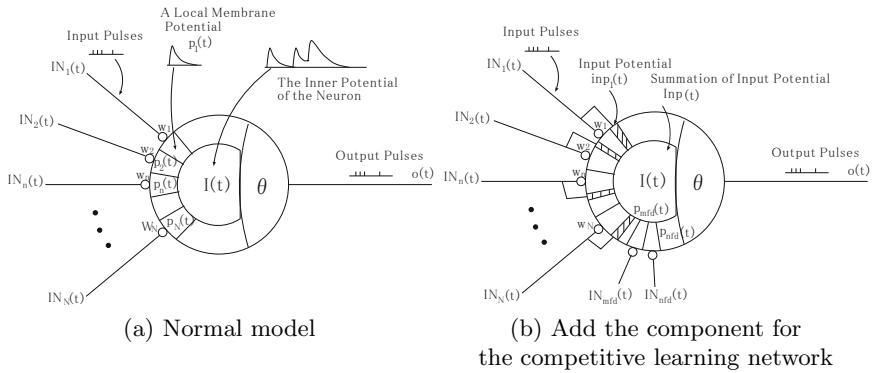


Fig. 1. Pulsed neuron model

2.3 Competitive Learning Network for PN Model

The Competitive Learning Network using Pulsed Neurons (CONP) proposed in [9]. This network recognize the kind of sound source from the input pulses based frequency of sound.

The CONP is constructed with PN models. In the learning process of CONP, the neuron with the most similar weights to the input (winner neuron) should be chosen for learning in order to obtain a topological relation between inputs and outputs. However, in the case of two or more neurons firing, it is difficult to decide which one is the winner, as their outputs are only pulses, and not real values. In order to this, CONP has extra external units called control neurons. Based on the output of the Competitive Learning (CL) neurons, the control neurons' outputs increase or decrease the inner potential of all CL neurons, keeping the number of firing neurons equal to one. Controlling the inner potential is equivalent to controlling the threshold. Two types of control neurons are used in this work. The No-Firing Detection (NFD) neuron fires when no CL neuron fires, increasing their inner potential. Complementarily, the Multi-Firing Detection (MFD) neuron fires when two or more CL neurons fire at the same time, decreasing their inner potential [9].

Considering all the described adjustments on the inner potential of CONP neurons, the output equation (II) of each CL neurons becomes:

$$o(t) = H \left(\sum_{n=1}^N p_n(t) - \theta + p_{nfd}(t) - p_{mfd}(t) - \beta \cdot Inp(t) \right) \quad (4)$$

where $p_{nfd}(t)$ and $p_{mfd}(t)$ corresponds respectively to the potential generated by NFD and MFD neurons' outputs, $Inp(t)$ is the summation input potential and β ($0 \leq \beta \leq 1$) is a parameter.

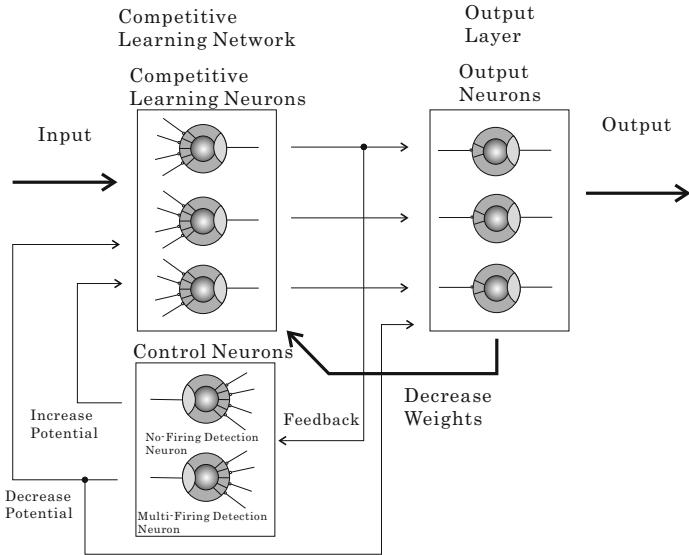


Fig. 2. Competitive Learning Network and Output Layer

2.4 Output Layer

In this paper, we propose to add the new layer of PN model to extract the objective sound. Figure 2 shows the pulsed neural networks parts in proposed system. Each neuron in the output layer receive the pulses from its correspondent CONP neuron independently. This layer works by reducing the weights of its neuron corresponding to the dominant sound, similar to the short term depression. To realize this work, these neurons' weight $w_j^d(t)$ changes by the rate of input pulses as equation (5)(6):

$$w_j^d(t) = \eta \cdot w_j^d(0) \quad (w_j^d \geq 0) \quad (5)$$

$$\eta = \begin{cases} \frac{Inp_{max}^d - \varepsilon \cdot Inp_j^d(t)}{Inp_{max}^d} & (p_{mfd}^d(t) > \theta_{mfd}^d \quad and \quad I_k^d(t) > \theta^d) \\ \frac{Inp_{max}^d + \varepsilon \cdot Inp_j^d(t)}{Inp_{max}^d} & (p_{mfd}^d(t) < \theta_{mfd}^d \quad and \quad I_k^d(t) < \theta^d) \end{cases} \quad (6)$$

where $Inp_j^d(t)$ is a summation of input potentials and Inp_{max}^d is a maximal value of summation of input potential, $p_{mfd}^d(t)$ is a potential from the MFD and θ_{mfd}^d is a threshold for the $p_{mfd}^d(t)$. If the input pulses based the background sound is inputted, the neuron corresponding to that sound in CONP continues to fire and the input potential of the respective neuron in the output layer increases. Then, weight of the neuron in output layer is decreased by this equation and the neuron stops to fire. Next, the maximal potential $\max_j\{inp_j(t)\}$ of neuron in output layer is found and the weights of each neurons in CONP w_{nj}^m are decreased by Eq.(7):

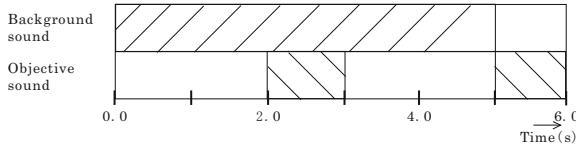


Fig. 3. Input patterns on experiments

Table 1. Parameters of recognition experiment

Number of CL neurons	J	5[units]
Time Constant	τ^d	0.02(s)
Input Time Constant	τ_{in}^d	2.5(s)
Threshold for potential	θ^d	10
Threshold for MFD	θ_{mfd}^d	2
Rate of Weight Decay	ε	2.0
Rate of Weight Decreasing	γ	0.0008

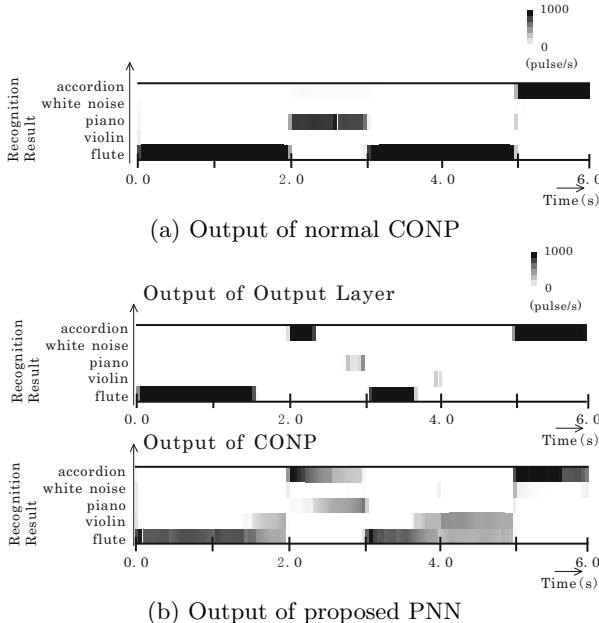


Fig. 4. Output pulses on experiment

$$w_{nj}^c(t) = w_{nj}^c(0) - \gamma \cdot \max_j\{inp_j(t)\} \cdot w_{nmax}^c(0) \quad (w_{nj}^c(t) \geq 0) \quad (7)$$

where $w_{nj}^c(0)$ is weight of j th CL neuron after the learning, $w_{nmax}^c(0)$ is weight of output layer neuron which has the maximal potential, and γ is a parameter of the decreasing rate for CL neurons' weight. Using this equation, the neuron

corresponding to the background sound in CONP stop to fire and it enables the other neuron to fire. The neuron corresponding to the objective sound can fire in many cases. When the decreasing the weights in the CONP, many neurons will fire. Hence, the potential and threshold from the MFD is used in the Eq.6.

3 Experimental Results

In this paper, several artificial sound signals were used: four musical instruments' sounds ("Accordion", "Flute", "Piano" and "Violin") and "White Noise". Initially, the firing information of each type of sound source is extracted. This data is used for training CONP, according to the same parameters shown in Fig.5.

Fig.3 shows the input patterns in experiments. That x-axis is time, and y-axis is the sound pattern. The gray-area shows the sound is inputted. The background sound lasts until 5.0s, and the other sound is mixed in 2.0s~3.0s period. The aim of this experiments is that the system can recognize the objective sound in 2.0s~3.0s period.

Fig.4 shows the output result in the normal CONP (a) and proposed system (b) when the "Accordion" is the objective sound with the "Flute" as the background sound. The x-axis is time, and y-axis is the neuron index and the

Table 2. Recognition Rate of Sound Recognition[%] (A = Accordion, F = Flute, P = Piano, V = Violin, W = White Noise, - = No Firing)

Objective Sound	Background Sound	Time					
		~ 1.0	~ 2.0	~ 3.0	~ 4.0	~ 5.0	~ 6.0
A	F	F(100.0)	F(99.8)	A(89.6)	F(95.7)	W(100.0)	A(100.0)
	P	P(100.0)	P(98.4)	A(100.0)	P(94.1)	F(75.0)	A(100.0)
	V	V(100.0)	V(100.0)	A(100.0)	A(72.4)	-	A(100.0)
	W	W(100.0)	W(100.0)	A(99.6)	W(100.0)	-	A(99.9)
F	A	A(100.0)	A(99.4)	F(97.3)	A(97.9)	-	F(98.5)
	P	P(100.0)	P(98.4)	F(99.8)	P(99.1)	F(80.0)	F(98.6)
	V	V(100.0)	V(100.0)	F(100.0)	V(100.0)	P(100.0)	F(100.0)
	W	W(100.0)	W(100.0)	F(100.0)	W(96.7)	-	F(100.0)
P	A	A(100.0)	A(99.4)	F(99.4)	A(50.0)	-	P(100.0)
	F	F(100.0)	F(99.8)	P(100.0)	F(99.6)	W(100.0)	P(83.6)
	V	V(100.0)	V(100.0)	P(99.5)	-	-	P(94.4)
	W	W(100.0)	W(100.0)	P(93.4)	W(100.0)	-	P(86.1)
V	A	A(100.0)	A(99.4)	F(100.0)	A(91.7)	-	V(100.0)
	F	F(100.0)	F(99.8)	V(52.8)	F(99.6)	-	V(98.2)
	P	P(100.0)	P(98.4)	V(78.2)	P(100.0)	F(87.5)	V(100.0)
	W	W(100.0)	W(100.0)	V(99.8)	W(100.0)	P(100.0)	V(75.9)
W	A	A(100.0)	A(99.4)	W(100.0)	A(99.5)	-	W(100.0)
	F	F(100.0)	F(99.8)	W(100.0)	F(99.6)	-	F(100.0)
	P	P(100.0)	P(98.4)	W(100.0)	P(93.2)	F(66.7)	W(100.0)
	V	V(100.0)	V(100.0)	W(100.0)	V(100.0)	P(100.0)	W(100.0)

recognition result. The gray level represents the rate of the pulse train, with white corresponding to no pulse and black corresponding to many pulses. Table 1 shows the parameter used in this experiment.

In this case, the system should output a result of the objective sound “Accordion” in 2.0~3.0s period. The result of normal CONP presents a false recognition result. Because of this, the frequency component of the mixed sound is nearly for other sound(in this case “Piano”). Some of output of the proposed method is short time. As an effect of short term depression in the output layer, the neuron cannot continue to fire. The proposed method can recognize the objective sound “Accordion” correctly by the effect of CONP’s weight decreasing. The “Accordion” neurons’ weight is decreased and the “Piano” neuron presents some firing between 2.0s~3.0s period. These outputs are lower than the threshold θ^d , there is not influence to output layer.

Table 2 shows the recognition result for all combinations of sounds. The recognition rate is defined as the ratio between the number of neuron’s firing corresponding to the sound and the total number of firings. The correct sound could be recognized in most of the cases. Table 3 shows the firing time rate in each result. The firing time rate is defined as ratio between the number of neuron’s firing corresponding to the sound and the total length of unit time.

Table 3. Firing time rate of sound recognition[%] (A = Accordion, F = Flute, P = Piano, V = Violin, W = White Noise, - = No Firing)

Objective Sound	Background Sound	Time					
		~ 1.0	~ 2.0	~ 3.0	~ 4.0	~ 5.0	~ 6.0
A	F	F(97.7)	F(59.0)	A(35.2)	F(60.6)	W(2.4)	A(97.4)
	P	P(97.3)	P(31.6)	A(59.5)	P(9.5)	F(0.3)	A(93.8)
	V	V(90.9)	V(10.9)	A(65.9)	A(2.1)	-	A(94.7)
	W	W(97.7)	W(54.3)	A(80.8)	W(39.4)	-	A(68.7)
F	A	A(97.4)	A(47.4)	F(65.0)	A(38.1)	-	F(6.7)
	P	P(97.3)	P(31.6)	F(77.3)	P(11.5)	F(0.4)	F(13.9)
	V	V(90.9)	V(10.9)	F(76.9)	V(10.3)	P(0.1)	F(77.8)
	W	W(97.7)	W(54.3)	F(92.8)	W(46.9)	-	F(91.2)
P	A	A(97.4)	A(47.7)	F(17.3)	A(0.2)	-	P(81.2)
	F	F(97.7)	F(59.0)	P(27.4)	F(24.4)	W(0.2)	P(54.9)
	V	V(90.9)	V(10.9)	P(21.5)	-	-	P(1.7)
	W	W(97.7)	W(54.3)	P(36.9)	W(28.2)	-	P(3.1)
V	A	A(97.4)	A(47.7)	F(6.7)	A(1.1)	-	V(24.4)
	F	F(97.7)	F(59.0)	V(11.5)	F(27.9)	-	V(11.2)
	P	P(97.3)	P(31.6)	V(6.1)	P(0.3)	F(0.7)	V(12.3)
	W	W(97.7)	W(54.3)	V(55.9)	W(38.1)	P(0.1)	V(2.2)
W	A	A(97.4)	A(47.7)	W(74.3)	A(40.3)	-	W(97.2)
	F	F(97.7)	F(59.0)	W(46.0)	F(22.4)	-	F(12.2)
	P	P(97.3)	P(31.6)	W(90.2)	P(27.5)	F(0.2)	W(89.7)
	V	V(90.9)	V(10.9)	W(65.8)	V(3.2)	P(0.1)	W(90.0)

The recognition rates in the 4.0s~5.0s period are high but last for short periods. Hence, these wrong results can be ignored.

Some extraction results(e.g. when the objective sound is “Piano” and the background sound is “Accordion”) are wrong, but it could recognize using other parameter. Also almost no neuron fired in 5.0s because of the weight decay in output layer. Thus, these neurons stop to fire for background sound.

4 Conclusions

This paper proposes a system for multiple sound source recognition based on a PN model. The system is composed of a CONP model which recognizes the sound source type and the output layer which has a short term depression. The short term depression is implemented with the decreasing of the CONP weights and decaying the weight on the output layer.

The experimental results confirm that the output layer can successfully decrease the weights of CONP and the proposed system can extract the objective sound source from the mixed sound.

Future works include changing these parameter automatically, the application of the proposed system to real sound signals and also the use of the information of the sound type for locating its source with high precision.

References

1. Pickles, J.O.: An Introduction to the Physiology of Hearing. Academic Press, London (1988)
2. Maass, W., Bishop, C.M.: Pulsed Neural Networks. MIT Press, Cambridge (1998)
3. Kuroyanagi, S., Iwata, A.: Perception of Sound Direction by Auditory Neural Network Model using Pulse Transmission – Extraction of Inter-aural Time and Level Difference. In: Proceedings of IJCNN 1993, pp. 77–80 (1993)
4. Jutten, C., Herault, J.: Blind separation of sources, part I: An adaptive algorithm based on the neuromimetic architecture. Signal Process 24, 1–10 (1991)
5. Iwasa, K., Kuroyanagi, S., Iwata, A.: A Sound Localization and Recognition System using Pulsed Neural Networks on FPGA. In: Proceeding of International Joint Conference of Neural Networks 2007 (August 2007) (to appear)
6. Abbott, L.F., Valera, J.A., Sen, K., Nelson, S.B.: Synaptic depression and cortical gain control. Science 412, 513–541 (1989)
7. Abbott, L.F., Gerstner, W.: Homeostasis And Learning Through Spike-Timing Dependent Plasticity. Methods and models in neurophysics (2004)
8. Ralf, S., Sakaba, T., Erwin, N.: Vesicle pools and short-term synaptic depression: lessons from a large synapse. Trends Neurosci 25, 206–212 (2002)
9. Kuroyanagi, S., Iwata, A.: A Competitive Learning Pulsed Neural Network for Temporal Signals. In: Proceedings of ICONIP 2002, pp. 348–352 (2002)

A Model of Path Integration and Navigation Based on Head Direction Cells in Entorhinal Cortex

Tanvir Islam¹ and Ryutaro Fukuzaki²

¹ Laboratory for Dynamics of Emergent Intelligence, RIKEN BSI,

2-1 Hirosawa, Wako-shi, Saitama, Japan

tanvir@brain.riken.jp

² Department of Applied Physics and Chemistry,

The University of Electro-Communications, Tokyo, Japan

r-fukuzaki@brain.riken.jp

Abstract. So far it is widely believed that mammalian navigation is based on map-like place representations created by hippocampal place cells. Models supporting this compute returning path to origin as a sequence of places, or as a sequence of firing place cells. However, these models often fail to compute the shortest returning path to start point. Moreover, one big constraint of these models is that the space has to be well explored to find a returning path from a certain endpoint. Here we propose a computational model for path integration and navigation based on Head Direction (HD) cells and grid cells. We show that an ensemble of HD cells can accumulate distance and compute the direction from the start point to the present location during exploration. Based on this vector navigation, and with the help of properties of grid cells in entorhinal cortex, it is possible to navigate the way back to the origin. We show that our recent model of rat entorhinal cortex provides these functional mechanisms of path integration and vector navigation.

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, are believed to construct a path that represents a continuous series of locations during navigation. Almost all current models of rodent navigation support that hippocampal place cells provide a place code [1] and the lesions in the hippocampus may cause damage to navigation [9]. However, while it is showed that place cells firing can identify the current location [10], it remains a problem to efficiently compute the return path to the start point based on place cell firing. A common theme of all these navigation models is that one way or another, before initiating movement or during travel, a sequence of place cells must be activated to navigate between two locations, therefore, requiring extensive exploration of the environment prior to navigation, because a navigation route driven by place cells cannot cross a previously unexplored area. On the other hand, in case of mammals like rats, even in darkness, their ability to come back to their nest on a straight line from some location

is found (Mittlestaedt et al. 1980). This ability of the rats suggests that some kind of vector navigation is performed similar to that found in insects (Collett et al., 1986, 1998; Wehner, 2003), which computes efficient routes between locations in even unexplored space. Head-direction cells found mostly in the rat's entorhinal cortex (EC) have the remarkable property of firing at maximal rates when the animal is heading to a particular azimuthal angle. These cells with various angles of preference are connected to grid cells in EC layer II and III, which give major input to hippocampal CA3 and CA1 place cells. Some earlier works [6] [7] suggest that the periodic firing fields of grid cells are metrics of space computation. In our previous work [8] we proposed a computational model of periodic grid fields based on HD cells in EC deeper layers. In this paper, we elucidate with our computational model that an ensemble of HD cells can compute the distance and direction of the current location from the start point, thus integrating the path in an accumulative way. We further show that the characteristic of grid cells helps the vector navigation system in finding efficient paths back to the origin. We show with computer simulation that based on this sort of vector navigation, it is possible to find optimal routes in uncharted territory.

2 Path Integration and Navigation

Path integration is the ability of using self-motion cues to compute a vector between a navigator's current location and an origin (Mittlestaedt, 2000). To perform this operation, there has to be (1) an integrator that sums movements in distance units, (2) a directional representation that keeps track of the direction of the animal's movements with reference to an inertial frame and (3) a mechanism to find out the correct direction from the current position to the origin with the help of the properties of the vector navigation system.

We consider that path integration and navigation is accomplished by an ensemble of HD cells in EC deeper layers. In our model, these HD cells are connected downstream to grid cells in the EC layer II and III. HD cells in the deeper layer are assumed to be modules with hexagonal direction system, which consists of a subset of six HD cells (60° apart) present in deeper layers (Fig. 1B). While the animal is moving freely, HD cells in the hexagonal direction system receive running velocity and head direction information through sensory system. Each component or HD cell in the hexagonal direction system then computes the amount of displacement along its preferred direction vector. These sets of six HD cells are connected to three (or more) grid units in the superficial layer whose firing field can altogether cover the whole 2D space [8]. These three units have equidistant appropriate spatial phase, common grid spacing and common angular orientation. Each hexagonal direction system connected to three grid cells form a column-like structure and many of these columns are expanded from the deeper layers to the superficial layers of EC. Also, we consider the gradual increase of grid spacing and variety in angular orientation in columns distributed from dorsal to ventral region of the medial entorhinal cortex, as found in previous experiment data.

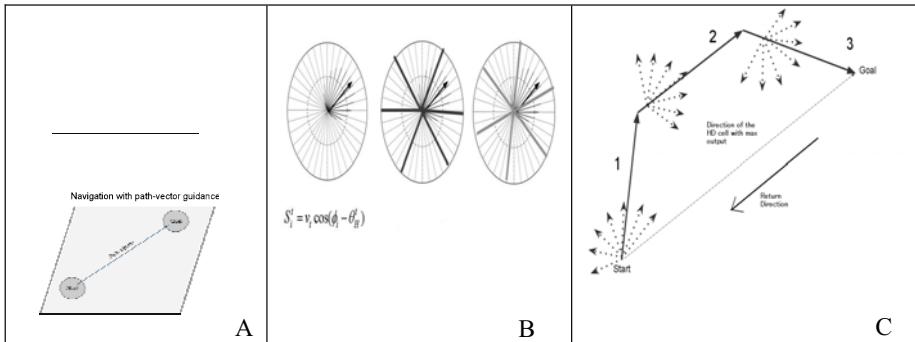


Fig. 1. A: Concept of navigation by place cells (firing sequence) and HD cells (vector-based). B: example of an ensemble of HD cells with the black arrow showing the present head direction of the animal. Cells with nearby preferred angle have high outputs (Showed by the thickness of lines). Middle and right: Two subsets of six HD cells that are connected to two sets of grid cells. C: The returning path to start point is the direction opposite to the preferred angle of the HD cell with maximum output value at the endpoint (dotted line connecting the start and endpoint), thus computing vector navigation

After receiving the present velocity and head direction signal, the six components in the hexagonal system compute the amount of displacement of the animal along their preferred direction angles, as depicted in eq. (1) below.

$$S'_i = [v_t \cos(\phi_i - \theta'_H) + p_i^{t-1}] \quad (1)$$

Where,

$$p_i^{t-1} = [v_{t-1} \cos(\phi_i - \theta_H^{t-1}) + p_i^{t-2}] \quad (2)$$

Here, the velocity of the animal is v_t (unit: cm/s), and the head direction angle, with the reference of horizontal axis, is θ'_H (unit: degree). We denote these preferred angles as ϕ_i , where $i = 1, 2, \dots, 6$ is the notation for six angles of the HD subset. The preferred direction angle of the six HD cells varies 60 degrees from the members of the same subset (Fig. 1). For example, one subset may have HD cells with preferred directions of 0, 60, 120, 180, 240 and 300 degrees, while another subset may have preferred directions of 10, 70, 130, 190, 250, 310 degrees. The first term in the right hand side of eq. (1) corresponds to the amount of displacement along the preferred direction vector, and the second term considers the effect of spatial phase. The third term clarified in eq. (2) provides the functionality of path integration by the direction-system modules based on the accumulation of displacement along each direction of preferred angle.

3 Simulation Results

Based on our model of EC, we created a network of 360 HD cells with preferred angle of 1~360 degrees, with the population having 1 degree resolution. A set of 6 HD cells (60 degrees apart in preferred angle as described below) are connected to three

grid cells in EC III and II. These triplets have same grid spacing and angular orientation, but appropriate spatial phase to cover the space in a non-overlapping way. There are 60 triplets of grid cells with various angular orientations between 0 to 60 degrees, and spacing ranging from 20-80cm. A $2m \times 2m$ area is chosen for initial random movement at 10cm/s speed. During movement, 360 HD cells accumulate S_i^t as distance along their preferred direction. Also the sum of outputs of HD cells (notation $\text{sum}(|S_i^t|)$) is calculated at each point, which corresponds to the total distance from the origin.

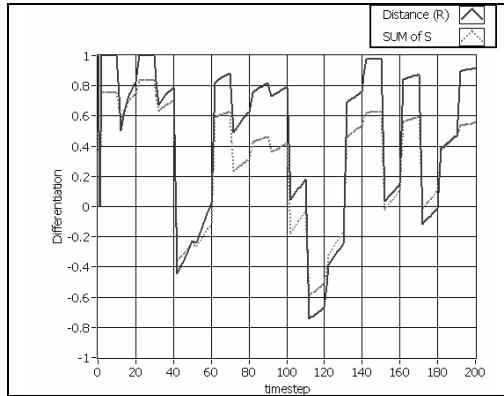


Fig. 2. Two graphs showing the differentiation of distance from origin and $\text{sum}(|S_i^t|)$ are similar. This suggests that while returning home, the animal should take a path where $\text{sum}(|S_i^t|)$ decreases all the time.

As described before, at the endpoint, the direction from the origin is computationally same as the preferred angle of the HD cell with the maximum value of S_i^t (notation of $\text{Max}(S_i^t)$ is used from here). The most efficient return path to the origin is therefore the exact opposite angle of this. If the animal knows the preferred angle of the HD cell with maximum S_i^t directly from the head direction system, then it can simply choose the directly opposite direction to proceed until $\text{Max}(S_i^t)$ gets almost zero or $\text{sum}(|S_i^t|)$ is the same. However, what if the precise information about the preferred angle of individual HD cells cannot be obtained directly by the animal brain? An HD cell will fire maximally when the animal heads to its preferred direction, but the firing intensity may not directly provide the value of the angle. If so, how does the animal find the correct direction to origin? We can assume that in this case the animal can just start heading a random direction from the end point and change the direction of movement a certain amount if the current values either the $\text{Max}(S_i^t)$ or $\text{sum}(|S_i^t|)$ starts increasing compared to the previous time-step. Fig. 2 shows nine examples of initial movement (gray trajectory) and return path (dark black trajectory), where the amount of direction-change at a time is 60 degrees. It can be observed in

fig. 2 that the speed of movement needs to be reduced near the origin so that $\text{Max}(S'_i)$ converges to the small threshold.

Experiments often show that rodents like rats come back home on a straight-line, after wondering around the environment even in darkness. To find the most efficient straight path home from the current location, the preferred angle of the HD cell with maximum S'_i has to be calculated. The animal can move in small amounts to several directions (test movement) and can choose the direction moving along which causes the most decrease in both $\text{sum}(|S'_i|)$ and $\text{Max}(S'_i)$. For example, the test movements can be done towards the six directions corresponding to the preferred angles of the grid cell to which the HD cell with $\text{Max}(S'_i)$ is connected. Fig. 4 shows that in such way, straight returning paths can be obtained through test movements in six directions (not showed in the graphs).

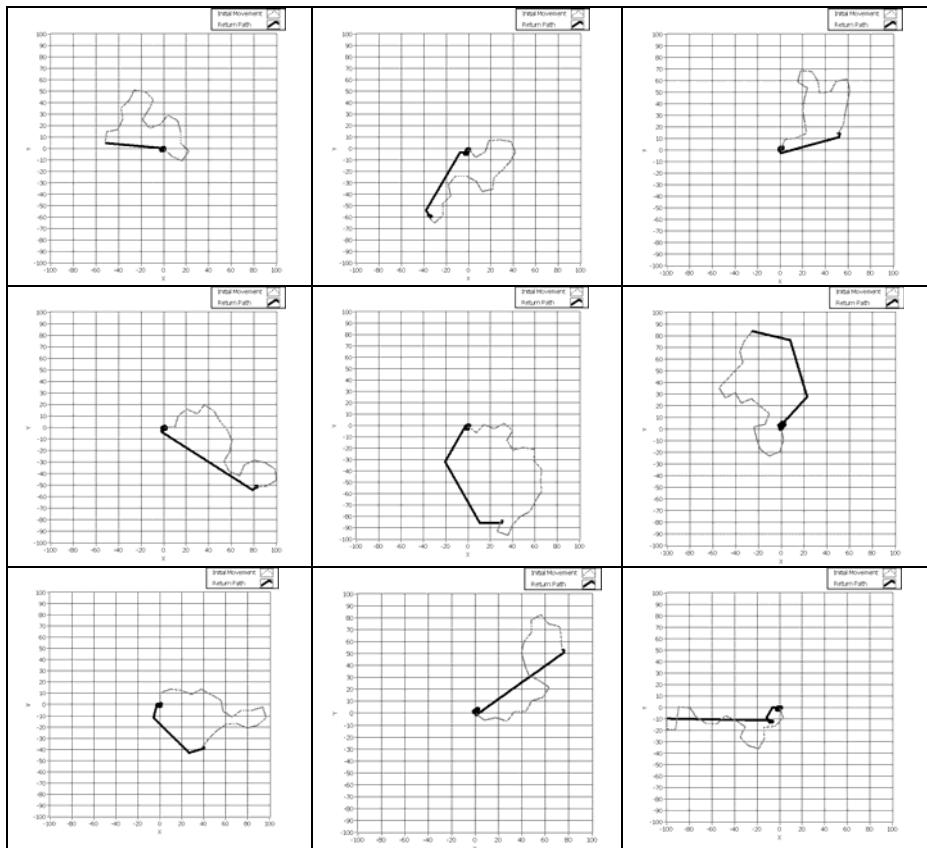


Fig. 3. Nine examples of paths of initial movement (in gray) from the origin ($X, Y = (0, 0)$) and the calculated return path (in black) with random searching

However, it should be noted that so far we have not considered the current head direction of the animal while computing the returning direction. If the animal performs test navigation along the six directions of the grid cell, without considering current head direction, the computed returning path will be a straight-line like fig. 4. However, if current head direction is added to the six preferred angles of the grid cell, then returning paths like fig. 5 are computed. The necessary conditions for direction changes in returning path are same as before: S'_i and $\text{sum}(|S'_i|)$ both have to decrease along the path. We have not found significant differences among returning paths of fig. 3 and 5. Fig. 6 shows the graphs of $\text{Max}(S'_i)$ and $\text{sum}(|S'_i|)$, showing how these values decrease along the returning path.

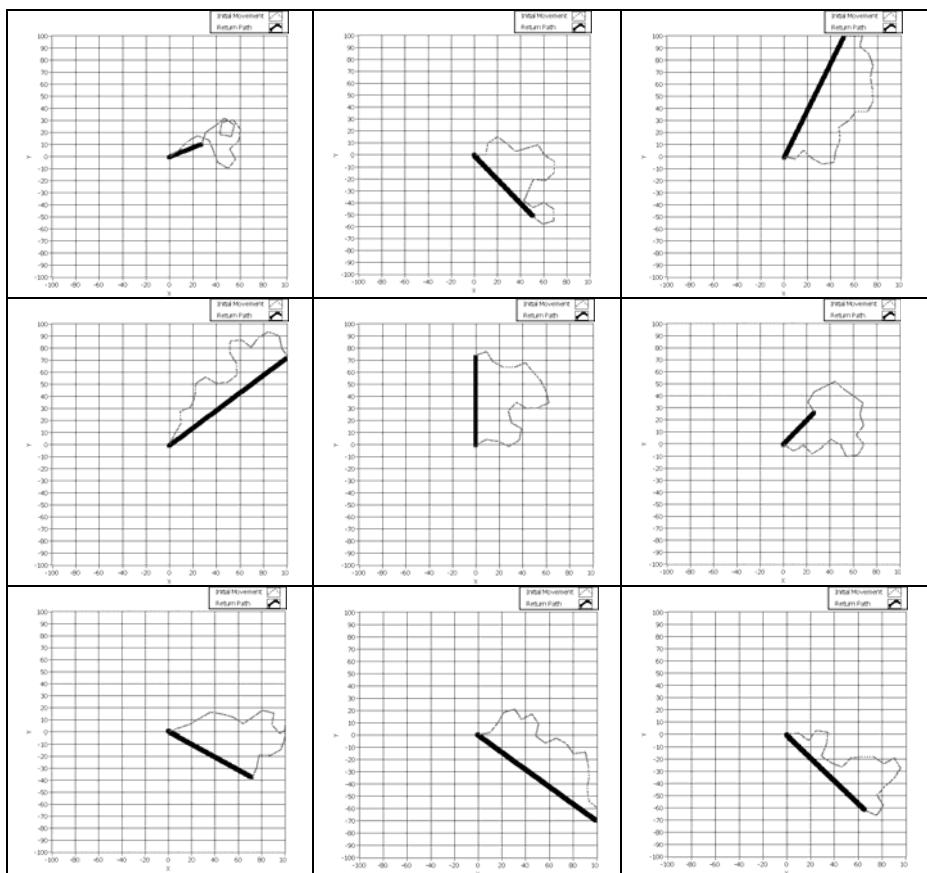


Fig. 4. Nine examples of paths of initial movement (in gray) from the origin $(X, Y) = (0, 0)$ and the calculated return path (in black) with searching by grid angles at the endpoint. The trajectories of direction searching are not showed. Return paths calculated in this way are straight-lines.

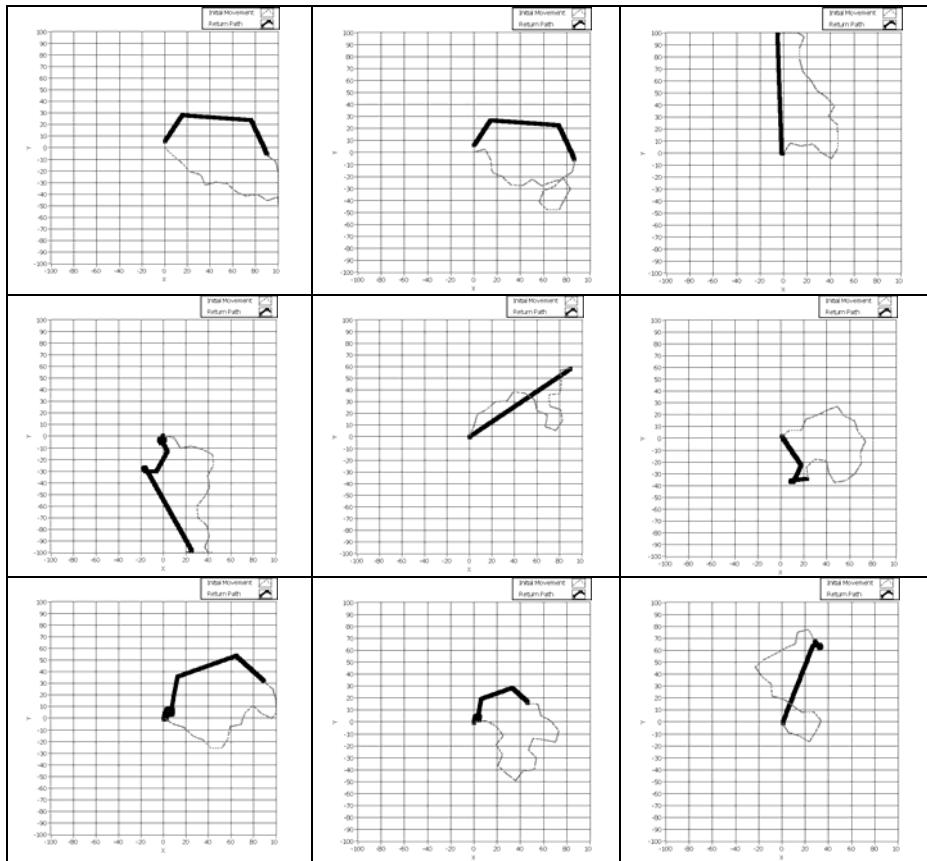


Fig. 5. Nine examples of paths of initial movement (in gray) from the origin $(X, Y) = (0, 0)$ and the calculated return path (in black) with searching by grid angles, while considering the head direction at the endpoint. The trajectories of direction searching are not showed. Return paths calculated in this way are not straight-lines and resemble return paths in fig. 3.

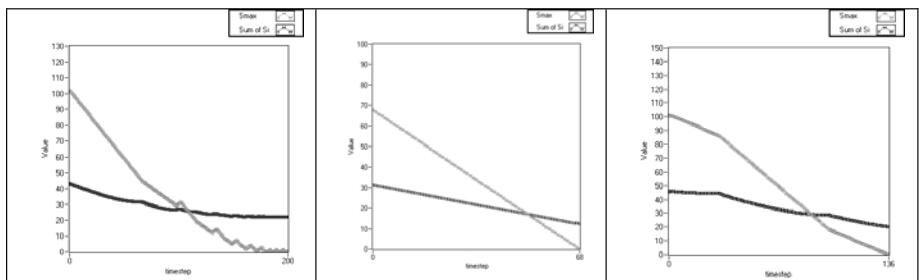


Fig. 6. Graphs showing the gradual decrease in $\text{sum}(|S'_i|)$ (in black) and $\text{Max}(S'_i)$ (in gray) for random direction search (left), search along the grid angles (middle) and search with considering the head direction at endpoint (right)

4 Discussion and Conclusion

In this paper we have showed our model of EC consisting of HD cells and grid cells. We described that the functional mechanism of path integration and navigation is possible through the vector accumulation of an ensemble of HD cells. The return path can be calculated by searching for the direction that satisfies gradual decrease of both $\text{Max}(S_i^t)$ and $\text{sum}(|S_i^t|)$. The direction can be found either random search or exploring from a subset of directions like the example of grid cell angles we showed. In this paper we hypothesized that the direction opposite to the preferred angle of $\text{Max}(S_i^t)$ is not known directly by the animal, therefore, the animal has to find a direction with which the two constraints satisfy. Again, we considered head direction angle at the current location, which addresses the problem of finding the direction to origin relative to the present head direction. Earlier works supporting vector navigation [11][12] do not consider these questions while providing a solid mathematical base of the vector navigation.

In our navigation model, we used $\text{Max}(S_i^t)$ and $\text{sum}(|S_i^t|)$ as two monitoring factors. When the animal moves a long way from the origin, the values of individual S_i^t and $\text{Max}(S_i^t)$ become too big to properly encode in firing rate of HD cells. In our earlier model of grid cells [8], a modulus of grid spacing over S_i^t is considered to reset the value of S_i^t at certain locations. If we consider this resetting, the value of $\text{Max}(S_i^t)$ will not provide the correct direction because of the modulus. In this case, the $\text{sum}(|S_i^t|)$ can still be used as a value the animal may monitor while heading home. However, future experiments on mammal navigation system may provide more knowledge about the exact mechanism of navigation in mammals.

Acknowledgement. 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.: 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.: 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.: Path integration and the neural basis of the “cognitive map”. *Nature Rev. Neuroscience* 7, 663–678 (2006)

8. 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)
9. Morris, R.G., Garrud, P., Rawlins, J.N., O'Keefe, J.: Place navigation impaired in rats with hippocampal lesions. *Nature* 297, 681–683 (1982)
10. Wilson, M.A., McNaughton, B.: Dynamics of the hippocampal ensemble code for space. *Science* 261, 1055–1058 (1993)
11. Kubie, J.L., Fenton, A.A.: Heading vector Navigation based on head direction cells and path integration. *Hippocampus* 19, 456–479 (2009)
12. Touretzky, D.S., Redish, A.D., Wan, H.: Neural representations of space using sinusoidal arrays. *Neural Computation* 5, 869–884 (1993)

Model Studies on Time-Scaled Phase Response Curves and Synchronization Transition

Yasuomi D. Sato^{1,2}, Keiji Okumura³, and Masatoshi Shiino³

¹ Department of Brain Science and Engineering, Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology,
2-4, Hibikino, Wakamatsu-ku, Kitakyushu, 808-0196, Japan
sato-y@brain.kyutech.ac.jp

² Frankfurt Institute for Advanced Studies, Johann Wolfgang Goethe University,
Ruth-Moufang-Str. 1, 60438 Frankfurt am Main, Germany
sato@fias.uni-frankfurt.de

³ Department of Physics, Faculty of Science, Tokyo Institute of Technology,
2-12-1, Oh-okayama, Meguro-ku, Tokyo 152-8855, Japan
{kokumura,mshiino}@mikan.ap.titech.ac.jp

Abstract. We studied possibilities of classification of single spike neuron models by their intrinsic timescale parameters, because little is known about changes of timescale on spiking dynamics, and its influence on other spike properties and network dynamics such as synchronization. Using both FitzHugh-Nagumo (FHN) type and Terman-Wang (TW) type of theoretically tractable models, analysis of the phase response curve (PRC) found common and unique dynamic characteristics with respect to two parameters of timescale and injected current amplitude in the models. Also, a scheme of synchronization transition in the identical pair systems, in which two identical models mutually interact through the same model of synaptic response, was systematically explained by controlling these parameters. Then we found their common and unique synchronous behaviors.

Keywords: Time Scale Controlling, Deterministic Phase Reduction, Phase Response Curves, Single Spike Neuron Models.

1 Introduction

Knowing dynamical mechanisms on generation of coherent behavior of small or large ensembles of nonlinear oscillators is very interesting topics in the research fields of statistical physics as well as computational neuroscience. In particular, locally and globally synchronous behavior between neuronal ensembles has been considered to play an important role in understanding brain mechanisms of the information processing. One of the representatives is “Binding-by-Synchrony.” The purely theoretical formulations of this “Binding-by-Synchrony” hypothesis were proposed earlier by [1].

To understand dynamical mechanisms of such synchronization generated in the neural network in any case, we will have to predominantly know about

dynamical properties of the spiking model within the networks, for instance, physiological conductance-based models of the Hodgkin-Huxley type [2] and the Morris-Lecar type [3], or theoretical tractable models such as the FitzHugh-Nagumo (FHN) type [4] and the Terman-Wang (TW) type [5]. As usual, their spike dynamics consists of fast and slow dynamics and varies by controlling a time scale between the two dynamics. Phenomenologically, they behave as if firing frequencies as well as action potential durations vary with changes of temperature. In the research field of computational neuroscience, it has been well-known that these models can often be classified into two types, with analyses of phase response curves (PRCs) [6] and a constant inject current amplitude-firing frequency properties [7]. However, it is still unknown whether spiking models can be separated by controlling timescales between their inherent fast and slow dynamics.

The FHN and TW spiking models are employed in this paper. They are different types, in terms of recovery dynamics. One is the monotonic linear increase while the other is the sigmoid-like curve. The slow dynamics for the recovery variable is time-scaled by μ (called a time scale parameter), relative to the first dynamics for the membrane potential variable. We then compute each of their phase response curves (PRCs) within a framework of the phase reduction method [8], to show difference between the PRC shape shifts controlled by the time scale parameter μ . Following the PRC analysis results for each type, we also simulate behavior of a pair of the identical models that coupled each other through the model of synaptic response. The synaptic model is based on two-dimensional ordinary differential equations time-scaled by μ , assuming that timescales for post synaptic potential are slower than the one for membrane potential. The equation originally has solutions of the α -function [9]. Then, we demonstrate different or common synchronous behavior in the two-neuron system, in terms of the model type.

2 Spiking Neuron Models

We will begin by studying timescale effects on spiking properties of both the FHN and TW types, with a physiological plausible assumption that a timescale for a membrane potential variable is faster than one for recovery and synaptic potential variables [10] and these slow variables are commonly time-scaled by μ :

$$\frac{dV}{dt} = -\frac{1}{3}V^3 + V - W + I_{app}, \quad (1)$$

$$\frac{dW}{dt} = \mu g(V, W), \quad (2)$$

$$\frac{ds}{dt} = \frac{\mu}{\tau_{syn}}(-s + h), \quad (3)$$

$$\frac{dh}{dt} = \frac{\mu}{\tau_{syn}}(-h + H(V)), \quad (4)$$

where

$$g(V, W) = \begin{cases} V + a - bW, & \text{for FHN} \\ \gamma(1 + \tanh(V/\beta)) - W, & \text{for TW} \end{cases},$$

$$H(V) = \begin{cases} 1, & x > \Theta \\ 0, & x \leq \Theta \end{cases}.$$

Here τ_{syn} is a synaptic time constant. $a = 0.8$ and $b = 0.01$ for the FHN model. The FHN model exhibits repetitive firings when $I_{app} > [-58/3]$. Meanwhile $\gamma = 4.0$ and $\beta = 0.1$ for the TW model with $I_{app} > [2/3]$. $H(V)$ is a Heaviside step function with the threshold taking $\Theta = 0.0$.

2.1 Phase Response Curves

We study how timescale and applied current variations influence shape of the PRC for each of the FHN and the TW types. The PRCs are typically obtained by a phase reduction method as described in Appendix. PRCs can frequently be divided into two types: Type-I PRC is well-known to have nearly all positive values while Type-II has both positive and negative values.

For the TW model, we found two types of different PRC shape shifts, $I_{app} = 0.67$ and $I_{app} = 0.9$. In the first case, when $\mu = 0.5$, we showed Type-I PRC, which was frequently observed in previous PRC analysis. However, with a decrease to $\mu = 0.05$, the bell-shape of the PRC with positive values became sharper, with less negative regions in the PRC but similar in appearance to the Type-II previously mentioned. We have also found changes of membrane potential dynamics. In particular, amplitude of the action potential varies drastically with changes in the time-scale (Fig.1(a)).

Next, we analyzed the second case, $I_{app} = 0.9$. PRC apparently takes negative values as well, regardless of timescale value (Fig.1(b)). To be more precise, the PRC has a minus region in action potential duration. The other region of the PRC in the second case shows a very similar bell-shape shift to the first case.

We then briefly examined the similarities and differences between the PRCs of the aforementioned TW type and FHN types, to determine dynamic mechanisms of how a pair system coupled with identical model types behaves. For the FHN type, we obtained the well-known Type-II PRC, independent of different values of μ and I_{app} as shown in Fig.1(c). This result indicates that the FHN type and the TW type share a common PRC shape with the typical Type-II when their timescale parameter is small. The common PRC shape is independent of I_{app} values. However when μ is larger, the TW type's PRC, dependent on I_{app} , takes a different form from that of the FHN type.

3 A Pair of Coupled Spiking Models

Since we have found how PRC shapes of the TW and FHN types are shifted by changing the value of the time scale, it would be interesting to analyze the

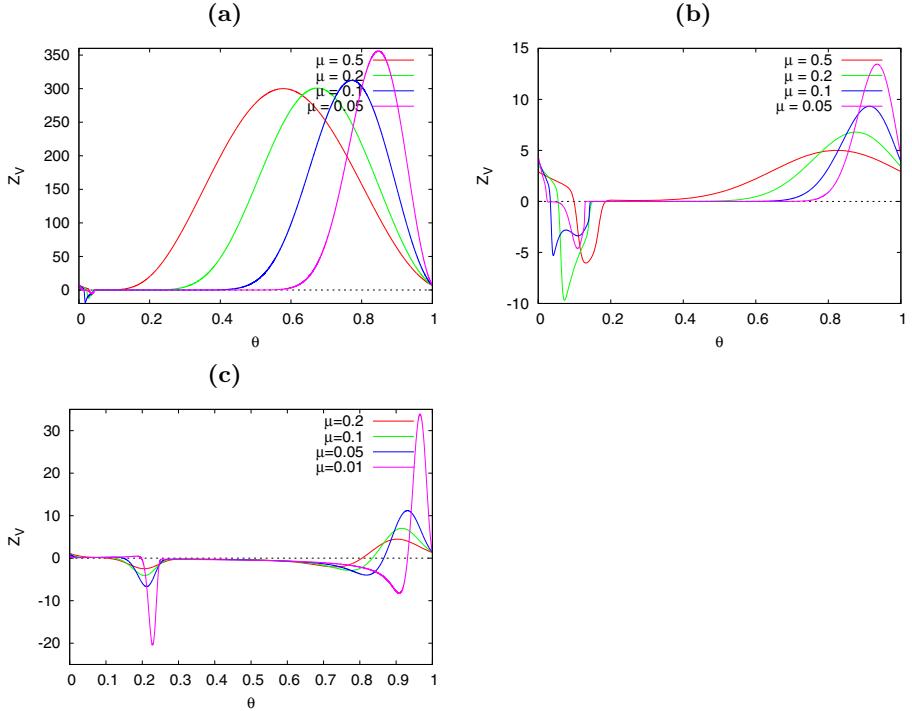


Fig. 1. μ -modulated phase response curves (PRCs) ($Z_V(\theta)$). (a) and (b) show cases for the TW type with $I_{app} = 0.67$ and 0.9, respectively. (c) is a case for the FHN type with $I_{app} = 0.0$.

dynamic mechanisms of time-scale modulated synchronization even in the pair system in which neurons excitatorily interact with each other:

$$\frac{d\mathbf{x}^{(n)}}{dt} = \mathbf{F}^{(n)}(\mathbf{x}^{(n)}) + \epsilon \mathbf{G}^{(n)}(\mathbf{x}^{(n)}, \mathbf{x}^{(\bar{n})}), \quad (n = 1, 2). \quad (5)$$

The neuronal state $\mathbf{x} = (V, W, s, h)^T \in \mathbb{R}^4$ and T denotes a transpose. \bar{n} represents a counterpart of the n th neuron. $\mathbf{F}^{(n)}(\mathbf{x}^{(n)})$ represents right-hand sides of the Eqs.(1)–(4). The excitatory synaptic coupling vector $\epsilon \mathbf{G}^{(n)}(\mathbf{x}^{(n)}, \mathbf{x}^{(\bar{n})}) = (\epsilon s^{(\bar{n})}, 0, 0, 0)^T$ is weak ($0 < \epsilon \ll 1$). Following the previous subsection, a phase reduction method is also used to analyze synchronization phenomena in the two-neuron system. In the phase reduction method, the weakly coupled oscillatory models of the TW and FHN types can be reduced to equations consisting of the phase degrees of freedom [Appendix]. The obtained phase equations are given as follows:

$$\frac{d\phi}{dt} = \epsilon H_2(-\phi) - \epsilon H_1(\phi) \equiv \Gamma(\phi), \quad (6)$$

where

$$H_n(\phi) = \frac{1}{T_p} \int_0^1 \{\mathbf{Z}^{(n)}(\theta)\}^T \cdot \mathbf{G}^{(n)} \left(\mathbf{x}_p^{(n)}(\theta), \mathbf{x}_p^{(n)}(\theta + \phi) \right) d\theta. \quad (7)$$

where $\phi = \theta_1 - \theta_2$ denotes the phase difference between the two neurons. $\Gamma(\phi)$ is expressed as an average interaction function of ϕ . The V -component of $\{\mathbf{Z}^{(n)}(t)\}^T$, $Z_V(t)$ is a so-called PRC.

The linear stability analysis for Eq.(6) provides a simple explanation of how the stationary synchronous states arise in a two-neuron system. Synchronous solutions are represented as fixed points, ϕ_0 , being satisfied with the stationary condition $G(\phi_0) = 0.0$. The inphase and antiphase synchronous solutions are defined respectively as $\phi_0 = 0.0$ or 1.0 , and $\phi_0 = 0.5$. In general, for $\epsilon > 0$, such synchronous solutions are stable if $G'(\phi_0) < 0$, while it is unstable if $G'(\phi_0) > 0$.

3.1 For the TW Type Model

We showed how a time-scaling factor μ affects the scheme of synchronization of oscillations with respect to α by the stability analysis for $G(\phi)$. Numeric calculations were used for all results obtained by using the linear stability analysis for the phase equation.

We investigated the case for $I_{app} = 0.67$ in the α region $(0.0, 2.0]$. As shown in Fig. 2(a), for $\mu = 0.05$, we found two supercritical pitchfork bifurcations around $\alpha = 0.1$ and $= 0.3$. In particular, the bifurcation around $\alpha = 0.3$ shows that the stable antiphase synchronous solution undergoes a transition to two stable inphase synchronous solutions and one unstable antiphase synchronous solution with an increase of α . However, when $\mu > 0.05$, the supercritical pitchfork bifurcation around $\alpha = 0.1$ for $\mu = 0.5$ disappears while the other supercritical pitchfork bifurcation breaks down to transit subcritical pitchfork bifurcation with additionally new saddle-node bifurcations as shown in Fig. 2(b). Conversely, for much smaller μ , the two supercritical pitchfork bifurcation points collide with each other to transit to inphase synchronous states of $\mu = 0.0$ or 1.0 . In addition, another new subcritical pitchfork bifurcation appears for larger α .

We computed the linear stability for ϕ when $I_{app} = 0.9$. We then found a trivial pitchfork bifurcation when $\mu = 0.5$. This was already observed by Van Vreeswijk et al. in the integrate-and-fire neuronal oscillators with a-function derived from Eqs. (3) and (4) [12,13]. However when decreasing gradually, we observed numerically a similar synchronization transition to the case of $I_{app} = 0.67$, namely, from two supercritical pitchfork bifurcations' results to one subcritical pitchfork bifurcation result.

3.2 For the FHN Type Model

Synchronous behavior of two coupled FHN type oscillators was also studied. For large ϕ , there has no bifurcation. Independent of α , $\phi = 0.0$ and 1.0 are stable while $\phi = 0.5$ is unstable. When μ decreases to 0, one new subcritical pitchfork

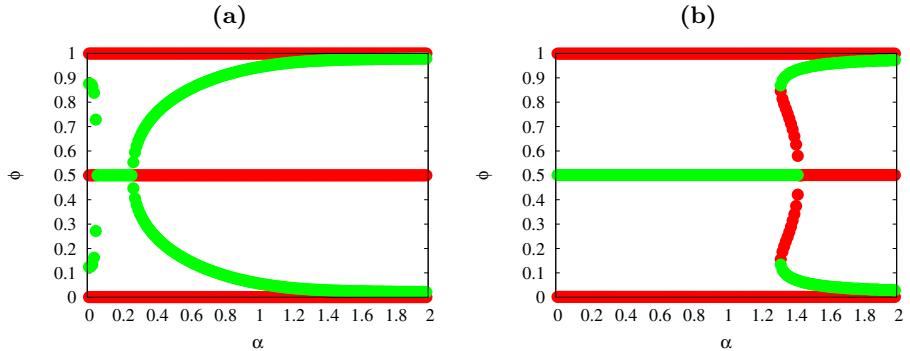


Fig. 2. Bifurcation diagrams for synchronous states ϕ of a pair system of the TW type with $I_{app} = 0.67$ in terms of α . (a) $\mu = 0.05$. (b) $\mu = 0.5$. Light green and red lines represent stable and unstable states, respectively.

bifurcation point appears. When α increases, one unstable antiphase synchronized solution transits two new unstable synchronized solutions and one stable antiphase synchronized solution. Such synchronization transitions controlled by the timescale of the firing dynamics do not rely on amplitude of the injected current. These numeric results are not as complicated as numeric results for the TW type model. But we have found a common ϕ - α bifurcation diagram in the smaller μ region, regardless of the model type.

4 Discussions and Conclusions

In this work, using two different types of spiking oscillators, the TW type and the FHN type, we have analyzed their PRC shape shift influenced by the timescale parameter. For the TW type, we found drastic change of the so-called Type-I PRC to the Type-II when the injected current amplitude I_{app} takes the value immediately after firing oscillation just started. For larger I_{app} , the TW type model has no apparent PRC shape shift, keeping the so-called Type-II PRC with both negative and positive values. The lack of PRC shape shift is also observed for the FHN type.

From these points of views, it is the most interesting point that both the FHN and TW models have the common form of the Type-II PRC in a smaller μ region, independent of I_{app} . This also indicates that there is no difference between dynamic properties of the FHN and TW models when μ is so small. On the other hand, the reason why the PRCs for the TW type are different from the ones for FHN type when μ is relatively large can be related to well-known saddle-node bifurcation descriptions: For increasing I_{app} , the V-nullcline shifts upwards in order for the stable fixed point and the saddle to merge with each other and then disappear. The unstable fixed point remains, around which there must be a limit cycle. The limit cycle passes through the two merging fixed points where the

velocity of the trajectory is extraordinary slow. Correspondingly, the PRC takes very high positive values over most phases because of a normalization condition written in Appendix. However, no saddle-bifurcation description is seen for the FHN type. So, the relevant PRC takes not only positive but also negative values.

In any case, we will have to analyze and theoretically understand mechanisms of switching of nonlinear dynamical properties of the TW model are related to controlling the timescales for dynamics. However, in this work, we found a common PRC shape between the FHN and TW models for smaller μ . Also, in such a small μ region, we demonstrated the same synchronization phenomena in a pair of two coupled identical models.

Acknowledgments. Y.D.S. participated in the “Bernstein Focus: Neurotechnology through research grant 01GQ0840” funded by the German Federal Ministry of Education and Research (BMBF). This work was supported by the Hertha Foundation and by the Grant-in-Aid for Young Scientist (B) No. 22700237. K.O. was financially supported by the Global Center of Excellence Program by MEXT, Japan through the “Nanoscience and Quantum Physics” Project of the Tokyo Institute of Technology.

References

1. von der Malsburg, C.: The correlation theory of brain function. Internal Report 81-2, Dept. of Neurobiology, Max-Planck-Institute for Biophysical Chemistry, Göttingen, Germany (1981)
2. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117(4), 500–544 (1952)
3. Morris, C., Lecar, H.: Voltage oscillations in the barnacle giant muscle fiber. *Bioophys. J.* 35(1), 193–213 (1981)
4. FitzHugh, R.: Mathematical models of excitation and propagation in nerve. In: Schwan, H.P. (ed.) *Biological Engineering*. ch. 1, pp. 1–85. McGraw-Hill Book Co, N.Y (1969)
5. Terman, D., Wang, D.L.: Global competition and local cooperation in a network of neural oscillators. *Physica D* 81, 148–176 (1995)
6. Ermentrout, B.: Type I membranes, phase resetting curves and synchrony. *Neural Computation* 8, 979–1001 (1996)
7. Gerstner, W., Kistler, W.M.: *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, New York (2002)
8. Kuramoto, Y.: *Chemical Oscillations, Waves, and Turbulence*. Springer, Tokyo (1984)
9. Rall, W.: Distinguishing theoretical synaptic potentials computed for different soma-dendritic distributions of synaptic inputs. *J. Neurophysiol.* 30, 1138–1168 (1967)
10. Frankel, P., Kiemel, T.: Relative phase behavior of two slowly coupled oscillators. *SIAM J. Appl. Math.* 53, 1436–1446 (1993)
11. Sato, Y.D.: *Synchronization Phenomena in a Pair of Coupled Neuronal Oscillator Systems*. PhD Thesis, Tokyo Institute of Technology (2005)

12. Van Vreeswijk, C., Abbott, L.F., Ermentrout, G.B.: Inhibition, not excitation, synchronizes coupled neurons. *J. Comput. Neurosci.* 1, 303–313 (1994)
13. Van Vreeswijk, C.: Partially synchronized states in networks of pulse-coupled neurons. *Phys. Rev. E* 54, 5522–5537 (1996)
14. Mehrotra, A., Sangiovanni-Vincentelli, A.: Noise Analysis of Radio Frequency Circuits. Kluwer Academic Publishers, Dordrecht (2004)
15. Sato, Y.D., Shiino, M.: Generalization of coupled spiking models and effects of the width of an action potential on synchronization phenomena. *Phys. Rev. E* 75, 011909 (2007)

Appendix: Phase Reduction Method

The general formulation of a phase reduction method is presented [14][15]. Let $\mathbf{x}_p^{(n)}(t)$ be a stable T_p -periodic solution for the uncoupled dynamics,

$$\dot{\mathbf{x}}_p^{(n)} = \mathbf{F}^{(n)}(\mathbf{x}_p^{(n)}), \quad \mathbf{x}_p^{(n)}(t + T_p) = \mathbf{x}_p^{(n)}(t). \quad (8)$$

The stable solution for the evolution equation (2) is approximated as

$$\mathbf{x}^{(n)}(t) = \mathbf{x}_p^{(n)}(t + \tau^{(n)}) + \epsilon \mathbf{u}^{(n)}(t + \tau^{(n)}(t)), \quad (9)$$

where $\tau^{(n)}(t)$ means a small perturbation in the phase direction on the periodic orbit, and $\mathbf{u}^{(n)}$ is orbital deviation from the periodic orbit, $\mathbf{x}_p^{(n)}$. Substituting Eq. (9) into Eq. (5) and expanding both the sides into a Taylor series, we obtain the time evolution equation for $\tau^{(n)}$:

$$\dot{\tau}^{(n)} = \epsilon \{ \mathbf{Z}^{(n)}(t + \tau^{(n)}) \}^T \cdot \mathbf{G}^{(n)} \left(\mathbf{x}_p^{(n)}(t + \tau^{(n)}), \mathbf{x}_p^{(\bar{n})}(t + \tau^{(\bar{n})}) \right), \quad (10)$$

using the normalization condition $\{ \mathbf{Z}^{(n)}(t) \}^T \cdot [d\mathbf{x}_p^{(n)}(t)/dt] = 1$ for every t . $\{ \mathbf{Z}^{(n)}(t) \}^T$ is the unique solution to

$$\frac{d\mathbf{Z}^{(n)}(t)}{dt} = - \left[\frac{\partial \mathbf{F}^{(n)} \left(\mathbf{x}_p^{(n)}(t) \right)}{\partial \mathbf{x}_p^{(n)}(t)} \right]^T \mathbf{Z}^{(n)}(t). \quad (11)$$

The first component of $\{ \mathbf{Z}^{(n)}(t) \}^T$ can often be the phase response curve (PRC). Introducing phase variables $\theta_n(t) = (t + \tau^{(n)}(t))/T_p$, Eq. (10) can be rewritten as

$$\frac{d\theta_n}{dt} = \frac{1}{T_p} + \frac{\epsilon}{T_p} \{ \mathbf{Z}^{(n)}(\theta_n) \}^T \cdot \mathbf{G}^{(n)} \left(\mathbf{x}_p^{(n)}(\theta_n), \mathbf{x}_p^{(\bar{n})}(\theta_{\bar{n}}) \right), \quad (12)$$

Let the phase difference of the two oscillators $\phi(t) = \theta_2(t) - \theta_1(t)$ be a slow variable. We have obtained the evolution equation for the phase difference as

$$\frac{d\phi}{dt} = \epsilon H_2(-\phi) - \epsilon H_1(\phi) \equiv \Gamma(\phi), \quad (13)$$

where

$$H_n(\phi) = \frac{1}{T_p} \int_0^1 \{ \mathbf{Z}^{(n)}(\theta) \}^T \cdot \mathbf{G}^{(n)} \left(\mathbf{x}_p^{(n)}(\theta), \mathbf{x}_p^{(\bar{n})}(\theta + \phi) \right) d\theta. \quad (14)$$

Roles of Early Vision for the Dynamics of Border-Ownership Selective Neurons

Nobuhiko Wagatsuma^{1,2} and Ko Sakai³

¹ Research Fellow of the Japan Society for the Promotion of Science

² Laboratory for Neural Circuit Theory, Brain Science Institute, RIKEN

2-1 Hirosawa, Wako, Saitama, 351-0198, Japan

nwagatsuma@barain.riken.jp

³ Graduate School of Systems and Information Engineering, University of Tsukuba,

1-1-1 Ten-nodai, Tsukuba, Ibaraki, 305-8573, Japan

sakai@cs.tsukuba.ac.jp

Abstract. The border ownership (BO) that indicates which side of the contour owns the border plays a fundamental role in object perception[1]. The responses of BO-selective cells exhibit rapid transition when a stimulus is fliped along its classical receptive field so that the opposite BO is presented, while the transition is significantly slower when a clear BO is turned into an ambiguous edge such as when a square is enlarged extensively[2]. This phenomenon appears to be a crucial clue for understanding the neural mechanisms underlying the credibility of BO. We hypothesize that dynamics of BO-selective cells and networks behind them play a crucial role in the credibility, and that the credibility is related to early visual areas as an appearance of a salient object evokes bottom-up attention. To investigate these hypotheses, we examined the dynamics of BO-selective cells with a computational model that include recurrent pathways among V1, V2 and Posterior Parietal (PP) areas[3]. The model cells have been shown to reproduce effects of spatial attention. Simulations of the model exhibited distinct response time depending on the ambiguity of BO, indicating a crucial role of dynamics in the credibility. The recurrent network between PP and V1 appear to play a crucial role for the time course of BO-selective cells that governs simultaneously both credibility of BO and bottom-up attention.

Keywords: Border-ownership (BO), visual attention, early visual areas, time course, computational model.

1 Introduction

Neural mechanisms underlying figure/ground segregation have been studied for decades. In particular, the determination of the direction of figure (DOF) is the most fundamental process to perceive and understand objects and their locations. Physiological studies have reported that a majority of neurons in monkey's V2 and V4 show selectivity to border ownership (BO): their responses change depending on which side of a border owns the contour [e.g. 1]. The characteristics of BO-selective neurons are currently under investigation. Attentional effect has been reported on these responses

to BO signals [4]. Moreover, the persistence of the BO signal depends on the ambiguity of DOF [2]. Stimuli with clear DOF induce faster modification of the responses of BO-selective cells than stimuli with ambiguous DOF.

Computational studies have suggested that BO is determined by the surrounding suppression/facilitation observed in early visual areas; luminance contrast around the classical receptive field (CRF) is thus crucial for the determination of BO [5]. Based on that this idea, we have proposed a network model that account for the switch of DOF based on the modulation by attention in the early visual area [3, 6]. The model consists of mutually connected V1, V2, and posterior parietal (PP) modules. Spatial attention from PP alters and enhances the contrast gain in V1. This modification of the surrounding contrast in the early vision leads to the modulation of the activities of BO-selective model cells in V2. The model showed good agreement with human perception in the attentional modulation of response magnitude. It is expected that the model also reproduces the time course of the response, specifically the persistence of the BO signal depending on the ambiguity of DOF.

We investigated the dynamics of the proposed network model in order to address the problems regarding dynamics of BO signaling. Specifically, we aim to examine :

- V1-PP network originates the time course of BO-selective cells depending on the ambiguity of the DOF and
- Inhibition of return through attention plays a critical role in perceiving and detecting a new object on the retina.

2 The Model

The architecture of the model is based on [3], in which spatial attention alters the contrast gain in early vision, which in turns modifies the activities of BO-selective model cells. The model consists of three entities: V1, V2 and PP module, as illustrated in Fig.1. Top-down and bottom-up pathways mutually connect these modules, except for PP to V2. We excluded PP to V2 connection to avoid direct influence of the attention to BO-selective model cells.

Each module consists of 100x100 cells distributed retinotopically. In the absence of external input, the activities of a cell at time t , $A(t)$, is given by

$$\tau \frac{\partial A(t)}{\partial t} = -A(t) + \mu F(A(t)), \quad (1)$$

where the first term on the right-hand side corresponds to a decay, and the second term shows the excitatory, recurrent signals among the excitatory model cells. The non-linear function, $F(x)$, is given by

$$F(x(t)) = \frac{1}{T_r - \tau \log(1 - (1/\alpha(t)))}, \quad (2)$$

where τ is a membrane time-constant, and T_r an absolute refractory period. These time constants depend on whether this neuron is excitatory or inhibitory. However, these parameters are common to all visual modules [7, 8]. The dynamics of equations as well as appropriate values for constants have been extensively studied [8, 9].

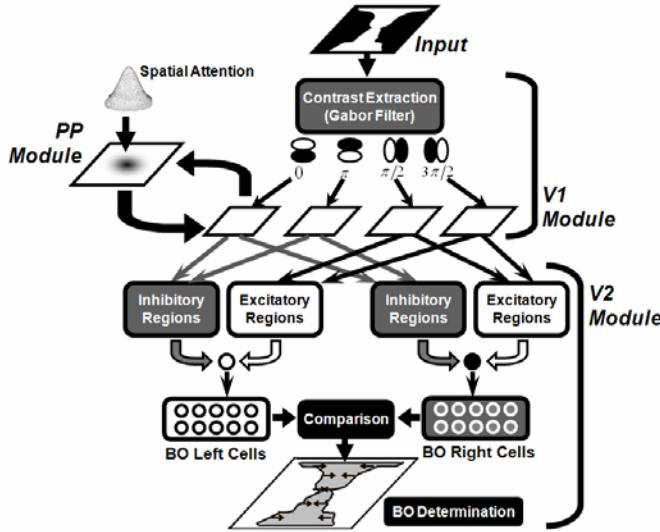


Fig. 1. An illustration of the model architecture[3]. This model consists of three modules, V1, V2 and PP, with mutual connections, except for PP to V2 pathway to avoid the direct influence of attention from PP to V2.

2.1 V1 Module

The V1 module models the primary visual cortex, in which local contrast is extracted from an input stimulus; in addition, spatial attention modulates the contrast gain. The input image *Input* is a 124x124 pixel, gray scale image with intensity values ranging between zero and one.

The local contrast $C_{\theta\omega}(x, y, t)$ is extracted by the convolution of the image with a Gabor filter, $G_{\theta\omega}$,

$$C_{\theta\omega}(x, y, t) = \text{Input}(x, y, t) * G_{\theta\omega}(x, y), \quad (3)$$

where indices x and y are spatial positions, and ω represents spatial frequency. Orientation, θ , was selected from $0, \pi/2, \pi$ and $3\pi/2$. The extracted contrast is modulated by spatial attention, thus the contrast at the attended location is enhanced. The activity of a model cell in V1 module, $A_{\theta\omega xy}^{V1}$, is given by

$$\tau \frac{\partial A_{\theta\omega xy}^{V1}(t)}{\partial t} = -A_{\theta\omega xy}^{V1}(t) + \mu F(A_{\theta\omega xy}^{V1}(t)) + I_{\theta\omega xy}^{V1,E}(t) + I_o, \quad (4)$$

where I_o is random noise, and μ is a factor that scales contrast. The local contrast, $C_{\theta\omega}$, is modulated by the feedback from PP to V1, I_{xy}^{V1-PP} , denoted by [10]:

$$I_{\theta\omega xy}^{V1,E}(t) = \frac{(C_{\theta\omega}(x, y, t))^{\gamma I_{xy}^{V1-PP}(t)}}{S^{\delta I_{xy}^{V1-PP}(t)} + \sum_{\theta\omega} \left(\frac{1}{(2I+1)(2J+1)} \sum_{j=-J}^J \sum_{i=-I}^I C_{\theta\omega}(x+j, y+i, t) \right)^{\delta I_{xy}^{V1-PP}(t)}}, \quad (5)$$

where S in eq(5) prevents the denominator to become zero. γ and δ are constants and scale the attetional strength.

2.2 V2 Module

The V2 module models BO-selective cells reported in V2 that determine the direction of BO. Activities of the BO-selective model cells is determined based on the surrounding contrast signal extracted by the V1 module, as illustrated in Fig.2[5]. Each BO-selective model cell has single excitatory and inhibitory regions. The activity of a BO-selective model cell is modulated based on the location and shape of these surrounding regions. To reproduce a wide variety of BO selectivity, we implemented ten types of left and right BO-selective model cells with distinct surrounding regions.

The activity of a BO-selective model cell is given by

$$\tau \frac{\partial A_{xyN}^{V2,BO}(t)}{\partial t} = -A_{xyN}^{V2,BO}(t) + \mu F(A_{xyN}^{V2,BO}(t)) - \gamma F(A_{xyN}^{V2,inh}(t)) + I_{xyN}^{V2-V1,BO}(t) + I_o, \quad (6)$$

where $I_{xyN}^{V2-V1,BO}$ represents bottom-up input from V1. An index BO shows left- or right-BO selectivity, and N represents the type of BO-selective model cells that is distinguished by their surround region. If right BO-selective model cells are more active than left cells, a figure is judged as located on the right side. The third term of the equation represents the input from inhibitory cells that gathers signals from all model cells in this module. The activity of an inhibitory V2 model-cell is given by

$$\tau \frac{\partial A^{V2,inh}(t)}{\partial t} = -A^{V2,inh}(t) + \mu F(A^{V2,inh}(t)) + \kappa \sum_{Nxy} F(A_{xyN}^{V2,BO}(t)), \quad (7)$$

where κ is a constant. This inhibitory cell receives inputs from excitatory neurons in V2, and inhibits these neurons.

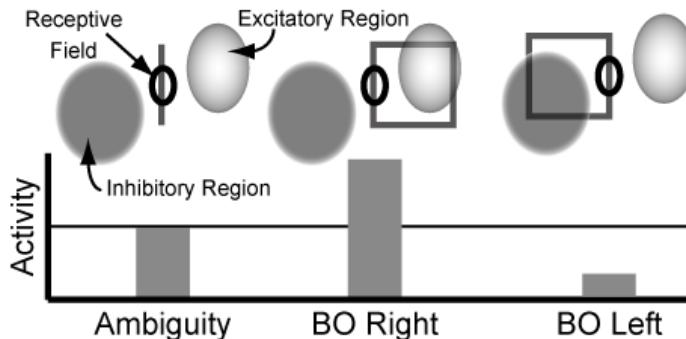


Fig. 2. A mechanism of right BO-selective cell [5]. If there is contrast in a excitatory surrounding region, the activity of the cell is enhanced. If the contrast exists in a inhibitory region, the activity is suppressed. The balance of the right and left BO-selective cells determines the DOF on the stimulus.

2.3 Posterior Parietal (PP) Module

PP module encodes spatial location and aims to facilitating the processing of the attended location. The presence of spatial attention is given explicitly in this module, which will boost the contrast gain of the location in V1 module.

PP module receives bottom-up inputs from V1 module. The activity of an excitatory model-cell in PP module is given by

$$\tau \frac{\partial A_{xy}^{PP}(t)}{\partial t} = -A_{xy}^{PP}(t) + \mu F(A_{xy}^{PP}(t)) - \gamma F(A^{PP,inh}(t)) + I_{xy}^{PP-V1}(t) + I_{xy}^{PP,A}(t) + I_o. \quad (8)$$

$I_{xy}^{PP,A}$ represents the bias of spatial attention. I_{xy}^{PP-V1} represents bottom-up inputs from V1 to PP module, which behaves in a similar fashion to a saliency map based on luminance contrast. When there is no attentional bias ($I_{xy}^{PP,A} = 0$), PP module is activated by bottom-up signals from V1. The third term describes influx from the inhibitory PP model-cell. The activity of the inhibitory cell is determined from the activities of all excitatory PP cells as in eq(7). The PP module encodes spatial location, and facilitates the processing in and around the attended location in V1. In this model spatial attention does not directly affect BO-selective model-cells in V2 module, because the main focus of this model has been the effect of spatial attention in early vision V1 (details are in [3]).

3 Simulation Results -Dynamics of the Model Neurons

To examine the dynamics of the proposed model, we carried out simulations with the stimulus corresponding to O'Herron's experiment[2]. In their experiment, a single square was presented with its right edge aligned along the CRF of BO-selective cell. The square was then enlarged, as a clear DOF (to the left) was changed to an ambiguous DOF of the midline though the screen center, as illustrated in Fig.3(a). The responses of BO-selective cells decreased slowly with this switch to ambiguous edge, in comparison with the switch between clear DOFs as observed in flipping the square with respect to the CRF (Fig.3(b)). This difference in time course appears to be a crucial clue for understanding the dynamics of BO-selective cells and network behind them.

3.1 The Dynamics of BO-Selective Model Cells in V2 Module

We carried out the simulations of the model with the stimuli replacing BO-left square with BO-right square or the ambiguity. Fig. 3 (c) shows the time course of the BO-selective model cells in V2 module with stimuli corresponding to O'Herron's experiment. We compute the difference between responses of left and right BO-selective cells to compare the magnitude of left- with right-BO signals. Black line in Fig. 3 (c) represents in the case of Fig. 3 (a). White line was in the case of Fig. 3 (b). Positive responses indicates the dominance of left BO-selective cells and likewise with negative values for right BO-selective cells. We replaced the stimulus after 25 iterations shown by a dashed line in Fig. 3 (c).

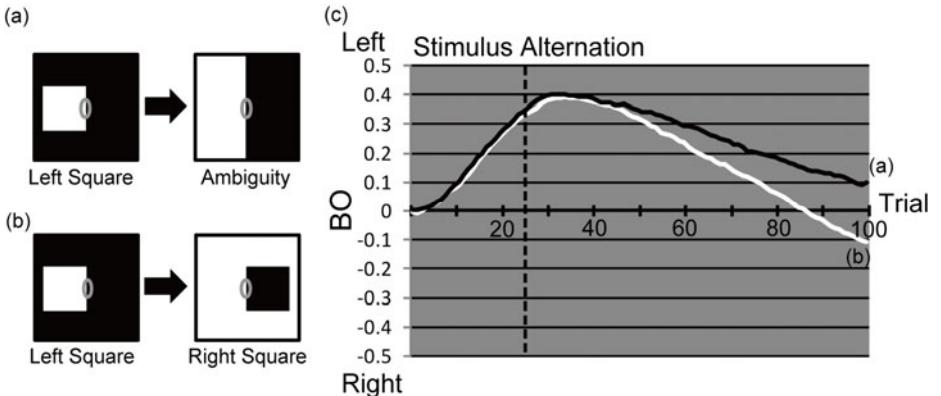


Fig. 3. Stimuli for the corresponding physiological experiment[2] and the time course of BO-selective model cells in the V2 module. (a) A square located on the left of the CRF was replaced by an ambiguous edge. (b) A square was switched from the left to right. Gray ellipses represent the location of the CRF of BO-selective model cells in the V2 module. Note that the local contrast projected onto the CRF was identical throughout the stimuli. (c) The time course of BO-selective model cells in the V2 module. A dashed vertical line indicates the 25 iterations where a stimulus was replaced. Black and white lines show the responses to the stimulus set (a) and (b), respectively. The activities of BO-selective model cells for the stimulus set (b) without ambiguity were more rapidly modulated compared with (a) including ambiguous BO.

When BO-left was switched to BO-right (stimulus set (b); a white line), the responses of BO-selective model cells were modified more quickly than when the BO-left was replaced with ambiguity (stimulus set (a); a black line). The dynamics of the BO-selective model cells are affected by ambiguity of the BO signal in a manner similar to physiological observations[2].

3.2 Time Course of PP Module

The dynamics of the BO-selective model cell in V2 module showed the same qualitative tendency to the physiological results. However, more precisely, the modification of monkey's BO-selective cells appears to be faster than our model. In order to investigate the reason for this difference between the model and biology, we visualized the responses of PP module to stimulus set (b). Recall that PP module encodes the location of the object. Fig. 4 displays the time course of responses of PP module to the stimulus replacing BO-left square with BO-right. The switching between stimuli occurred after 25 iterations. However, the presentation of new stimulus does not seem sufficient to reverse the activation of the PP module compared to the initial pattern induced by first stimulus, even after the 100 iterations. A hypothesis is that a lack of inhibition of return[11] prevents our model from evoking a full detection of the new stimulus, as developed further in Discussion. These results suggest that not only the ambiguity of BO signal but also the V1-PP network plays a critical role for the time course of BO-selective cell in V2.

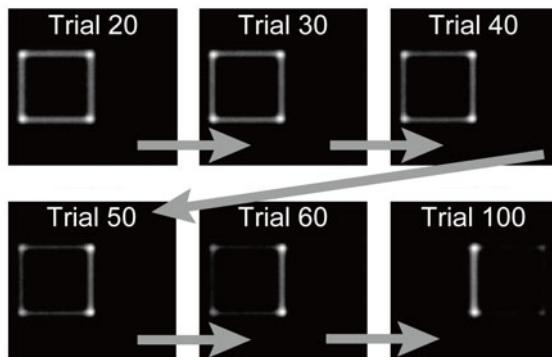


Fig. 4. The time course of the PP module for the stimulus corresponding to Fig.3 (b). We flipped the left square to right at trial 25. After the presentation of left square, the PP module did not strongly response to the right square until the end of the simulation.

4 Discussion

We have investigated the dynamics of a network model that consists of three modules representing the V1, V2 and PP cortical areas. Mutual connections between them include both bottom-up and top-down pathways, at the exception of those from PP to V2[3]. In this model, attention targets early vision and, consequently, affects the activities of BO-selective cells.

We carried out numerical simulation under the same condition as O'Herron's physiological experiment[2]. The dynamics of the BO-selective model cells exhibited the same tendency as that of monkey's cells in V2 in terms of the BO signals for incoming stimulus. However the speed of the modulation appeared to be slower in our model than in physiological results. In addition, when the switching the stimulus between BO-left and BO-right squares, PP module cannot successfully detect a newly presented stimulus following an initial stimulus.

It may be that the failure in detecting a new stimulus in PP as shown in Fig.4 is a major factor for the difference between our model and physiology on the speed of the modification of BO-selective cells. Bottom-up attention due to the detection of a new object in PP module could enhance the activities of BO-selective cells for new stimuli. The reason for which our model cannot respond to a new square with the opposite DOF may stem from a lack of the inhibition of return[11], since this biases attention away from the cued location. In our model, as a result of the sustentation of the responses to the initial stimulus in PP module, the detection of a new stimulus location could be difficult. In this way, the inhibition of return for spatial attention may play an important role to perceive and detect a new object on the retina. These results suggest that V1-PP network where attentional modulation to the early visual areas takes place is critical for the dynamics of BO-selective cells in early visual areas.

Our model predicts that bottom-up attention through the network between early vision and posterior parietal areas mediates the dynamics of BO-selective cells in V2. However, top-down attention should also modify the speed for the responses of BO-selective cells and the reaction time for the detection of DOF. It is expected to examine

these predictions from psychophysical and physiological view points. Our results provide essential and testable predictions for the fundamental problems of figure/ground segregation and attention.

References

1. Zhou, H., Friedman, H.S., von der Heydt, R.: Coding of border ownership in monkey visual cortex. *Journal of Neuroscience* 20, 6594–6611 (2000)
2. O’Herron, P., von der Heydt, R.: Short-term memory for figure-ground organization in the visual cortex. *Neuron* 61, 801–809 (2009)
3. Wagatsuma, N., Shimizu, R., Sakai, K.: Spatial attention in early vision for the perception of border ownership. *Journal of Vision* 8, 1–19 (2008)
4. Qiu, F.T., Sugihara, T., von der Heydt, R.: Figure-ground mechanisms provide structure for selective attention. *Nature Neuroscience* 10, 1492–1499 (2007)
5. Sakai, K., Nishimura, H.: Surrounding suppression and facilitation in the determination of border ownership. *Journal of Cognitive Neuroscience* 18, 562–579 (2006)
6. Wagatsuma, N., Shimizu, R., Sakai, K.: Effect of spatial attention in early vision for the modulation of the perception of border-ownership. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part I. LNCS, vol. 4984, pp. 348–357. Springer, Heidelberg (2008)
7. Rolls, E.T., Deco, G.: Computational Neuroscience of Vision (2002)
8. Deco, G., Lee, T.S.: The role of early visual cortex in visual integration: a neural model of recurrent interaction. *European Journal of Neuroscience* 20, 1089–1100 (2004)
9. Gerstner, W.: Population dynamics spiking of neuron: Fast transients, asynchronous states, and locking. *Neural Computation* 12, 43–89 (2000)
10. Lee, D.K., Itti, L., Koch, C., Braun, J.: Attention activates winner-take-all competition among visual filters. *Nature Neuroscience* 2, 375–381 (1999)
11. Itti, L., Koch, C.: Computational modelling of visual attention. *Nature Reviews Neuroscience* 2, 194–203 (2001)

Theoretical Analysis of Various Synchronizations in Pulse-Coupled Digital Spiking Neurons

Hirofumi Ijichi and Hiroyuki Torikai

Graduate School of Engineering Science, Osaka University,
1-3, Machikaneyama, Toyonaka, Osaka, Japan
ijichi@hopf.sys.es.osaka-u.ac.jp,
torikai@sys.es.osaka-u.ac.jp

Abstract. A digital spiking neuron is a wired system of shift registers that can mimic nonlinear dynamics of simplified spiking neuron models. In this paper, we present a novel pulse-coupled system of the digital spiking neurons. The coupled neurons can exhibit various pseudo-periodic synchronizations, non-periodic synchronizations, and related bifurcations. We derive theoretical parameter conditions that guarantee occurrence of typical synchronization phenomena. Also, the theoretical results are validated by numerical simulations.

Keywords: Pulse-coupled neural network, Cellular automaton, Synchronization, Bifurcation.

1 Introduction

Various spiking neuron models and their pulse-coupled networks have been studied intensively from various viewpoints, e.g., engineering applications of artificial neural networks and mathematical modelings of biological neural networks [1]-[10]. In the current of such researches, we have developed digital spiking neurons (ab. DSNs) whose concept is explained in Fig. 1. A DSN is a wired system of shift registers and can exhibit various neuron-like bifurcations and responses [8]-[10]. The dynamics of the DSN can be regarded as a special kind of *asynchronous cellular automaton* and thus it can be easily implemented in a digital hardware. We note that, in order to realize the neuron-like bifurcations and responses, the DSN do not utilize floating-point calculations (such as numerical integration) whose hardwares are power and space consuming. We also note that the parameter of the DSN is a wiring pattern among the registers and thus its dynamic parameter adjustment (learning) can be easily implemented on a dynamically reconfigurable field programmable gate array [8]. In this paper, we present and analyze a pulse-coupled system of two DSNs as shown in Fig. 1(b), where each DSN can generate various pseudo-periodic and non-periodic spike-trains depending on the wiring pattern. We derive theoretical results that guarantee occurrence of pseudo-periodic synchronizations and non-periodic synchronizations. Furthermore, we validate the theoretical results by numerical simulations. The novelties and significances of this paper include the following points. (a)

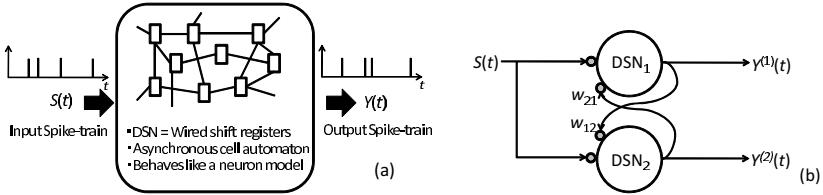


Fig. 1. (a) Concept of digital spiking neuron. (b) Pulse-coupled digital spiking neurons. In this paper, we analyze the case where $w_{21} = 0$.

This paper presents a pulse-coupled system of DSNs for the first time. So, this paper is the starting point for developing a large-scale pulse-coupled neural network of DSNs that is expected to be applicable for neural prosthesis chip with after-implant-learning capability (see [8]-[10]). (b) Synchronization phenomena of spiking neurons have been investigated intensively since they play important roles in engineering applications of artificial pulse-coupled neural networks and they are believed to play important roles in biological neural networks [6][7].

2 Digital Spiking Neuron

In this section, we introduce our *digital spiking neuron* (ab. DSN) [8] and show its various bifurcation phenomena. As shown in Fig. 2(a), the DSN consists of three parts: $M^{(1)}$ pieces of p -cells, $N^{(1)}$ pieces of x -cells and reconfigurable wires from the p -cells to the x -cells, where $^{(1)}$ is the neuron index used in a network presented in the next section. First, we explain the dynamics of the p -cells. Each p -cell has a binary state $p_m^{(1)}(t) \in \{0, 1\}$, where $m \in \{0, 1, \dots, M^{(1)} - 1\}$, $M^{(1)}$ is the number of p -cells, and $t \in [0, \infty)$ is a continuous time. In this paper, we assume that one p -cell has a state 1 and the other p -cells have states 0. Under this

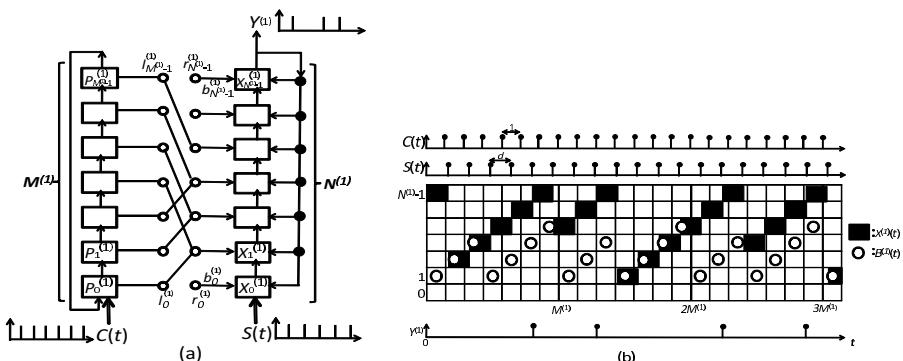


Fig. 2. (a) Digital spiking neuron (ab. DSN). (b) Time-domain waveform.

assumption, the state vector $(p_0^{(1)}(t), \dots, p_{M^{(1)}-1}^{(1)}(t))$ can be represented by the following scalar integer state $P^{(1)}(t)$: if $p_m^{(1)}(t) = 1$ then $P^{(1)}(t) = m$, $P^{(1)}(t) \in \{0, 1, \dots, M^{(1)} - 1\}$. We can assume $P^{(1)}(0) = 0$ without loss of generality. The p -cells commonly accept the following internal clock $C(t)$:

$$C(t) = \begin{cases} 1 & \text{if } t = 0, 1, 2, \dots, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

which has the normalized period of 1. The transition of the state $P^{(1)}$ is then described by

$$P^{(1)}(t+1) = P^{(1)}(t) + 1 \pmod{M^{(1)}}, \quad (2)$$

where $t \in \{0, 1, 2, \dots\}$. The state $P^{(1)}(t)$ oscillates periodically with period $M^{(1)}$. Second, we explain the reconfigurable wires. The binary state vector $(p_0^{(1)}(t), \dots, p_{M^{(1)}-1}^{(1)}(t))$ of the p -cells is input to the left terminals $(l_0^{(1)}, \dots, l_{M^{(1)}-1}^{(1)})$ of the wires. Each left terminal $l_m^{(1)}$ has one wire to one of the right terminals $(r_0^{(1)}, \dots, r_{N^{(1)}-1}^{(1)})$, and each right terminal $r_n^{(1)}$ can accept an arbitrary number of wires. In order to describe the pattern of the wires, we introduce the following wiring function $A^{(1)}(m) : \{0, 1, \dots, M^{(1)} - 1\} \rightarrow \{0, 1, \dots, N^{(1)} - 1\}$.

$$A^{(1)}(m) = n \text{ if } l_m^{(1)} \text{ is wired to } r_n^{(1)} \quad (3)$$

which is characterized by a parameter vector $\mathbf{A}^{(1)} = (A^{(1)}(0), \dots, A^{(1)}(M^{(1)} - 1))$. The right terminals $(r_0^{(1)}, \dots, r_{N^{(1)}-1}^{(1)})$ in Fig.2(a) output a binary signal vector $(b_0^{(1)}(t), \dots, b_{N^{(1)}-1}^{(1)}(t))$ which can be represented by the following scalar integer signal $B^{(1)}(t)$.

$$B^{(1)}(t) = A^{(1)}(P^{(1)}(t)). \quad \text{If } B^{(1)}(t) = n \text{ then } b_n^{(1)}(t) = 1, \quad (4)$$

where $n \in \{0, 1, \dots, N^{(1)} - 1\}$. Third, we explain the dynamics of the x -cells. Each x -cell has a binary state $x_n^{(1)}(t) \in \{0, 1\}$. In this paper, we assume that one x -cell has a state 1 and the other x -cells have states 0. Under this assumption, we can represent the state vector $(x_0^{(1)}(t), \dots, x_{N^{(1)}-1}^{(1)}(t))$ by the following scalar integer $X^{(1)}(t)$: if $x_n^{(1)} = 1$ then $X^{(1)}(t) = n$, $X^{(1)}(t) \in \{0, 1, \dots, N^{(1)} - 1\}$. In this paper, we assume $X^{(1)}(0) = N^{(1)} - 1$. From a neuron model viewpoint, the state $X^{(1)}(t)$ is regarded as a membrane potential. The x -cells commonly accept an external stimulation spike-train $S(t)$. In this paper, we focus on the following periodic stimulation spike-train.

$$S(t) = \begin{cases} 1 & \text{if } t = 0, d, 2d, \dots, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Then the transition of the membrane potential $X^{(1)}$ is described by

$$X^{(1)}(t+d) = \begin{cases} X^{(1)}(t) + 1 & \text{if } X^{(1)}(t) \neq N^{(1)} - 1, \quad (\text{shift}) \\ B^{(1)}(t+d) & \text{if } X^{(1)}(t) = N^{(1)} - 1, \quad (\text{reset}) \end{cases} \quad (6)$$

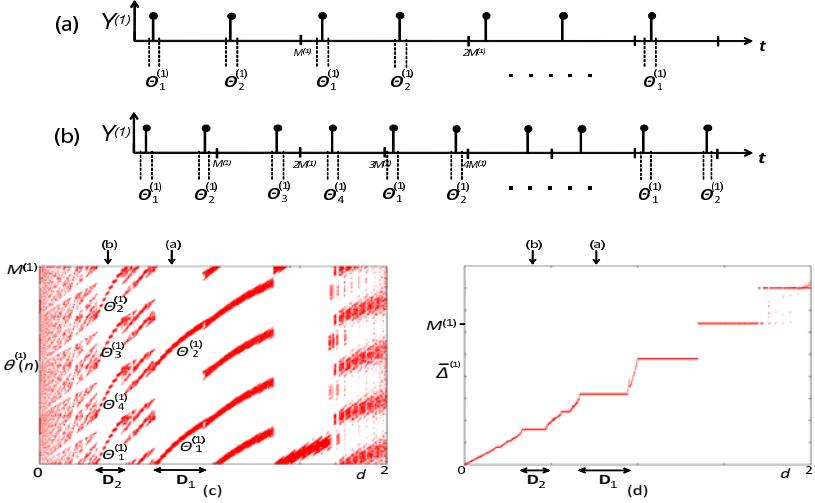


Fig. 3. Pseudo-periodic spike-trains and bifurcations. The parameter of the DSN is in Equation (9). (a) Pseudo-periodic spike-trains in 2-islands $\Theta_1^{(1)}$ and $\Theta_2^{(1)}$. The input period is $d = 0.73$. (b) Pseudo-periodic spike-trains in 4-islands $\Theta_1^{(1)}$, $\Theta_2^{(1)}$, $\Theta_3^{(1)}$ and $\Theta_4^{(1)}$. The input period is $d = 1.14$. (c) Bifurcation diagram of spike phase $\theta^{(1)}(n)$. (d) Characteristics of the average ISI $\bar{\Delta}^{(1)}$.

where $t \in \{0, d, 2d, \dots\}$. The DSN repeats the above shift and reset, and generates the following spike-train $Y^{(1)}(t)$.

$$Y^{(1)}(t) = \begin{cases} 1 & \text{if } X^{(1)}(t) = N^{(1)} - 1 \text{ and } t \in \{0, d, 2d, \dots\}, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

We note that the shift-and-reset dynamics corresponds to the integrate-and-fire dynamics of a spiking neuron model. In order to analyze the neuron dynamics, we denote the n -th spike position by $t(n)$, $t(n) \in \{0, d, 2d, \dots\}$, and introduce the following variable $\theta^{(1)}(n)$ on a co-ordinate $\theta^{(1)} = t^{(1)} \pmod{M^{(1)}}$.

$$\theta^{(1)}(n) = t^{(1)}(n) \pmod{M^{(1)}}. \quad (8)$$

In this paper, we refer to $\theta^{(1)}(n)$ as the n -th spike phase with respect to the period $M^{(1)}$ of the base signal $B^{(1)}(t)$. Figs 3(a) and (b) show examples of spike-trains $Y^{(1)}(t)$. Concerning these spike-trains, we introduce the following definition.

Definition 1. The spike-train $Y^{(1)}(t)$ is said to be *pseudo-periodic in k -islands* if there exists a positive number ϵ (which is relatively small compared to $M^{(1)}$) such that $|\theta^{(1)}(n + mk) - \theta^{(1)}(n)| \leq \epsilon$ for all $n = 1, 2, 3, \dots$ and $m = 1, 2, 3, \dots$. □

¹ If $\epsilon = 0$, then a pseudo-periodic spike-train in k -islands is a k -periodic spike-train.

For example, the spike-trains $Y^{(1)}(t)$ in Figs.3(a) and (b) are pseudo-periodic in 2-islands and 4-islands, respectively, where the DSN is characterized by the following parameter values

$$M^{(1)} = N^{(1)} = 32, \quad A^{(1)}(m) = \begin{cases} m + 8 & \text{for } 0 \leq m \leq 7, \\ m & \text{for } 8 \leq m \leq 15, \\ m - 8 & \text{for } 16 \leq m \leq 23, \\ m - 16 & \text{for } 24 \leq m \leq 31. \end{cases} \quad (9)$$

We denote existence regions of the spike phases $\theta^{(1)}(n + mk)$, $n = 1, 2, \dots$, $m = 1, 2, \dots$, by $\Theta_r^{(1)}$, $r = 1, 2, \dots, k$ as shown in a bifurcation diagram in Fig.3(c). As shown in this figure, for $d \in \mathbf{D}_1$, the spike phases exist in the 2-islands $\Theta_1^{(1)}$ and $\Theta_2^{(1)}$. These islands correspond to the quasi-periodic spike-train $Y^{(1)}(t)$ in 2-islands in Fig.3(a). For $d \in \mathbf{D}_2$, the spike phases exist in the 4-islands $\Theta_1^{(1)}$, $\Theta_2^{(1)}$, $\Theta_3^{(1)}$, and $\Theta_4^{(1)}$. These islands correspond to the quasi-periodic spike-train $Y^{(1)}(t)$ in 4-islands in Fig.3(b). In order to further characterize the spike-train $Y^{(1)}(t)$, we consider the n -th *inter-spike interval* (ISI) defined by

$$\Delta^{(1)}(n) = t^{(1)}(n + 1) - t^{(1)}(n). \quad (10)$$

We then define the following average ISI $\bar{\Delta}^{(1)}$:

$$\bar{\Delta}^{(1)} = \frac{1}{\alpha} \sum_{n=1}^{\alpha} \Delta^{(1)}(n), \quad (11)$$

where α is an appropriately large number of ISIs for averaging. Fig.3(d) shows the characteristics of the average ISI $\bar{\Delta}^{(1)}$ for the input period d . We regard the characteristics of $\bar{\Delta}^{(1)}$ as a response characteristics of the DSN to the input spike-train $S(t)$ since the firing rate $1/\bar{\Delta}^{(1)}$ is a fundamental quantity for investigating pulse coding abilities of a neuron model [4]. The DSN can exhibit various response characteristics depending on the wiring pattern $A^{(1)}$ (see for example [8]).

3 Synchronization in Pulse-Coupled Digital Spiking Neurons

3.1 Pseudo-Periodic $k : 1$ Synchronization

In this section, we present a one-way pulse-coupled system of two DSNs (ab. PC-DSN) as shown in Fig.4(a). The DSN2 has a synaptic coupling weight w from the DSN1. Fig.4(b) shows a typical time-domain waveform. The dynamics of the state $X^{(2)}$ and the output $Y^{(2)}$ of the DSN2 are described by

$$X^{(2)}(t + d) = \begin{cases} X^{(2)}(t) + 1 & \text{if } X^{(2)}(t) < N^{(2)} - 1 \text{ and } Y^{(1)}(t) = 0, \\ X^{(2)}(t) + w & \text{if } X^{(2)}(t) < N^{(2)} - 1 \text{ and } Y^{(1)}(t) = 1, \\ B^{(2)}(t + d) & \text{if } X^{(2)}(t) = N^{(2)} - 1, \end{cases} \quad (12)$$

$$Y^{(2)}(t) = \begin{cases} 1 & \text{if } X^{(2)}(t) = N^{(2)} - 1 \text{ and } t \in \{0, d, 2d, \dots\}, \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

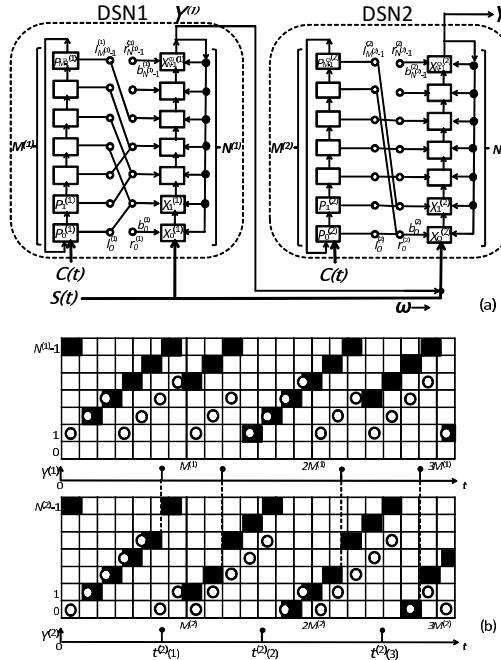


Fig. 4. (a) Pulse-coupled DSNs (ab. PC-DSN). (b) Time-domain waveform.

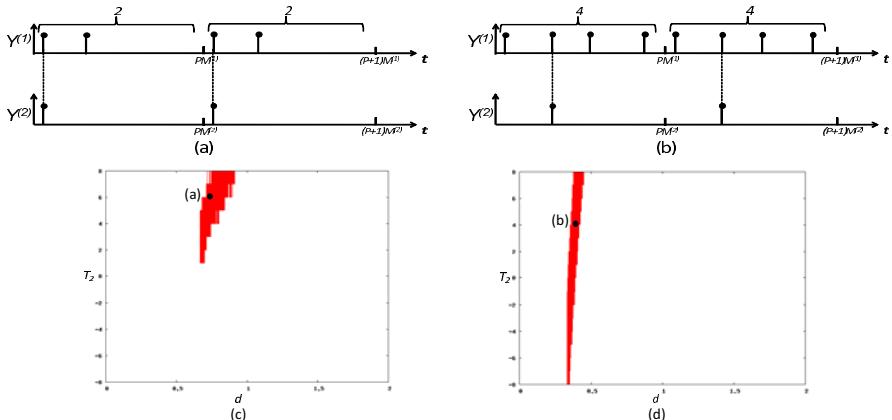


Fig. 5. (a) 2:1 synchronization. $T_2 = 6$, $d = 0.81$. (b) 4:1 synchronization. $T_2 = 4$, $d = 0.39$. (c) Theoretical parameter condition for 2:1 synchronization. $w = 5$, $N^{(2)} = 64$. (d) Theoretical parameter condition for 4:1 synchronization. $w = 10$, $N^{(2)} = 128$.

The DSN2 has the following two types of firings.

- Self-firing:** $Y^{(2)}(t) = 1$ and $Y^{(1)}(t) = 0$.
- Compulsory-firing:** $Y^{(2)}(t) = 1$ and $Y^{(1)}(t) = 1$.

In Fig 4(b), the DSN2 exhibits self-firings at $t = t^{(2)}(2), t^{(2)}(3)$ and a compulsory-firing at $t = t^{(2)}(1)$. Same as the DSN1, we denote the n -th firing moment (that includes both self-firings and compulsory-firings) of the DSN2 by $t^{(2)}(n)$ and define the following n -th spike phase $\theta^{(2)}(n)$.

$$\theta^{(2)}(n) = t^{(2)}(n) \pmod{M^{(2)}},$$

where $n = 1, 2, 3, \dots$. We then give the following definition.

Definition 2. The DSN2 is said to exhibit a $k : 1$ pseudo-periodic synchronization to the DSN1 if the following three statements are true.

- The DSN1 generates a pseudo-periodic spike-train $Y^{(1)}(t)$.
- Every firing of the DSN2 is compulsory-firing.
- The average ISI of the DSN2 is $\bar{\Delta}^{(2)} = k\bar{\Delta}^{(1)}$, $k \in \{1, 2, 3, \dots\}$.

For example, the DSN2s in Figs 5(a) and (b) exhibits $2 : 1$ and $4 : 1$ pseudo-periodic synchronizations, respectively, where the DSN2 is characterized by the following parameter values

$$M^{(1)} = N^{(1)} = M^{(2)} = 32, \quad A^{(2)}(m) = \begin{cases} m + 16 + T_2 & \text{for } 0 \leq m \leq 15, \\ m + T_2 & \text{for } 16 \leq m \leq 31, \\ m - 16 + T_2 & \text{for } 32 \leq m \leq 47, \\ m - 32 + T_2 & \text{for } 48 \leq m \leq 64. \end{cases} \quad (14)$$

We then provide the following theorem.

Theorem 1. The DSN2 exhibits a $k : 1$ pseudo-periodic synchronization to the DSN1 if the following parameter conditions are satisfied.

$$\begin{aligned} N^{(2)} - 1 - w &\leq A^{(2)}(i(j)) + \text{str} + (k-1)(w-1), \\ A^{(2)}(i(j)) + \text{str} + (k-1)(w-1) &\leq N^{(2)} - 2 - (\max(\mathbf{J}) - \min(\mathbf{J})), \\ \text{str} &= \begin{cases} \text{Int}\left(\frac{M^{(2)} - (j - \min(\mathbf{J}))d}{d}\right) - 1 & \text{if } M^{(2)} \pmod{d} = 0, \\ \text{Int}\left(\frac{M^{(2)} - (j - \min(\mathbf{J}))d}{d}\right) & \text{otherwise,} \end{cases} \quad (15) \\ \mathbf{J} &= \{j | \min(\Theta_r^{(1)}) \leq dj + \phi \leq \max(\Theta_r^{(1)}), \phi = \theta^{(1)} \pmod{d}, j = \text{Int}(\theta^{(1)}/d)\}, \\ \mathbf{I} &= \{i(j) | dj \leq i(j) \leq d(j+1), j \in \mathbf{J}\}, \end{aligned}$$

where the function $\text{Int}(x)$ gives the integer part of a real number x .

A proof will be given in a journal paper. Figs 5(c) and (d) shows the parameter conditions in Equation (15) of Theorem 1 for $k = 2$ and $k = 4$, respectively. By extensive numerical simulations, we confirmed that Theorem 1 is correct in the black parameter region in Figs 5(c) and (d). In addition to the pseudo-periodic synchronization, we also study non-periodic synchronization in the next subsection.

3.2 Non-periodic 1 : 1 Synchronization

The DSN1 is said to generate a non-periodic spike-train $Y^{(1)}(t)$ if $Y^{(1)}(t)$ is neither pseudo-periodic nor periodic. Then we give the following definition.

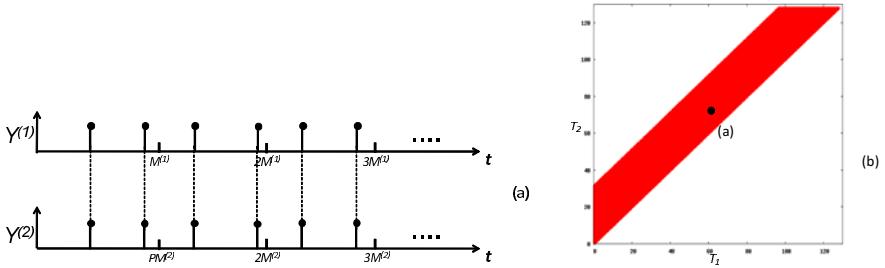


Fig. 6. 1:1 synchronization. The parameters of the PC-DSN are in equation. (16), $d = 0.99$ and $w = 32$. (a) 1:1 non-periodic synchronization. $T_1 = 63$, $T_2 = 72$. (b) Parameter condition.

Definition 3. The DSN2 is said to exhibit a 1 : 1 non-periodic synchronization to the DSN1 if the following statements are true.

- The DSN1 generates a non-periodic spike train $Y^{(1)}(t)$.
- Every firing of the DSN2 is compulsory-firing.
- Every firing moment of the DSN2 coincides with that of the DSN1, i.e., $t^{(1)}(n) = t^{(2)}(n)$ for all $n = 0, 1, 2, \dots$

For example, the DSN2 in Fig. 6(a) exhibits a 1 : 1 non-periodic synchronization, where the DSN2 is characterized by the following parameter values.

$$M^{(i)} = N^{(i)} = 128, \quad A^{(i)}(m) = \text{Int}\left(24.2 \sin\left(\frac{\pi}{64}m\right) + 128 - T_i\right). \quad (16)$$

Then we can give the following theorem.

Theorem 2. The DSN2 exhibits a 1 : 1 non-periodic synchronization or a 1 : 1 pseudo periodic synchronization to the DSN1 if the following parameter conditions are satisfied.

$$\begin{cases} 0 \leq (X^{(1)}(0) - N^{(1)}) - (X^{(2)}(0) - N^{(2)}) \leq w - 1, \\ 0 \leq (A^{(1)}(m) - N^{(1)}) - (A^{(2)}(m) - N^{(2)}) \leq w - 1. \end{cases} \quad m = 0, 1, 2, \dots. \quad (17)$$

A proof will be given in a journal paper. Fig. 6(b) shows the parameter conditions in equation (17) of Theorem 2. By extensive numerical simulations, we confirmed that Theorem 2 is correct in the black parameter region in Fig. 6(b).

4 Conclusion

We have presented the PC-DSN that can exhibit various $k : 1$ pseudo-periodic synchronizations and 1 : 1 non-periodic synchronizations. We have derived the theorems that give the sufficient parameter conditions for the synchronizations. Then we have validated the theorems by the numerical simulations. Since these synchronizations are typically observed in biological neurons, the result

of this paper will be a first step towards reproduction of neurons dynamics in a cell-automaton-based reconfigurable digital chip. Future problems include: (a) detailed analysis of bifurcation phenomena, (b) development of a methodology to reproduce given neural synchronizations phenomena, and (c) hardware implementation.

References

1. Izhikevich, E.M.: *Dynamical systems in neuroscience*. MIT Press, Cambridge (2006)
2. Izhikevich, E.M.: Simple model of spiking neurons. *IEEE Trans. Neural Networks* 14, 1569–1572 (2003)
3. Haken, H.: *Brain Dynamics*, 2nd edn. Springer, Heidelberg (2008)
4. Cariani, P.A., Delgutte, B.: Neural Correlates of the Pitch of Complex Tones. I. Pitch and Pitch Salience. *Journal of Neurophysiology* 76(3), 1698–1716 (1996)
5. Hishiki, T., Torikai, H.: Neural behaviors and nonlinear dynamics of a rotate-and-fire digital spiking neuron. In: Proc. IEEE-INNS/IJCNN, pp. 2921–2928 (2010)
6. Torikai, H., Hamanaka, H., Saito, T.: Reconfigurable spiking neuron and its pulse-coupled networks: basic characteristics and potential applications. *IEEE Trans. CAS-II* 53(8), 734–738 (2006)
7. Nakano, H., Saito, T.: Grouping synchronization in a pulse-coupled network of chaotic spiking oscillators. *IEEE Trans. Neural Networks* 15(5), 1018–1026 (2004)
8. Hashimoto, S., Torikai, H.: A novel hybrid spiking neuron: Bifurcations, Responses, and On-chip learning. *IEEE Trans. CAS-I* 57(8), 2168–2181 (2010)
9. Torikai, H., Funew, A., Saito, T.: Digital spiking neuron and its learning for approximation of various spike-trains. *Neural Networks* 21(2-3), 140–149 (2008)
10. Hashimoto, S., Torikai, H.: Bifurcation analysis of reconfigurable hybrid spiking neuron and its novel online learning algorithm. In: Proc. IEEE-INNS IJCNN, pp. 1134–1141 (2009)

Emergence of Highly Nonrandom Functional Synaptic Connectivity Through STDP

Hideyuki Kato¹ and Tohru Ikeguchi^{1,2}

¹ Graduate School of Science and Engineering, Saitama University,
255 Shimo-Ohkubo, Saitama city, Saitama 338-8570, Japan

² Saitama University, Brain Science Institute,
255 Shimo-Ohkubo, Saitama city, Saitama 338-8570, Japan
kato@nls.ics.saitama-u.ac.jp,
tohru@ics.saitama-u.ac.jp

Abstract. We investigated the network topology organized through spike-timing-dependent plasticity (STDP) using pair- and triad-connectivity patterns, considering difference of excitatory and inhibitory neurons.

As a result, we found that inhibitory synaptic strength affects statistical properties of the network topology organized through STDP more strongly than the bias of the external input rate for the excitatory neurons. In addition, we also found that STDP leads highly nonrandom structure to the neural network. Our analysis from a viewpoint of connectivity pattern transitions reveals that STDP does not uniformly strengthen and depress excitatory synapses in neural networks. Further, we also found that the significance of triad-connectivity patterns after the learning results from the fact that the probability of triad-connectivity-pattern transitions is much higher than that of combinations of pair-connectivity-pattern transitions.

1 Introduction

It is generally believed that synaptic plasticity is essential to realize higher brain functions of memory, learning and development. Using such a function, neurons in the brain organize a large network with synaptic connections to effectively process a huge amount of information.

In the last decade, it has been clarified that the synaptic plasticity depends not only on firing frequencies of neurons but also on precise relative timing between pre- and postsynaptic neurons, which is spike-timing-dependent plasticity (STDP) [1, 2, 3]. Triggered by such experimental observations, STDP has been extensively investigated both experimentally and theoretically. Recent studies in the field of neuroscience show that STDP leads specific structures to a recurrent neural network, for example, wireless clusters [4, 5], feedforward structures [6, 7], and complex network structures [8]. Such structures emerge in the recurrent network under the conditions that neurons are in the limit cycle oscillation [5, 6, 7]. In addition, the network assumed in Refs. [6, 7] included no inhibitory neurons.

Then, in this paper, we analyzed local connectivity patterns [9] composed of two or three neurons in recurrent networks with STDP including both excitatory and inhibitory neurons. In addition, when we analyzed the local connectivity patterns, we

took the difference of excitatory and inhibitory neurons into account. To examine the network topology under various experimental conditions, we exhaustively investigated bias values of external input rate for excitatory neurons and inhibitory synaptic strength because we can realize not only excitable but also limit cycle neuronal states by varying the bias of the external input rate and inhibitory synaptic strength.

2 Materials and Methods

We constructed a neural network consisted of 1,000 neurons, which included both excitatory and inhibitory neurons. The ratio of excitatory to inhibitory neurons in the network was 4 to 1. We used a simple neuron model proposed by Izhikevich [10]. The excitatory neurons were a regular spiking type, whereas the inhibitory ones were a fast spiking type. Parameters of these neurons were same as in Ref. [10]. These neurons in the network interacted through synapses with 10% connection probability, which indicates that synaptic connections of each neuron targeted 100 postsynaptic neurons. Connectivity between neurons in the initial condition was random, where targets of inhibitory connections were only excitatory neurons for stable network dynamics. Excitatory connections had 1 to 10 ms conduction delays, whereas 1 ms were required to conduct a spike on all inhibitory connections.

In our network model, excitatory synapses were plastic but inhibitory synapses were fixed. The excitatory synapses were modified depending on pre- and postsynaptic neuronal activities. The synapses changed by Δg_{ij} at every firing event, and synaptic modification is defined as follows:

$$\Delta g_{ij} = \begin{cases} A_+ \cdot \exp(-|\Delta t|/\tau) & \Delta t > 0, \\ -A_- \cdot \exp(-|\Delta t|/\tau) & \Delta t \leq 0, \end{cases} \quad (1)$$

which is well known as the additive rule [11, 12]. In Eq. (1), Δt is the relative spike timing, namely $\Delta t = t_i - t_j$ where t_j is the firing time of the neuron j . The parameters A_+ ($= 0.1$) and A_- ($= 0.12$) are the learning rates, and τ ($= 20$ ms) is the time constant determining the exponential decay. Other parameter settings were same as in Ref. [13]. Introducing experimental results [14], we applied a Poissonian spike sequence of 10 Hz to individual neurons as external stimuli. In addition, we biased the external input rate for the excitatory neurons by v Hz. The amplitude of each spike in the sequence was set to 20, which corresponds to the suprathreshold input when a neuron is at the resting state.

We extracted connectivity patterns (CPs) from the network, and enumerated 5 pair-CPs (PCPs) and 49 triad-CPs (TCPs) in our network model (Fig. 1) because we avoided connecting between any pairs of inhibitory neurons. When we extracted the CPs, we neglected only excitatory connections that satisfies the condition that $g_{ij} < g_\theta$. In the following analysis, we fixed $g_\theta = 4$. Our results were not affected by the value of g_θ . To quantify the statistical significance of CPs, we used the Z-score of CPs defined by

$$Z_i = \frac{N_i - \langle N_i^{\text{rand}} \rangle}{\text{SD}(N_i^{\text{rand}})}, \quad (2)$$

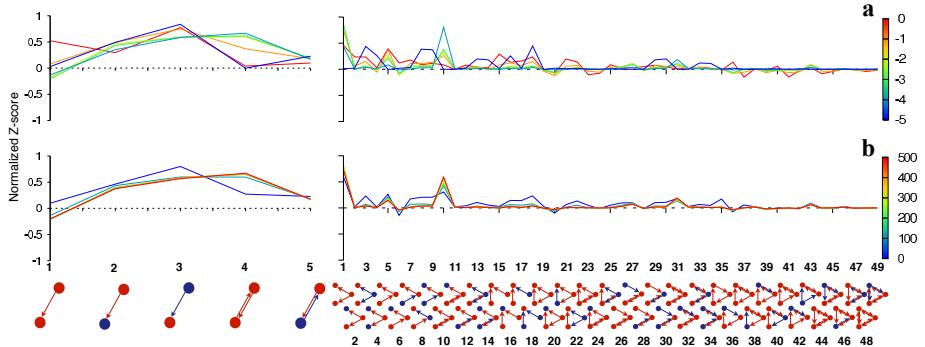


Fig. 1. Highly nonrandom functional synaptic connectivity in the neural network organized through STDP. Horizontal axes are pair- (PCPs) or triad-connectivity patterns (TCPs) that are subgraphs composed of two or three neurons. A circle and an arrow represent a neuron and a synaptic connection, respectively. Red and blue are excitatory and inhibitory neurons and synaptic connections. Avoiding connections between inhibitory neurons, we enumerated 5 PCPs and 49 TCPs from our neural network model. (a) Dependency of the functional connectivity of the neural network organized through STDP on the inhibitory synaptic strength. Colors corresponds to the inhibitory synaptic strength. (b) Dependency of the functional connectivity of the neural network organized through STDP on the bias of the external input rate for the excitatory neurons. All results are for $g_\theta = 4$, but this value dose not affect our results except for the case of $g_\theta = 0$.

where i is the ID of CP i , N_i is the number of CP i in the neural network and N_i^{rand} is the number of CP i in the corresponding randomized networks [9]. The descriptions of $\langle \cdot \rangle$ and $\text{SD}(\cdot)$ represent the average and the standard deviation. We generated 50 randomized neural networks from the neural network. In the randomized neural networks, both indegree and outdegree of excitatory and inhibitory connections were preserved. For comparison to the Z-score under different conditions, we normalized the Z-score in the same way as Ref. [9]:

$$\tilde{Z}_i = \frac{Z_i}{\sqrt{\sum_j Z_j^2}}. \quad (3)$$

If STDP uniformly strengthens and depresses the excitatory synapses in the neural network, no CPs are significant because the network in the initial condition has random connectivity. If we observe high Z-scores, it indicates that the synaptic modification is not based on the above-mentioned assumption. To investigate it, we introduced CP transitions (CPTs) (Fig. 2(a)). We defined a remaining probability p that is the number of the excitatory connections after the learning divided by that before the learning. If learning of the neural network satisfies the above assumption, the probability of the CPT from the ID 37 to 10 (Fig. 2(a)) should be theoretically written as $p(1-p)^3$. In contrast, we can compute actual probability of the TCPT as $n_{37 \rightarrow 10}^{(3)} / n_{37}^{(3)}$ where $n_{37 \rightarrow 10}^{(3)}$ is the number of the TCPT from the ID 37 to 10, and $n_{37}^{(3)}$ is the number of the TCP of the ID 37 in the initial condition. Then, we quantified the transitivity of the TCP by

$$T_{37 \rightarrow 10}^{(3)} = \frac{n_{37 \rightarrow 10}^{(3)}}{p(1-p)^3}. \quad (4)$$

We call this statistics a transitivity index. The transitivity index is close to unity when the synaptic modification is based on the above-mentioned assumption, whereas the index is high if the actual probability of the CPT is higher than theoretical one.

As is the case with the CPT, if the above mentioned assumption is held, each TCP can be represented as a combination of the PCPs because the TCP is composed of the PCPs (Fig. 3(a)). Then, a TCP transition (TCPT) can be also calculated as a combination of the PCP transitions (PCPTs) (Fig. 3(a)). To further investigate of the CPTs, we defined the likelihood of a TCPT for PCPTs. The likelihood of the TCPT for the TCPTs as shown in Fig. 3(a) was defined as the ratio of the transitivity index of the TCPT to the multiplied transitivity indices of the PCPTs:

$$L_{37 \rightarrow 10}^{(3)} = \frac{T_{37 \rightarrow 10}^{(3)}}{T_{4 \rightarrow 4}^{(2)} \times T_{1 \rightarrow 1}^{(2)} \times T_{1 \rightarrow \text{broken}}^{(2)}}. \quad (5)$$

The likelihood of the TCPT for the PCPTs is close to unity when the synaptic modification is based on the above assumption, whereas the likelihood is high when the TCPT more frequently occurs than the combination of the PCPTs. The broken in Eq. (5) and Fig. 3 means that although 2 or 3 neurons construct the CPs as shown in Fig. 1 in the initial condition, these neurons do not construct the CPs by the synaptic depression after the learning.

3 Results

3.1 Relativity to Random Networks

The network has much more ID 4 in the PCPs than the randomized networks when $-4 \leq g_{\text{inh}} \leq -1$ (Fig. 1(a), left), which indicates that bidirectional excitatory connections are strongly influenced by inhibitory synaptic strength. The form of the normalized Z-scores of the TCPs is significantly different depending on the inhibitory synaptic strength (Fig. 1(a), right). Then, the inhibitory synaptic strength strongly affects statistical properties of the neural network organized through STDP. In particular, almost all the connectivity patterns including the bidirectional excitatory connections are as few as the randomized networks in the case of $g_{\text{inh}} = -5$. In addition, when $g_{\text{inh}} = 0$, some connectivity patterns including bidirectional excitatory connections are fewer than the randomized networks.

Except for the case of $v = 0$, the forms of the normalized Z-scores of both PCPs and TCPs duplicates, although v is varied (Fig. 1(b)). This fact means that the variation of the bias of the external input rate for the excitatory neurons does not so much affect statistical properties of the organized network, compared with the variation of the inhibitory synaptic strength. Anyway, the neural network organized through STDP has highly nonrandom structure (Fig. 1(a), (b)).

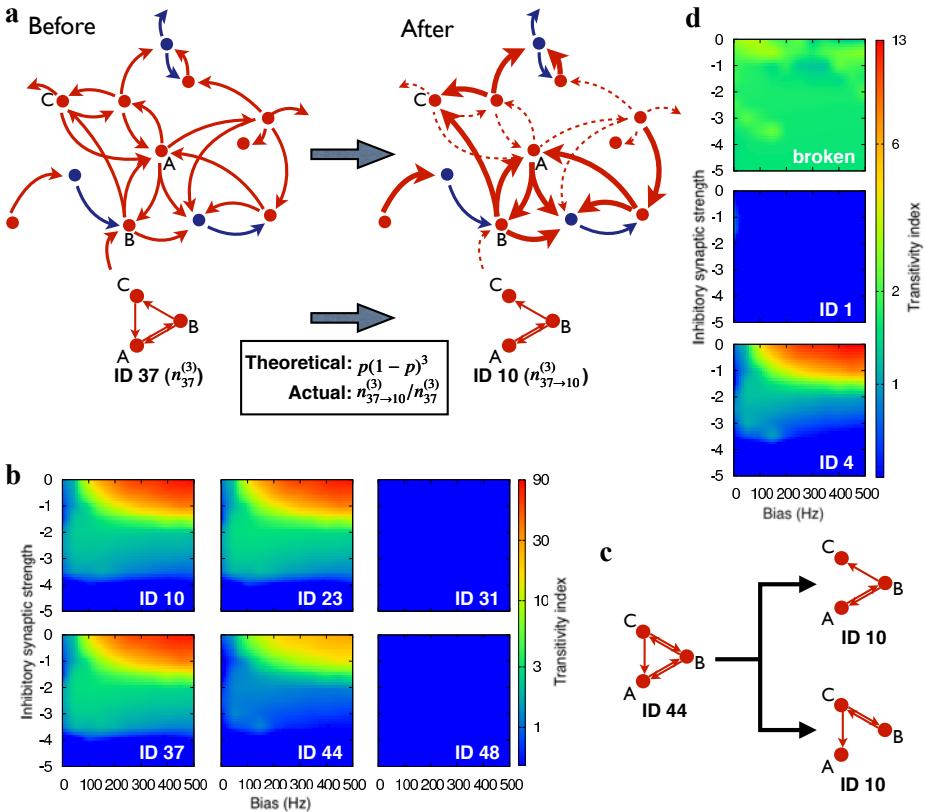


Fig. 2. Nonuniform synaptic potentiation and depression in the neural network. (a) A schematic diagram of our analysis from a viewpoint of connectivity-pattern transitions (CPTs). Dashed arrows in the neural network after the learning represent depressed synapses that do not satisfy $g_{ij} > g_\theta$, and they were neglected when we extracted CPTs. Then, we can regard that the TCP composed of neurons A, B and C before learning is different from after the learning because the TCP before learning has the ID 37 but that after the learning has the ID 10. The probability of this triad-CPT (TCPT) can be theoretically computed from the remaining probability p that is the number of excitatory connections satisfying $g_{ij} > g_\theta$ after the learning divided by that before learning as $p(1 - p)^4$. In contrast, we can calculate the actual probability of the TCPT as $n_{37 \rightarrow 10}^{(3)} / n_{37}^{(3)}$ where $n_{37 \rightarrow 10}^{(3)}$ is the number of the TCPT from the ID 37 to 10, and $n_{37}^{(3)}$ is the number of the TCP of the ID 37 in the initial condition. (b) The transitivity indices of the TCPT from six TCPTs to the TCP with the ID 10 in $\nu - g_{inh}$ space. Colors correspond to the value defined in Eq. (4) when we varied the parameters of the bias of the external input rate for the excitatory neurons, ν Hz, and the inhibitory synaptic strength, g_{inh} . (c) Two ways of the TCPTs from the TCP with the ID 44 to that with the ID 10. (d) The transitivity indices of the PCPTs from the PCP with the ID 4 to three PCPs (broken, IDs 1 and 4).

3.2 Connectivity-Pattern Transitions

Here, we focus on the ID 10 in the TCPs whose Z-score is relatively large as shown in Fig. 1. Six TCPs in the initial condition (IDs 10, 23, 31, 37, 44 and 48) can be translated into the ID 10 after the learning (Fig. 2(b)). Four TCPs (IDs 10, 23, 37 and 44) of the six TCPs are easy to translate into the TCP with the ID 10 when the external input rate for the excitatory neurons is strongly biased and the inhibitory synaptic strength is weak (Fig. 2(b)). However, the transitivity index of the TCPT from the ID 44 to 10 takes smaller value than that of the other three TCPTs. This is because there are two ways for the TCP with the ID 44 to be the TCP with the ID 10 (Fig. 2(c)). In the upper way of the TCPT, the PCPT from the ID 4 to 1 between neurons B and C is essential, but it is rare for the PCPT to occur (Fig. 2(d), middle). In contrast to the upper transition, the lower transition includes no rare PCPTs (Fig. 2(c)). In addition, the transitivity index of the PCPT is the sum of the transitivity indices of the two ways. Then, the transitivity index of the TCPT from the ID 44 to 10 is totally smaller than that of the other three TCPTs.

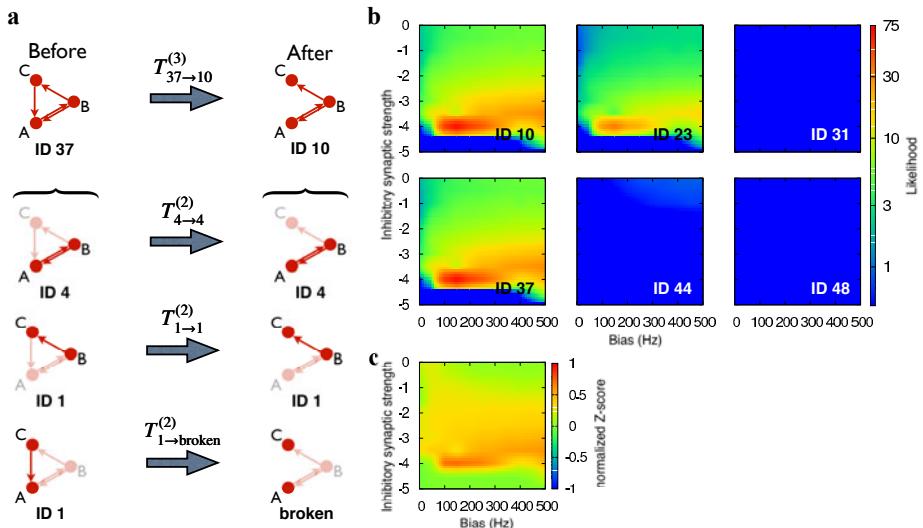


Fig. 3. Triad-connectivity-pattern transitions (TCPTs) are not simply combinations of pair-connectivity-pattern transitions (PCPTs). (a) A schematic diagram of the TCPTs as combinations of the PCPTs. A TCP is composed of three PCPs (dark-red PCPs), so that a TCPT is regarded as combinations of three PCPTs. We can compute the transitivity index of all PCPTs and TCPTs with the same way written in Eq. (4), and then defined the ratio of the transitivity index of the TCPT to the multiplied transitivity indices of the PCPTs as the likelihood of the TCPT for the PCPTs. (b) The likelihood of the TCPT from six TCPs to the TCP with the ID 10 in v - g_{inh} space. Colors correspond to the value defined in Eq. (5) when we varied the parameters of the bias of the external input rate for the excitatory neurons, v , and the inhibitory synaptic strength, g_{inh} . (c) The normalized Z-score of the TCP with the ID 10 after the learning when we varied the parameters of the bias of the external input rate for the excitatory neurons, v , and the inhibitory synaptic strength, g_{inh} . Colors correspond to the value of the normalized Z-score.

3.3 Triad-Connectivity-Pattern Transitions Are Not Simply Combinations of Pair-Connectivity-Pattern Transitions

Different from the transitivity indices of TCPTs, the likelihood of three TCPTs (IDs 10, 23 and 37) is high when the inhibitory synaptic strength is around -4 (Fig. 3(b)). This indicates that the probability of the TCPTs is significantly higher than that of the combinations of the PCPTs. Interestingly, the tendency of the normalized Z-score of the TCP with the ID 10 after the learning in $v-w_{inh}$ space (Fig. 3(c)) is similar to that of the likelihood of the three TCPTs (IDs 10, 23 and 37) (Fig. 3(b)). Namely, both the normalized Z-score and the likelihood is the highest when the inhibitory synaptic strength is about -4 and the bias of the external input for the excitatory neurons is between 100 and 200 Hz. The Z-score of the TCP correlates with the likelihood of the three TCPTs (Fig. 3(b), (c)), while the Z-score does not correlate with the transitivity indices of other three TCPTs (Figs. 2(b) and 3(c)). Then, the significance of the TCPs after the learning arises from the fact that the probability of the TCPTs is much higher than that of the combinations of the PCPTs. Our results imply that an analysis of the TCPs is as inevitable as that of the PCPs. In other words, the analysis of only PCPs is not enough to understand the connectivity of neural networks because our results show that the TCPs in the neural network is not simply combinations of the PCPs.

4 Conclusions

In this paper, we statistically analyzed the network topology constructed through STDP. Our neural networks included both excitatory and inhibitory neurons. In the analysis, we considered different roles of the excitatory and the inhibitory neurons. In addition, to investigate the organized network topology under several experimental conditions, we widely varied the bias of the external input rate for the excitatory neurons and the inhibitory synaptic strength as parameters.

As a result, we found that inhibitory synaptic strength affects statistical properties of the network topology organized through STDP more strongly than the bias of the external input rate for the excitatory neurons. Additionally, the neural network has highly nonrandom structure that is generated by STDP.

Our analysis from a viewpoint of the connectivity pattern transitions revealed that STDP does not uniformly strengthen and depress excitatory synapses in a neural network. Further, we also found that the significance of triad-connectivity patterns after the learning arises from the fact that the probability of triad-connectivity-pattern transitions is much higher than that of combinations of pair-connectivity-pattern transitions.

Acknowledgment. The present authors thank Dr. N. Masuda of Univ. of Tokyo for fruitful discussions. The research of H.K. is partially supported by Research Fellowships of JSPS (No. 22·8144). The research of T.I. is partially supported by a Grant-in-Aid for Scientific Research (C) (No. 20560352) from JSPS.

References

- [1] Markram, H., Lüboke, J., Frotscher, M., Sakmann, B.: Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* 275, 213–215 (1997)
- [2] Bi, G.Q., Poo, M.M.: Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength and postsynaptic cell type. *The Journal of Neuroscience* 18(24), 10464–10472 (1998)
- [3] Abbott, L.F., Nelson, S.B.: Synaptic plasticity: taming the beast. *Nature Neuroscience Supplement* 3, 1178–1183 (2000)
- [4] Levy, N., Meilijson, D.H.I., Ruppin, E.: Distributed synchrony in a cell assembly of spiking neurons. *Neural Networks* 14, 815–824 (2001)
- [5] Câteau, H., Kitano, K., Fukai, T.: Interplay between a phase response curve and spike-timing-dependent plasticity leading to wireless clustering. *Physical Review E* 77, 051909 (2008)
- [6] Masuda, N., Kori, H.: Formation of feedforward networks and frequency synchrony by spike-timing-dependent plasticity. *The Journal of Computational Neuroscience* 22, 327–345 (2007)
- [7] Takahashi, Y.K., Kori, H., Masuda, N.: Self-organization of feed-forward structure and entrainment in excitatory neural networks with spike-timing-dependent plasticity. *Physical Review E* 79, 051904 (2009)
- [8] Kato, H., Ikeguchi, T., Aihara, K.: Structural analysis on stdp neural networks using complex network theory. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009. LNCS, vol. 5768, pp. 306–314. Springer, Heidelberg (2009)
- [9] Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenststat, I., Sheffer, M., Alon, U.: Superfamilies of evolved and designed networks. *Science* 303, 1538–1542 (2004)
- [10] Izhikevich, E.M.: Simple model of spiking neurons. *IEEE Transactions on Neural Networks* 14(6), 1569–1572 (2003)
- [11] Song, S., Miller, K.D., Abbott, L.F.: Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience* 3(9), 919–926 (2000)
- [12] Rubin, J., Lee, D.D., Sompolinsky, H.: Equilibrium properties of temporally asymmetric Hebbian plasticity. *Physical Review Letters* 86(2), 364–367 (2001)
- [13] Kato, H., Ikeguchi, T.: Analysis on complex structure of neural networks with STDP. *The IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (Japanese Edition)* J92-A (2) (2009)
- [14] Softky, W.R., Koch, C.: The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *The Journal of Neuroscience* 13(1), 334–350 (1993)

Modulation of Corticofugal Signals by Synaptic Changes in Bat's Auditory System

Yoshihiro Nagase^{1,*} and Yoshiki Kashimori^{1,2}

¹ Dept. of Human Media Systems, Univ. of Electro-Communications, Chofu, Tokyo,
182-8585 Japan

² Dept. of Science and Engineering, Univ. of Electro-Communications, Chofu, Tokoyo,
182-8585 Japan
recruitcf@gmail.com, kashi@pc.uec.ac.jp

Abstract. Most species of bats making echolocation use Doppler-shifted frequency of ultrasonic echo pulse to measure the velocity of target. The neural circuits for detecting the target velocity are specialized for fine-frequency analysis of the second harmonic constant frequency (CF2) component of Doppler-shifted echoes. To perform the fine-frequency analysis, the feedback signals from cortex to subcortical and peripheral areas are needed. The feedback signals are known to modulate the tuning property of subcortical neurons. However, it is not yet clear the neural mechanism for the modulation of the tuning property. We present here a neural model for detecting Doppler-shifted frequency of echo sound reflecting a target. We show that the model reproduce qualitatively the experimental results on the modulation of tuning shifts of subcortical neurons. We also clarify the neural mechanism by which the tuning property is changed depending on the feedback connections between cortex and subcortex neurons.

Keywords: auditory system, top-down signal, tuning property, synaptic plasticity, neural model.

1 Introduction

Animals usually receive complex sensory signals in external world. To perform sensory perception, they must select actively the sensory information relevant to their behavior. To extract such information from complex signals, the feedback signals from cortex to subcortical and peripheral regions are needed. However, it is not yet clear how the feedback signals contribute to the selection of sensory information. The behavioral characteristics of the sound selection determine that the underlying neuronal substrate must be a back projection system and is able to implement selective sound processing based on auditory information already registered in the higher auditory processing levels. To address the issue of sound selection, we study echolocation of mustached bats, because the physiological properties of neuronal activities modulated by the feedback signals have been actively investigated [1-4], and the functions of the cortical areas have been well characterized[5].

* Corresponding author.

Mustached bats emit ultrasonic pulses and listen to returning echoes for orientation and hunting flying insects. The bats analyze the correlation between the emitted pulses and their echoes and extract the detailed information about flying insects based on the analysis. This behavior is called echolocation. The neuronal circuits underlying echolocation detect the velocity of target with accuracy of 1 cm/sec and the distance of target with accuracy of 1 mm. To extract the various information about flying insects, mustached bats emit complex biosonar that consists of a long-constant frequency (CF) component followed by a short frequency-modulated (FM) component. Each pulse contains four harmonics and so eight components represented by (CF1, CF2, CF3, CF4, and FM1, FM2, FM3, FM4)[6]. The information of target distance and velocity are processed separately along the different pathways in the brain by using four FM components and four CF components, respectively [5].

In natural situation, large natural objects in environment, like bushes or trees, produce complex stochastic echoes, which can be characterized by the echo roughness. The echo signal reflecting from a target insect is embedded in the complex signal. Even in such a environment, bats can detect accurately the detailed information of flying insect. To extract the information about insects, the feedback signals from cortex to subcortical areas are needed.

To investigate the role of feedback signals in extracting the information about insect, we study the neural pathway for detecting velocity of target, which consists of cochlea, inferior colliculus (IC), and Doppler-shifted constant frequency (DSCF) area. The cochlea is remarkably specialized for fine-frequency analysis of the second harmonic CF component (CF2) of Doppler-shifted echoes. The information about echo CF2 (ECF2) is transmitted to IC, and the relative velocity of target insect is detected in DSCF area by analyzing the Doppler-shifted frequency. Xia and Suga[4] have shown on intriguing property of feedback signals that the electric stimulation of DSCF neurons evokes the best frequency (BF) shifts of IC neurons away from the BF of the stimulated DSCF neuron (centrifugal BF shift) and bucuculine (an antagonist of inhibitory GABA receptors) applied to the stimulation site changes the centrifugal BF shifts into the BF shifts towards the BF of stimulated DSCF neurons (centripetal BF shift). Although these BF shifts are generated by the feedback signals from DSCF neurons to IC neurons, it is not yet clear how the feedback signals determine the direction of BF shift.

In the present study, we propose a neural network model for detecting Doppler-shifted frequency of sound echoes. Using the model, we show the neural mechanism by which the centripetal and centrifugal BF shifts are elicited.

2 Model

The network model for detecting the Doppler-shifted frequency consists of cochlea, inferior colliculus (IC), and Doppler-shifted constant frequency (DSCF) area, each of which is a linear array of frequency-tuned neurons shown in Fig. 1. The neurons in the three layers are tuned in to specific echo frequency ranging from 60.0 to 63.0 kHz, which corresponds to the frequency range of the second harmonics CF2. The bat uses the Doppler-shifted frequency of echo sound ECF2 to detect the relative velocity of target.

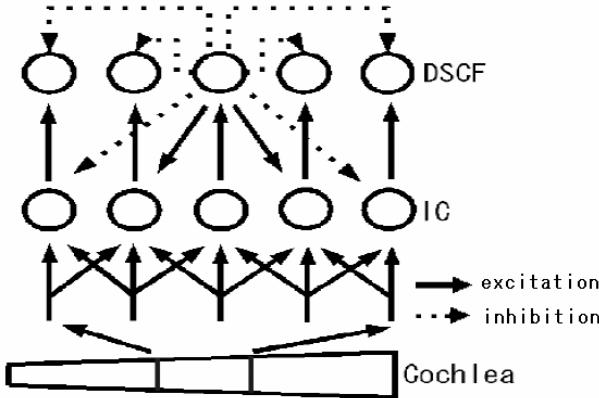


Fig. 1. Neural model for detecting Doppler-shifted frequency of echo sound. The excitatory synaptic connections are solid lines and the inhibitory synaptic connections are denoted by dotted lines.

2.1 Model of Cochlea

The function of cochlea is to decompose the echo sound into the intensity of frequency components. The cochlea neurons are tuned in to specific frequency. Because of broad tuning ability of cochlea, the firing rate of i th cochlea neuron is described by

$$X_i^{ch} = X_0^{ch} e^{-\frac{(i-i_0)^2}{\sigma_{ch}^2}}, \quad (1)$$

where i_0 is the location of cochlea neuron that maximally responds to echo sound ECF2, and X_0^{ch} is the maximum firing rate.

2.2 Model of IC

As shown in Fig. 1, the IC neurons are tonotopically connected with cochlea neurons, and receive the outputs of cochlea neurons. They also receive the feedback signals from DSCF neurons. The model of a single DSCF neuron was based on the leaky integrate-and-fire model.

The membrane potential of i th IC neuron, V_i^{IC} , is determined by

$$\tau_{IC} \frac{dV_i^{IC}}{dt} = -V_i^{IC} + \sum_{j=1}^N w_{ij}^{FF} X_j^{ch} + \sum_{k=1}^N w_{ik}^{FB} X_k^{DSCF}(t), \quad (2)$$

$$w_{ij}^{FF} = w_0 e^{-\frac{(j-i)^2}{\sigma_{IC}^2}}, \quad (3)$$

$$w_{ik}^{FB} = w_{ik}^{exc} e^{-\frac{(k-i)^2}{\sigma_a^2}} - w_{ik}^{inh} e^{-\frac{(k-i)^2}{\sigma_b^2}}, \quad (4)$$

$$X_j^{DSCF}(t) = \sum_{m=0}^M \frac{(t - t_{mj})}{\tau_s} e^{-\frac{t-t_{mj}}{\tau_s}}, \quad (5)$$

where τ_{lc} is the time constant of V_i^{IC} . w_{ij}^{FF} is the weight of synaptic connection from jth cochlea neuron to ith IC neuron, and w_{ik}^{FB} is the synaptic weight of feedback connection from kth DSCF neuron to ith IC neuron. The IC neuron integrates the outputs of cochlea neurons with ON center receptive field described in Eq. (3), and receives the feedback signals from DSCF neurons with ON center-OFF surrounding connections given by Eq. (4). The IC neuron generates an action potential whenever V_i^{IC} reaches the threshold value V_{th}^{IC} , and then the potential is reset to V_{rest} at 1 msec after the onset of the action potential. The refractory period was set at 3 msec. $X_j^{DSCF}(t)$ is the postsynaptic potential (PSP) evoked by spikes of jth DSCF neuron, which is described by sum of α -function. t_{mj} is the latest firing time of jth DSCF neuron, M is the number of spikes generated by jth DSCF neuron until the latest firing time.

2.3 Model of DSCF

The network model of DSCF is constructed with a linear array of frequency-tuned neurons, each of which is tonotopically connected with IC neurons, as shown in Fig. 1. The DSCF neuron feeds the outputs back to IC neurons with ON center-OFF surrounding connections given by Eq. (4). The DSCF neurons are mutually inhibited. The function of DSCF neurons is to decode the Doppler-shifted frequency of ECF2 on the frequency map of DSCF area in order to detect the relative velocity of target [7][8].

The DSCF neuron was modeled as the leaky integrate-and-fire model. The membrane potential of ith DSCF neuron, V_i^{DSCF} , is determined by

$$\tau_{DSCF} \frac{dV_i^{DSCF}}{dt} = -V_i^{DSCF} + \alpha X_i^{IC}(t) + \sum_{k=1}^N w_{ik}^{DSCF} X_k^{DSCF}(t), \quad (6)$$

where τ_{DSCF} is the time constant of V_i^{DSCF} . $X_i^{IC}(t)$ and $X_k^{DSCF}(t)$ are the output of ith IC neuron and that of kth DSCF neuron, respectively, which are given by the sum of α -functions similar to that given by Eq. (5). w_{ik}^{DSCF} is the weight of synapse from kth DSCF neuron to ith one. The DSCF neuron fires with the rule of leaky integrate-and-fire model similar to the IC neuron.

2.4 Synaptic Change of Feedback Connections

The synaptic weights of the feedback connections from DSCF to IC are changed by deviations from the initial values of excitatory and inhibitory synaptic weights, given by

$$w_{ij}^{exc}(t) = (w_0^E + w_{ij}^E(t))e^{-\frac{(i-j)^2}{\sigma_e^2}}, \quad (7a)$$

$$w_{ij}^{inh}(t) = (w_0^I + w_{ij}^I(t))e^{-\frac{(i-j)^2}{\sigma_i^2}}, \quad (7b)$$

where w_0^E and w_0^I are the initial values of the excitatory and inhibitory synapses, and $w_{ij}^E(t)$ and $w_{ij}^I(t)$ are the deviations from w_0^E and w_0^I , respectively.

The deviations, $w_{ij}^E(t)$ and $w_{ij}^I(t)$ are determined by

$$\tau_E \frac{dw_{ij}^E(t)}{dt} = -w_{ij}^E(t) + \lambda_E X_i^{IC}(t) X_j^{DSCF}(t), \quad (8a)$$

$$\tau_I \frac{dw_{ij}^I(t)}{dt} = -w_{ij}^I(t) + \lambda_I X_i^{IC}(t) X_j^{DSCF}(t), \quad (8b)$$

where τ_X and λ_X ($X=E, I$) are the time constant and learning rate of excitatory (E) and inhibitory (I) synapses, respectively.

3 Results

3.1 BF Shifts of IC Neurons Caused by Feedback Signals

Figure 2a shows the change in the tuning property of IC neurons in the case where tone stimulus is delivered and electric stimulus (ES) is applied to DSCF neuron tuned to 61.1 kHz. Before the ES, the IC neurons maximally respond to 60.6 kHz (vertical dashed line). When DSCF neuron tuned to 61.1 kHz is electrically stimulated, the BF of IC neuron is shifted from 60.6 kHz to 60.5 kHz. The ES results in the BF shift of the IC neurons away from the BF of the stimulated DSCF neuron, that is, centrifugal BF shift.

Activation of DSCF neuron with electric stimulus augments the auditory responses of matched IC neurons and sharpens their frequency-tuning without shifting their BFs, whereas it suppresses the auditory responses of unmatched IC neurons at their BFs. That is, it provides a compressed representation of Doppler-shifted frequency.

The centrifugal BF shift may play a critical role in detecting the direction of Doppler-shifted frequency elicited by a moving target. When echo stimulus reflecting from a moving target is applied to the auditory system, it evokes the activities of IC and DSCF neurons moving on their frequency maps, depending on the Doppler-shifted frequency of the target movement. During the movement of these activities, centrifugal BF shift evokes a compressed spread of activity of IC neurons in response to Doppler-shifted echo sound, because it augments the activity of matched IC neuron, and suppresses the activity of unmatched IC neurons. This leads to an accurate representation of moving direction of Doppler-shifted frequency.

Figure 2b shows the response properties of IC neurons when the antagonist of GABA, bicuculline (BMI), was applied to nearby the DSCF neuron tuned to 60.9 kHz before the ES. The BF of IC neurons shifts from 60.6 kHz to 60.8 kHz. The inhibition of GABA leads to the BF shift of the IC neuron towards the BF of the stimulated DSCF neuron, that is, centripetal BF shift.

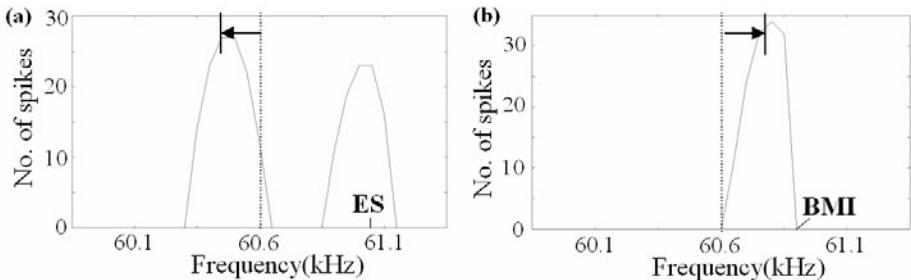


Fig. 2. Tuning properties of IC neurons. Centrifugal (a) and centripetal (b) BF shift. The dashed vertical lines indicate the center frequency of firing pattern in control. The arrows indicate BF shift caused by electric stimulation (ES) and application of bicuculline (BMI), respectively. ES used was 20 mV. BMI suppressed the lateral inhibition of DSCF neuron tuned to 60.9 kHz and of its neighbors within DSCF area.

3.2 Neural Mechanism for Determining Directions of BF Shifts

Figure 3 shows the temporal variations of the weights of the top-down connection from jth DSCF neuron to ith IC neuron. The excitatory and inhibitory synaptic weights are both gradually increased during the presence of the stimulus, and then remain nearly constant after the learning.

Figure 4a illustrate the changes in the synaptic potentials of the top-down connections from the DSCF neuron tuned to 61.6kHz to IC neurons, that is, the receptive field of the DSCF neuron, in the case of electric stimulation of a DSCF neuron. The ES makes the peak position of the synaptic potential shift away from the position of ES, resulting in the BF away from the BF of the stimulated DSCF neuron, that is, centrifugal BF shift. In contrast to the centrifugal BF shift, the application of bicuculline, an antagonist of GABA receptors, modulates the shape of the receptive field, as shown in Fig. 4b. The injection of bicuculline makes the peak position shift towards the injection site of bicuculline, thereby leading to the BF shift towards the BF of the stimulated DSCF neuron.

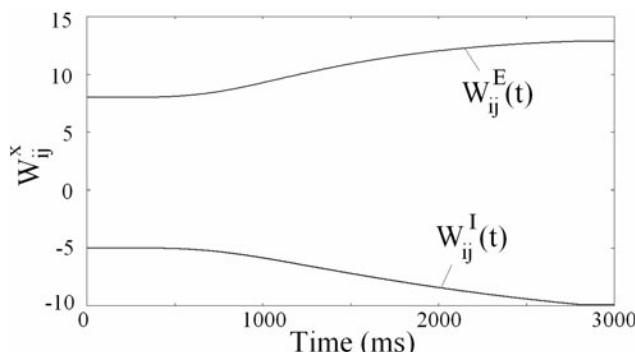


Fig. 3. Temporal variations of synaptic weights of top-down connection from ith DSCF neuron ($i=15$) to jth IC neuron ($j=15$)

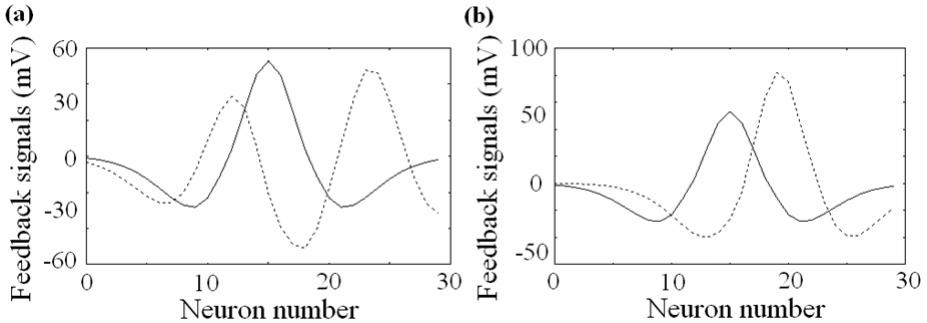


Fig. 4. The changes in synaptic potentials of top-down connections from a DSCF neuron to IC neurons. The solid lines indicate the synaptic potentials in control, and the dashed lines indicate those under ES (a) and application of bicuculline (b), respectively.

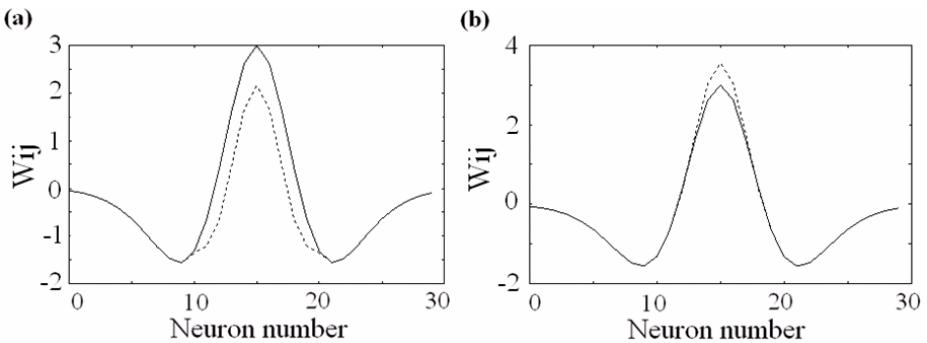


Fig. 5. The changes in synaptic weights of top-down connections from a DSCF neuron to IC neurons. The solid lines indicate the synaptic weights in control, and the dashed lines indicate those caused by a strong sound stimulus (a) and a weak stimulus (b), respectively.

3.3 Adaptive Changes of Synaptic Weights Depending on the Stimulus Intensity

Figure 5 shows the receptive fields of a DSCF neuron in the application of a strong and weak tone stimulus. For strong stimulus, the synaptic weights of top-down connections decrease to eliminate the incorrect frequency components of the stimulus, as shown in Fig. 5a. The weak stimulus, on the other hand, makes the synaptic weights increased to enhance the responses of IC neurons to the weak stimulus, as shown in Fig. 5b.

The synaptic changes for strong and weak stimulus are adjusted by the amount of feedback signals from DSCF to IC. The strong stimulus elicits an increased activity of IC neurons, resulting in a large amount of feedback to IC neurons. This large feedback makes the change of inhibitory synapses to be more than the change of excitatory synapses. By contrast, the weak stimulus elicits a weak firing of IC and DSCF neurons, producing a feedback less than that for strong stimulus. This facilitates the change of excitatory synapses. Thus the interaction between IC and DSCF via the

synaptic changes of the top-down connections leads to the compressed and expanded expression of auditory information, depending on the stimulus intensity.

4 Concluding Remarks

We have presented here the neural model for detecting Doppler-shifted frequency of echo sound reflecting from flying insect. We showed that the model well reproduces the two kinds of BF shifts of IC neurons, or centripetal and centrifugal BF shifts. We also clarified the neural mechanism by which the direction of BF shift is determined by the feedback signals from DSCF neurons.

References

1. Ma, X., Suga, N.: Long-term cortical plasticity evoked by electric stimulation and acetylcholine applied to auditory cortex. *Proc. Natl. Acad. Sci. USA* 102, 9335–9340 (2005)
2. Sakai, M., Suga, N.: Plasticity of cochleotopic (frequency) map in specialized and nonspecialized auditory cortices. *Proc. Natl. Acad. Sci. USA* 98, 3507–3512 (2001)
3. Suga, N., Ma, X.: Multiparametric corticofugal modulation and plasticity in the auditory system. *Nat. Rev. Neurosci.* 4, 783–794 (2003)
4. Xiao, Z., Suga, N.: Reorganization of the cochleotopic map in the bat's auditory system by inhibition. *Proc. Natl. Acad. Sci. USA* 99, 15743–15748 (2002)
5. Suga, N.: Cortical computational maps for auditory imaging. *Neural Networks* 3, 3–21 (1990)
6. O'Neill, W.E., Suga, N.: Encoding of target range and its representation in the auditory cortex of the mustached bat. *J. Neurosci.* 2, 17–31 (1982)
7. Liu, W., Suga, N.: Binaural and commissural organization of the primary auditory cortex of the mustached bat. *J. Comp. Physiol. A* 181, 599–605 (1997)
8. Suga, N., Manabe, T.: Neural bases of amplitude-spectrum representation in auditory cortex of the mustached bat. *J. Neurophysiol.* 47, 225–255 (1982)

Efficient Representation by Horizontal Connection in Primary Visual Cortex

Hiroaki Sasaki^{1,2}, Shunji Satoh³, and Shiro Usui²

¹ Faculty of Engineering, Tohoku University
6-6-5 Aoba, Aramaki, Aoba-ku, Sendai 980-8579, Japan

² Neuroinformatics, RIKEN BSI
2-1 Hirosawa, Wako, Saitama 351-0198, Japan
³ Graduate School of Information Systems,
The University of Electro-Communications

1-5-1 Chofugaoka, Chofu, Tokyo 182-8285, Japan
sasaki@riec.tohoku.ac.jp

Abstract. Neurons in the primary visual cortex (V1) encode natural images that are exposed. As a candidate encoding principle, the efficient coding hypothesis was proposed by Attneave (1954) and Barlow (1961). This hypothesis emphasizes that the primary role of neurons in the sensory area is to reduce the redundancy of the external signal and to produce a statistically efficient representation. However, the outputs of neurons in V1 are statistically dependent because their classical receptive fields largely overlap and natural images have structures such as edges and textures. As described in this paper, we propose that the computational role of horizontal connections (HCs) is to decrease statistical dependency and attempt to self-organize the spatial distribution of HCs from natural images. In addition, we show that our neural network model with self-organized HCs can reproduce some nonlinear properties of V1 neurons, *e.g.* size-tuning and contextual modulation. These results support the efficient coding hypothesis and imply that HCs serve an important role in decreasing statistical dependency in V1.

Keywords: primary visual cortex, sparse coding, efficient coding, horizontal connection.

1 Introduction

Retinal images are optical projections of natural images comprising lines, curves, textures, and so on. Spatially adjacent local image-patches in such natural images are not statistically independent. Therefore, there should exist an appropriate encoding algorithm that is optimally tuned for natural images. The efficient coding hypothesis [12] is a theory for encoding processes in our sensory systems and neural information processing.

The primary visual cortex (V1) is thought to be the stage executing such efficient encoding. The outputs of V1 neurons, which are the encoding results, can be modelled by spatial correlation between classical receptive fields (CRFs)

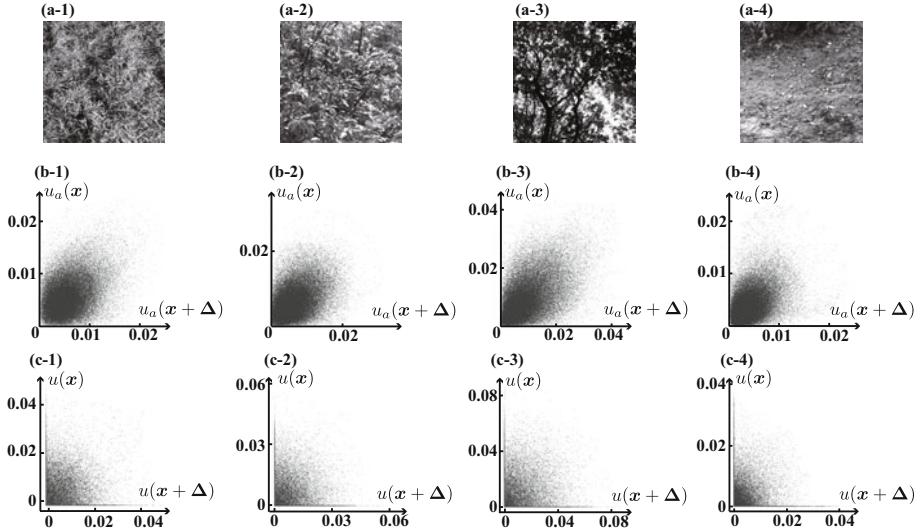


Fig. 1. Visualization of statistical dependency. (b-1)–(b-4) present scatter plots for $u_a(\mathbf{r})$ and $u_a(\mathbf{r} + \Delta)$ to input images (a-1)–(a-4) (in these figures, $\Delta = (5.0, 0)$). (c-1)–(c-4) are for $u(\mathbf{r})$ and $u(\mathbf{r} + \Delta)$.

and retinal images. Sparse coding proposed by Olshausen and Field is that of an efficient encoding algorithm; it shows that the principle of maximizing sparsity of neural response is sufficient to explain the emergence of V1 CRFs [6].

However, more efficient representation is possible if we do not use the “*linear*” V1 model. Actually, the outputs of V1 model for natural images show statistical dependency between their outputs (Fig. 1, explained later) because CRFs of the nearby neuron in V1 largely overlap [3] and neural images have a structure like edges and textures. More efficient representation would be achieved if this statistical dependency could be decreased. Moreover, the *linear* V1 model cannot explain nonclassical receptive field phenomena such as size-tuning [4] and contextual modulation [8].

As described in this paper, we first investigate the statistical dependency of V1 afferent signals. Next, we presume that the statistical dependency can be decreased by horizontal connections (HCs) and attempt to self-organize the spatial distribution of HCs using natural images. To confirm the physiological validity of the self-organized HCs and to examine whether our model can reproduce size-tuning and contextual modulation, we construct a V1 neural network model and conduct numerical experiments.

2 Statistical Dependency of Afferent Signals in V1

We first define V1 afferent signals in our model to investigate the statistical dependency. The afferent signals are designated as $u_a(\mathbf{r}, \theta)$, where θ and \mathbf{r} denote a preferred orientation and the centre position of CRFs.

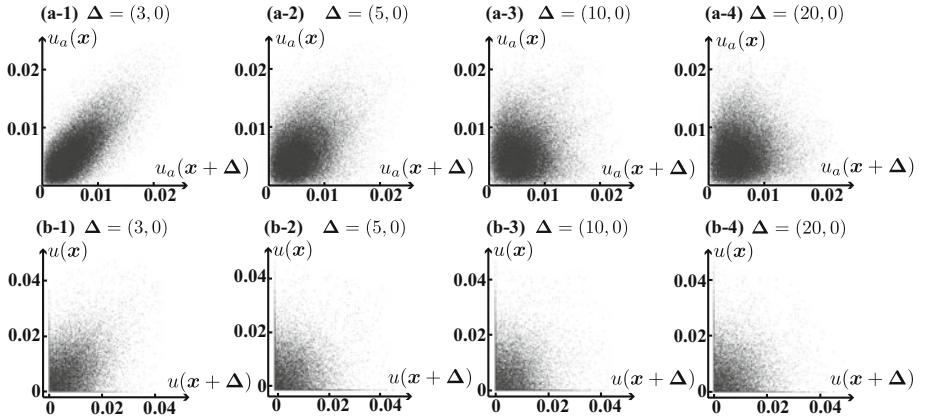


Fig. 2. Scatter plots for various Δ to the input image in Fig. II(a-1). The value Δ is shown in this figure.

$$u_a(\mathbf{r}, \theta) = \sqrt{(u_o(\mathbf{r}, \theta))^2 + (u_e(\mathbf{r}, \theta))^2}, \quad (1)$$

$$u_o(\mathbf{r}, \theta) = \left(\frac{d}{d\phi} g_\sigma \right) * J(\mathbf{r}), \quad (2)$$

$$u_e(\mathbf{r}, \theta) = \left(\frac{d^2}{d\phi^2} g_\sigma \right) * J(\mathbf{r}). \quad (3)$$

In the expressions above, $g_\sigma(\mathbf{r})$ denotes a Gaussian function with mean 0 and variance σ^2 , $\frac{d}{d\phi}$ signifies a derivative to $\phi (= \theta + 90^\circ)$ direction and $*$ denotes a linear convolution. In this afferent signal, $u_o(\mathbf{r}, \theta)$ and $u_e(\mathbf{r}, \theta)$ are the linear model for the simple cells, the CRFs of which are depicted as $\left(\frac{d}{d\phi} g_\sigma \right)$ and $\left(\frac{d^2}{d\phi^2} g_\sigma \right)$ [10], and $u_o(\mathbf{r}, \theta)$ is the afferent signal for complex cells.

Here, we visualize the statistical dependency of $u_a(\mathbf{r}, \theta)$ to natural images by scatter plots and investigate how statistical dependency exists in afferent signals. These natural images were downloaded from Van Hateren's Natural Image Database (<http://www.kyb.mpg.de/bethge/vanhateren/index.php>; ref. [9]). A parameter for eq. (2) and (3) is set to $\sigma = 3.0$. To increase the physiological validity, we adopt the computational V1 orientation map [5], which includes prominent features of the orientation map such as a singularity and a linear zone. Because θ is determined from the spatial position \mathbf{r} on this map, the afferent signals $u_a(\mathbf{r}, \theta(\mathbf{r}))$ are rewritten as $u_a(\mathbf{r})$ hereinafter.

The results for four natural images are depicted in Fig. II(b-1)–(b-4). The vertical and horizontal axes respectively show $u_a(\mathbf{r})$ and $u_a(\mathbf{r} + \Delta)$ ($\Delta = (5.0, 0)$) in Fig. II. Most points in Fig. II(b-1) are grouped on the slanted (about 45 deg) line. This result means that $u_a(\mathbf{r})$ to natural images is linearly correlated with $u_a(\mathbf{r} + \Delta)$. Furthermore, similar results for different images can be confirmed (Fig. II(b-2)–(b-4)). Then, we examine the Δ dependency of the linear correlation

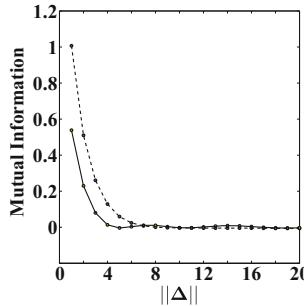


Fig. 3. Change of mutual information to $\|\Delta\|$. In this figure, the solid line signifies the mutual information between $u(\mathbf{r})$ and $u(\mathbf{r} + \Delta)$; the dotted line is for $u_a(\mathbf{r})$ and $u_a(\mathbf{r} + \Delta)$.

described above. Fig. 2 shows a scatter plot to the image in Fig. 1(a-1) for (a-1) $\Delta = (3.0, 0)$, (a-2) $\Delta = (5.0, 0)$, (a-3) $\Delta = (10.0, 0)$, and (a-4) $\Delta = (20.0, 0)$. Comparison with the case in $\Delta = (5.0, 0)$ (Fig. 2(a-2)) shows a stronger correlation between more adjacent neurons (Fig. 2(a-1)), although the linear correlation with more distant neurons becomes weaker (Figs. 2(a-3), 2(a-4)). In fact, mutual information between $u_a(\mathbf{r})$ and $u_a(\mathbf{r} + \Delta)$ decreases monotonically as $\|\Delta\|$ increases (Fig. 3). Therefore, these results mean that the afferent signals of nearby neurons are statistically dependent because of linear correlation.

3 Self-Organization of Horizontal Connection

In this section, we attempt to self-organize HCs to decrease the spatial correlation described above in afferent signals.

The V1 neural response $u(\mathbf{r})$ in our model is decided by two signals as the following (Fig. 4):

$$\begin{aligned} u(\mathbf{r}) &= u_a(\mathbf{r}) + \iint w(\xi) u(\mathbf{r} - \xi) d\xi, \\ &= u_a(\mathbf{r}) + w * u(\mathbf{r}), \end{aligned} \quad (4)$$

where the first term in (4) is the afferent signal, as defined in eq. (1), and the second term in (4) represents lateral signals weighted by $w(\xi)$ from the V1 neurons with different spatial position ξ ; also, the $w(\xi)$ can be interpreted as HCs. To self-organize $w(\xi)$ so that $u(\mathbf{r})$ has less linear correlation, the following cost function is adopted.

$$E[u, w] = \iint |u_a(\mathbf{r}) - u(\mathbf{r}) + w * u(\mathbf{r})|^2 d\mathbf{r} + \lambda \iiint |u(\mathbf{r}) u(\mathbf{r} + \Delta)| d\Delta d\mathbf{r}, \quad (5)$$

In that function, the first term is the fitting term; the second term denotes the approximation of the linear correlation between nearby neurons, and λ is a regularization parameter. To estimate $w(\xi)$ and $u(\mathbf{r})$, this cost function is minimized.

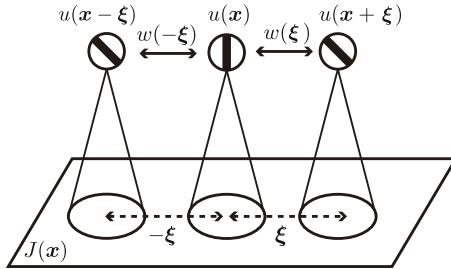


Fig. 4. Schematic representation of our model. Neurons in our model receive signals of two types: afferent signals and lateral signals by HCs.

For optimization, we use the steepest descent method. $w(\xi)$ is optimized after the optimization of $u(r)$. All of these functionals are calculated until they converge. To avoid a trivial possible solution $u(r) = 0$, we presume that during this optimization, the total amount of power for $u_a(r)$ is preserved in $u(r)$ as follows.

$$\iint u_a(r)^2 dr = \iint u(r)^2 dr \quad (6)$$

4 Numerical Experiment

In this section, we examine whether the linear correlation in afferent signals decreases in $u(r)$ and discuss how HCs are self-organized by numerical experiments. In this numerical experiment, parameters are set to $\sigma = 3.0$ and $\lambda = 1.0$. The range of $\Delta = (\Delta_x, \Delta_y)$ and $\xi = (\xi_x, \xi_y)$ is $\{|\Delta| - 19 \leq \Delta_x, \Delta_y \leq 19\}$ and $\{|\xi| - 21 \leq \xi_x, \xi_y \leq 21\}$. All of initial values for $u(r)$ and $w(\xi)$ are zero.

4.1 Correlation Reduction and Self-Organization of Horizontal Connection

Scatter plots for estimated $u(r)$ and $u(r + \Delta)$ are shown in Figs. 1(c-1)–(c-4). Stimuli to each scatter plot are natural images in Figs. 1(a-1)–(a-4). Most points in Figs. 1(c-1)–(c-4) concentrate on the horizontal and vertical axis. The 45 deg line in Figs. 1(b-1)–(b-4) on which most points gathered, disappears. We also investigated the Δ dependency like that shown in Figs. 2(a-1)–(a-4). These results presented in Figs. 2(b-1)–(b-4) also indicate that for all Δ , most points gather on the horizontal and vertical axis and the linear correlation decreases.

To evaluate the decrease of statistical dependency quantitatively, the mutual information for $u(r)$ and $u(r + \Delta)$ is shown in Fig. 3. In $||\Delta|| \leq 6$, the mutual information for $u(r)$ is apparently smaller than the mutual information for $u_a(r)$. Furthermore, the estimated $u(r)$ is shown as sparser than $u_a(r)$ (Fig. 5). These results indicate that, in the estimated $u(r)$, the statistical dependency of nearby neurons decreases and the sparsity of outputs increases.

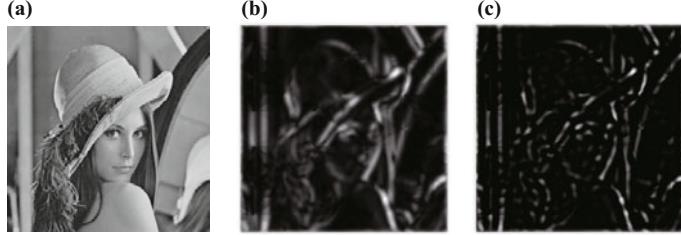


Fig. 5. More sparse output for $u(\mathbf{r})$: (a) is the input image, (b) and (c) respectively show $u_a(\mathbf{r})$ and estimated $u(\mathbf{r})$

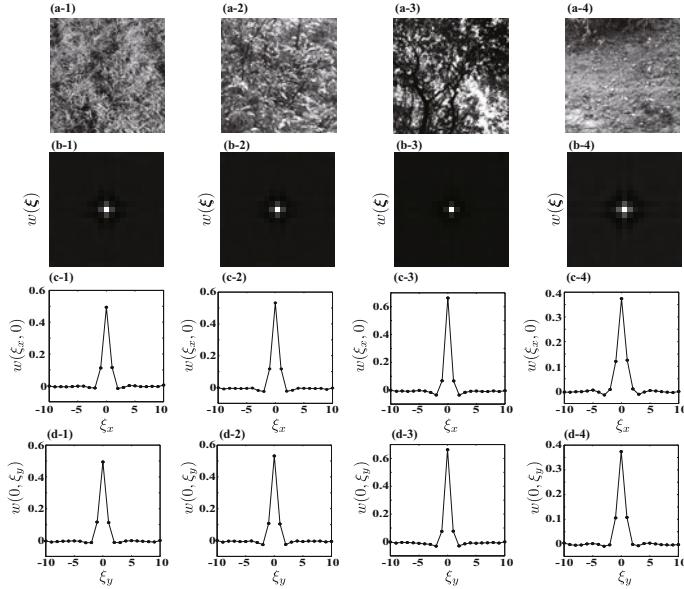


Fig. 6. Self-Organized $w(\xi)$ to four natural images (a-1)–(a-4). (b-1)–(b-4) are the two-dimensional distribution of $w(\xi)$. Also, (c-1)–(c-4) and (d-1)–(d-4) represent the one-dimensional distribution of $w(\xi)$ to ξ_x and ξ_y axis.

Next, we self-organize the spatial distribution of the horizontal connection $w(\xi)$ from eq.(5). The self-organized $w(\xi)$ to each image is shown in Fig. 6. Surprisingly, all the self-organized $w(\xi)$ to different images show a similar distribution, which is a Mexican hat shape with centre excitatory and surround inhibition (Fig. 6). This spatial distribution is qualitatively consistent with the horizontal connection adopted in the previous model [5].

4.2 Size-Tuning and Contextual Modulation

We execute the numerical experiment for size-tuning and contextual modulation to confirm the physiological validity of self-organized $w(\xi)$. In this numerical

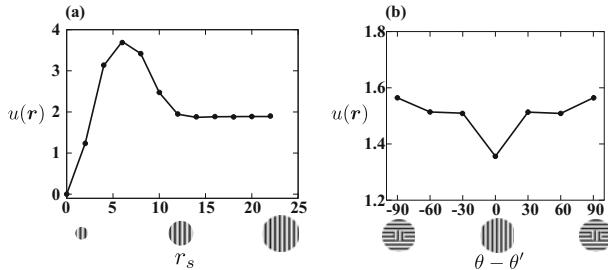


Fig. 7. Results of numerical experiments for (a) size-tuning and (b) contextual modulation

experiment, the stimulus for size-tuning is a grating with radius r_s , optimal orientation and spatial frequency for a fixed neuron. For contextual modulation, the stimulus orientation comprises θ within CRFs and θ' outside CRFs. Neural outputs $u(\mathbf{r})$ are calculated from eq. (4) using the estimated $w(\xi)$ from 20 natural images.

The results are shown in Fig. 7. In size-tuning, the neural output shows a maximal response to particular r_s and decreases to larger stimuli. This result is qualitatively consistent with physiological data [4]. Regarding the result for contextual modulation, the neuron strongly responds to a cross orientation pattern and shows minimal response to a uniform orientation pattern. This result also shows qualitative consistency with physiological data [8].

5 Discussion and Conclusion

As described in this paper, we pointed out that the afferent signals of nearby neurons are statistically dependent because of linear correlation. We estimated neural responses, which have a weaker linear correlation and self-organized the horizontal connection in V1 from natural images. These results indicate that statistical dependency in afferent signals decreases in the estimated neural response. The results also show a self-organized horizontal connection with similar distribution to different images. In addition, we constructed a V1 neural network model with self-organized horizontal connection. The model can reproduce size-tuning and contextual modulation. These results support the efficient coding hypothesis and imply that the computational role of horizontal connection is to decrease the statistical dependency of V1 outputs.

The possible source of this linear correlation is considered to be largely overlapping CRFs [3] and the statistical structure in natural images. These CRFs produce a set of non-orthogonal bases and generate a linear correlation in neural outputs even though the stimuli are linearly uncorrelated. In addition, some researchers report that statistical dependency remains in V1 afferent signals because a linear model can not capture the entire structure in natural images [7]. The difference between their research and ours is that non-orthogonal bases and

the computational orientation map[5] were adopted to increase the physiological validity. Moreover, the horizontal connection was self-organized.

Image compression is a presumable application of our model. Input images are represented by sparser outputs of the model neurons (Fig.5) by decreasing linear correlation of the initial outputs. This is our future work.

Acknowledgements

This research was partially supported by research funding from IIARE, Tohoku University and a Grant-in-Aid for Young Scientists (#20700279) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. Attneave, F.: Some informational aspects of visual perception. *Psychological review* 61(3), 183–193 (1954)
2. Barlow, H.B.: Possible principles underlying the transformation of sensory messages *Sensory Communication*. MIT Press, Cambridge (1961)
3. Buzas, P., Volgushev, M., Eysel, U.T., Kisvarday, Z.F.: Independence of visuotopic representation and orientation map in the visual cortex of the cat. *European Journal of Neuroscience* 18(4), 957–968 (2003)
4. DeAngelis, G.C., Freeman, R.D., Ohzawa, I.: Length and width tuning of neurons in the cat's primary visual cortex. *Journal of Neurophysiology* 71(1), 347–374 (1994)
5. Okamoto, T., Watanabe, M., Aihara, K., Kondo, S.: An explanation of contextual modulation by short-range isotropic connections and orientation map geometry in the primary visual cortex. *Biological Cybernetics* 91(6), 396–407 (2004)
6. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381(6583), 607–609 (1996)
7. Schwartz, O., Simoncelli, E.P.: Natural signal statistics and sensory gain control. *Nature neuroscience* 4(8), 819–825 (2001)
8. Sillito, A.M., Grieve, K.L., Jones, H.E., Cudeiro, J., Davis, J.: Visual cortical mechanisms detecting focal orientation discontinuities. *Nature* 378(6556), 492–496 (1995)
9. Van Hateren, J.H., Van der Schaaf, A.: Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society B: Biological Sciences* 265(1394), 359–366 (1998)
10. Young, R.A.: The gaussian derivative model for spatial vision: I. retinal mechanisms. *Spatial Vision* 2(4), 273–293 (1987)

Stimulation of the Retinal Network in Bionic Vision Devices: From Multi-Electrode Arrays to Pixelated Vision

Robert G.H. Wilke, Gita Khalili Moghaddam, Socrates Dokos, Gregg Suaning,
and Nigel H. Lovell

Graduate School of Biomedical Engineering, The University of New South Wales, Sydney,
Australia
R.Wilke@unsw.edu.au

Abstract. Bionic vision devices aiming to restore visual perception in the vision impaired rely on microelectrode arrays that can be implanted under the diseased retina. Arrays used today in first human trials are high density monopolar arrays comprising up to 1500 electrodes in a 3x3 mm a simple field calculations demonstrate that such high density arrays suffer from degradation of contrast between those 1500 stimulation sites when driven simultaneously. This effect can be described as electric crosstalk between the electrodes that strongly depends on the number of electrodes on such an array and proximity of electrodes to the target cells. The limit of spatial frequency of visual patterns that could be resolved by such arrays can be assessed to be 4.5; 1.2; and 0.7 cycles/mm, for an anticipated distance of target neurons of 20 µm, 200 µm, and 400 µm, respectively. This relates to a theoretically best achievable visual acuity of 2%, 0.6%, and 0.3% of normal vision, respectively (logMAR 1.7; 2.2; 2.5). These data suggest that novel strategies have to be pursued to either get closer to target structures, e.g. by the use of penetrating electrode arrays, or to create more confined stimulating fields within the retina, e.g. by the use of hexagonal arrays with multiple electrodes guarding one active electrode.

Keywords: bionic vision, retinal prosthesis, artificial vision, neural prosthesis, multi-electrode array.

1 The Concept of Retinal Prosthesis

More than 15 million people worldwide suffer from incurable blindness, mainly caused by retinitis pigmentosa (RP) and age-related macular degeneration (AMD) [1]. These diseases lead to a progressive degeneration of the photoreceptors of the retina. The retina contains first, second, and third order neurons of the visual pathway, and comprises intricate image analysis capabilities. During the disease process mainly the first order neurons degenerate depriving the neuronal system of its sensors and transducers. The remaining network is believed to be functional over a prolonged period of time [2]. A number of research groups worldwide have tested artificial vision devices that aim at replacing lost photoreceptor function. These devices must comprise a light

sensitive entity, a transducer to transform that information into a neural signal, and an interface to the biological system. A few of these devices are already in clinical trials [3-9]. It has been demonstrated that limited visual function in entirely blind persons can be restored up to a degree where recognition of letters is possible [10-12]. Such systems use either an external camera or an implantable micro-photodiode array as light sensors [13], convert this information into electrical signals and use microelectrode arrays as interfaces to second order neurons in the retina. Presently used arrays comprise 16 to 1500 electrodes of 50 – 600 μm diameters each, which are driven in a monopolar way, i.e. all electrodes of the array serve as a current source with a current sink in the form of a distant large return electrode (usually implanted outside the eye).

2 Electric Crosstalk in Micro-Electrode Arrays

It is tempting to think of one electrode of such a multi-electrode array as the picture element (pixel) of an image that would be perceived by a blind subject. However, there are many levels of complexity involved before currents injected by an electrode become perceived as an image, and many of these layers are yet poorly understood. The first level of complexity is the physical effects that occur when charge is injected by multiple sources into a complex and anisotropic tissue. Other levels of complexity include the questions of how extracellular currents activate neural structures within the retinal network, how this information is processed within a neuron, within the retinal network, as well as by higher order neurons, and how this information is eventually perceived.

We focus here solely on physical effects and discuss to what extent a separation of electrode array elements and picture elements must be considered.

One effect that accounts for degradation of the resolution of a perceived image is electric crosstalk between electrodes.

Crosstalk between electrodes describes the effect that the electric field that certain neural structures are surrounded and stimulated by is not only defined by a single electrode, but rather by the activity of all units on an electrode array. Considering one electrode as one single channel by means of which an image can be rendered, crosstalk simply means that this channel is compromised by the simultaneous activity on all other channels.

Electrode arrays used in clinical trials today are designed to use monopolar stimulation, i.e. a number of electrodes are driven with only one large and distant return electrode. This design is the most effective in terms of number of electrodes needed to create a single phosphene, and ignoring time multiplexing such possibly in terms of the electrode density in an array.

There are concepts to use advanced electrode configurations like dipolar-, tripolar-, or even septapolar electrode configurations, i.e. two, three, or seven electrodes to build up one functional unit of electrodes to generate one phosphene [14]. While these configurations seem to be less economic in terms of space they cover per pixel and possibly in terms of threshold currents, they may provide some distinctive advantages for they can deliver stimulation currents that are more closely confined to a certain target volume because they provide a current source and sink for each pixel. This is believed to lead to decreased crosstalk between electrodes.

We have formulated a simplified 3D finite element model of multi-electrode arrays to calculate electric field distribution over various distances from the surface of the array. Stimulation electrodes are defined by discs with a radius of 50 µm and a centre-centre spacing of 150 µm. The electrode array forms one boundary of a passive domain of 8x8x0.5 mm, a return electrode is assumed to be at infinity. The Poisson equation governing the voltage distribution, V , throughout the passive domain is:

$$-\nabla \cdot (\sigma \nabla V) = I \quad (1)$$

where σ is the conductivity of the medium and I is the volume current density injected into the medium at a given location. For physiological saline, σ is taken to be 1 S/m.

The boundary conditions of the model are such that a current of 1 mA was injected through each active stimulating electrode, whereas inactive electrodes were allowed to float. Therefore the boundary condition

$$\sigma \frac{\partial V(x, y)}{\partial z} = \frac{i_{stim}}{\sigma \pi r^2} \quad (2)$$

within each stimulus electrode with center coordinates (x, y) was introduced., where i_{stim} are absolute values of current, r is the electrode radius.

Simulations were conducted using COMSOL Multiphysics Version 3.5a (COMSOL AB, Sweden) on a Quad Core AMD Opteron Windows 64 server with 127 GB of RAM. Simulation time varied from 1 to 2 hours depending on server capacity and solver settings.

Crosstalk was calculated for a worst case scenario where all but the central electrode was active in a high density multi-electrode array (see fig. 1).

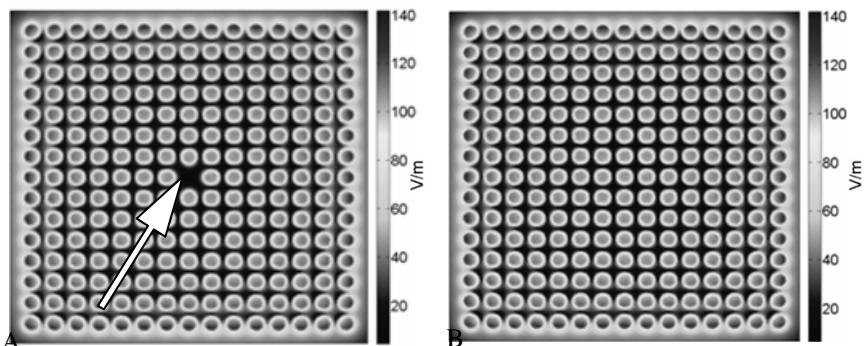


Fig. 1. Electric field distribution of 225 monopolar electrodes, 10 µm above the array. A) Central electrode (arrow) is OFF. B) All electrodes are active. Grey scale code denotes electric field in V/m.

The total number of electrodes was varied from 25 to 1681 electrodes. Crosstalk for the center electrode was calculated for each array as the ratio of electric field when all electrodes but the central electrode were active, $E_{center,OFF}$, to electric field when all electrodes were active, $E_{center,ON}$. The electric field was calculated for an imaginary target volume at a various distances along the z-axis, which was defined as a line from the center electrode leading orthogonal from the plane of the electrode array away from the array (positive z-direction):

$$Crosstalk\%(z) = \frac{E(z)_{center,OFF}}{E(z)_{center,ON}} \times 100 \quad (3)$$

It seems obvious that the further away from the surface of such an array, the more prominent the effect of crosstalk will get. Ideally one would like to implant such an electrode array as close as possible to the targeted volume, containing the 2nd or 3rd order neurons to be stimulated. However, due to surgical restrictions and disease process this is not always possible. Depending on different surgical approaches it must be anticipated that target volumes are at best 20 μm away from the electrode array, in many cases like subretinal or suprachoroidal implantation even more, in the order of 200 – 400 μm . Fig. 2 shows the effect that increasing number of electrodes and increasing distance of the target structures from an array has on crosstalk.

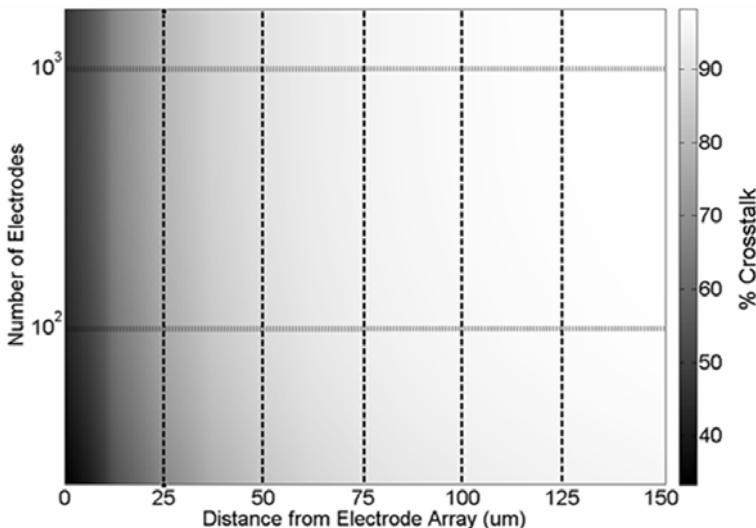


Fig. 2. Crosstalk dependency on number of electrodes and distance from array surface. Grey scale code denotes crosstalk in percentage.

As can be seen the ability to render pixelated images with the use of single electrodes is heavily impaired by crosstalk even with a moderate number of electrodes (100) and very close distances to target neurons (50 μm). However, this situation reflects the worst case scenario where all but one electrode is activated simultaneously. The importance of localized electrode activation on perceived image quality is considered in the next sections.

3 Degradation of Image Contrast by Crosstalk

Multi-electrode arrays for retinal implants are designed to be of high density in order to be able to render images with high spatial frequency. However, the higher the electrode

density gets, and the more electrodes are employed, the more pronounced the detrimental effect of crosstalk will become. This means the ability to generate pixelated patterns of stimulation with high spatial frequency is not only limited by the physical distance of electrodes, but also by crosstalk on that array. To assess this effect we have calculated electric field distribution using arrays where every second row is activated, simulating the generation of a grating pattern of highest spatial resolution the array could provide (see fig. 3A). This allows assessing the potential contrast in a perceived image of that grating, based on the contrast found in the electric field pattern. As our understanding how precisely extracellular stimulation currents lead to perception is very limited, we chose to use that strongly simplified approach of taken the contrast of electric fields as a surrogate marker for perceived image contrast. We calculated contrast in the electric fields as Michelson contrast:

$$C_m = \frac{E_{\max} - E_{\min}}{E_{\max} + E_{\min}} \quad (4)$$

With E_{\max} and E_{\min} (see fig. 3B) representing the maximum and minimum value of electric field at given points above the surface of the array. It returns a value between zero and one; the closer to one, the higher the resolution.

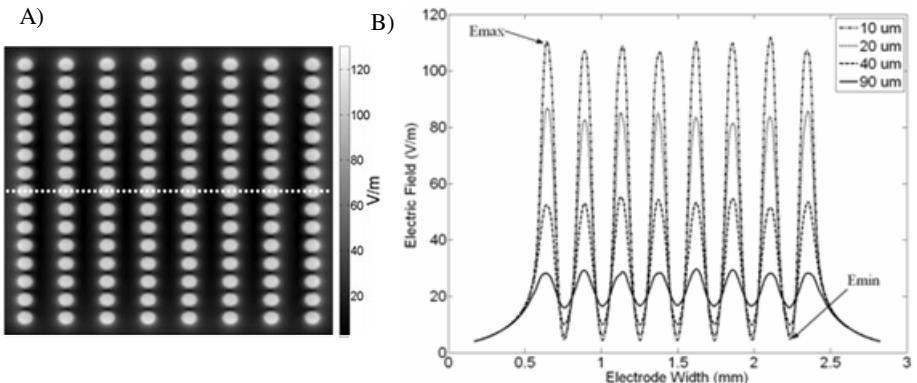


Fig. 2. A) Electric field (V/m) for biphasic stimulation of 255 monopolar electrodes 10 μm above the electrode array. Grey scale code indicates electric field (V/m); B) Electric field as a function of distance from electrode array for 225 electrodes along dashed line (see fig. 3A). Arrows indicate E_{\max} and E_{\min} of electric field.

This analysis shows that contrast is effectively low pass filtered in dependency on the distance between electrode array and target tissue.

Fig. 4 illustrates how the spacing between electrodes affects image contrast. As anticipated, contrast increases with increasing electrode spacing whilst, of course, losing spatial resolution. There is obviously a trade-off between density of electrodes on the array and acceptable contrast.

To estimate the best trade-off and therefore the limits of spatial resolution of such arrays in relation of their distance to target neurons, we defined a contrast of 0.5 as the minimal acceptable value in order to correctly perceive an image. Unfortunately it is poorly understood how the visual system processes such artificial stimulation patterns, therefore the cut-off value of 0.5 is arbitrary. We then calculated the respective contrast generated by arrays with a centre-centre distance of 110 μm up to 720 μm . Contrast was calculated for 3 different anticipated distances from the target neurons (20, 200 and 400 μm , respectively). The spacing at which a contrast of 0.5 was reached was taken as the minimal acceptable distance. From fig. 5 it can be seen that with an anticipated distance from target neurons of 400 μm , the densest array that leads to an acceptable contrast is 720 μm center-center distance between electrodes. This relates to a maximal spatial resolution of 0.7 cycles / mm. Considering that a distance on the retina of 278 μm relates to 1 degree of visual angle, a respective visual acuity of 0.003 (20/6200; logMAR 2.49) can be assessed. This is equivalent of 0.3% of normal visual acuity. Respective values for an anticipated distance of 200 μm are 1.25 cycles / mm, visual acuity of 0.005 (20/3500; logMAR 2.23), or 0.5%. Only for the closest anticipated distance, 20 μm , resolution is not limited by crosstalk, but by distance of electrodes. With an electrode diameter of 100 μm , the center-center distance cannot get much smaller than 110 μm without the electrodes touching each other. The resolution would be 4.55 cycles / degree, visual acuity of 0.02 (20/900; logMAR 1.68), or 2% of normal visual acuity.

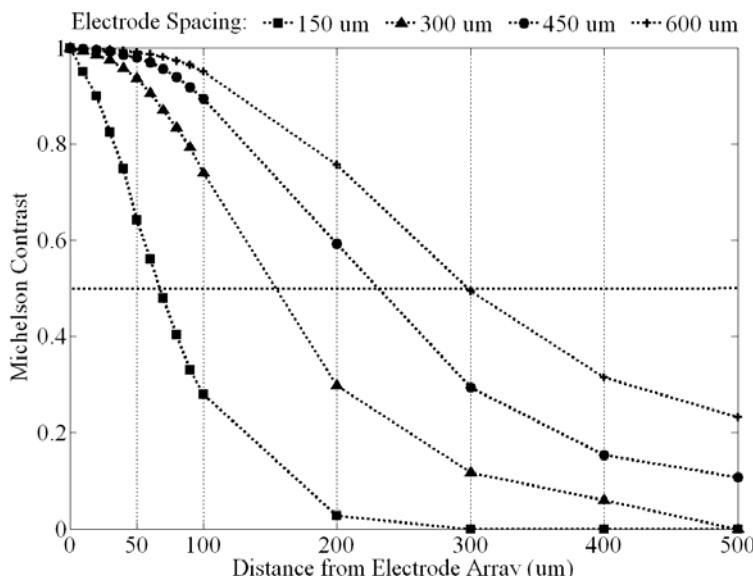


Fig. 3. Influence of electrode spacing and distance to target structure on contrast in electric field patterns

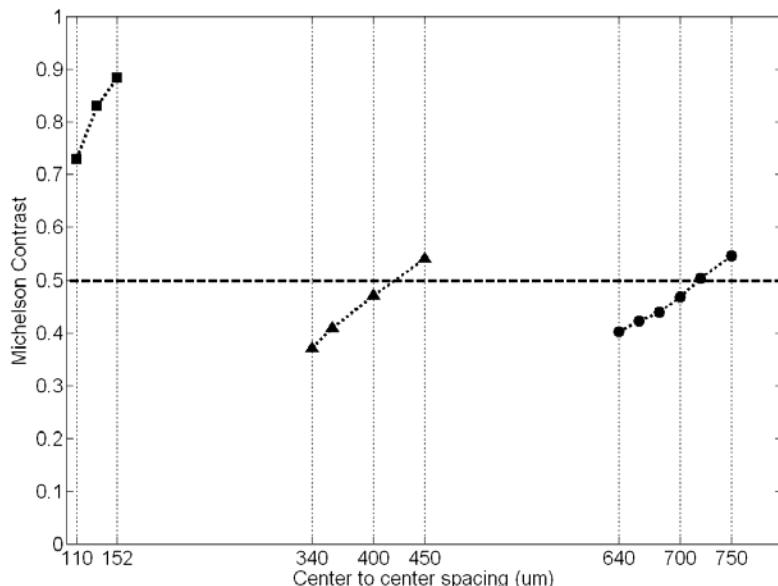


Fig. 5. Limiting effect of electrode spacing on spatial resolution; electrode radius = 50 μm . Michelson contrast is calculated for target cells situated ■: 20 μm ; ▲: 200 μm and ●: 400 μm away from the electrode array.

4 Consequences for Retinal Prosthesis Design

In this study we attempted to address the question of how electrode arrangement can be employed in tuning the resolution of perceived images. These data suggest that spatial resolution that can be achieved by an implantable bionic device is limited by current electrode array design. With current electrode designs using planar electrodes it is difficult to get closer to the target neurons, due to surgical restrictions. Typical distances to target neurons can be estimated to be at best 20 μm for epiretinal implantation and possibly 400 μm for suprachoroidal implantation. 3D structured penetrating microelectrode arrays which are under development may have the distinct advantage to get much closer to target neurons.

Another strategy would be not to use concurrent simultaneous stimulation, but some modes of sequential stimulation, at the expense of temporal resolution [15].

Finally methods of current focusing using multipolar electrode designs could significantly reduce crosstalk and therefore image degradation [14,16], possibly at the expense of higher stimulation currents and power consumption of implants. Further simulations and experimental studies will help to find the optimal strategy for the different implantation sites and electrode arrays.

References

1. Congdon, N., et al.: Causes and prevalence of visual impairment among adults in the United States. *Arch. Ophthalmol.* 122, 477–485 (2004)
2. Jones, B.W., Marc, R.E.: Retinal remodeling during retinal degeneration. *Exp. Eye Res.* 81, 123–137 (2005)

3. Yanai, D., et al.: Visual performance using a retinal prosthesis in three subjects with retinitis pigmentosa. *Am. J. Ophthalmol.* 143, 820–827 (2007)
4. Besch, D., et al.: Extraocular surgery for implantation of an active subretinal visual prosthesis with external connections: feasibility and outcome in seven patients. *Br. J. Ophthalmol.* 92, 1361–1368 (2008)
5. Gerdin, H., Benner, F.P., Taneri, S.: Experimental implantation of epiretinal retina implants (EPI-RET) with an IOL-type receiver unit. *J. Neural Eng.* 4, S38–S49 (2007)
6. Winter, J.O., Cogan, S.F., Rizzo, J.F.: 3rd. Retinal prostheses: current challenges and future outlook. *J. Biomater Sci. Polym. Ed.* 18, 1031–1055 (2007)
7. Hornig, R., et al.: A method and technical equipment for an acute human trial to evaluate retinal implant technology. *J. Neural Eng.* 2, S129–S134 (2005)
8. Walter, P., et al.: Cortical activation via an implanted wireless retinal prosthesis. *Invest Ophthalmol Vis. Sci.* 46, 1780–1785 (2005)
9. DeMarco Jr., P.J., et al.: Stimulation via a subretinally placed prosthetic elicits central activity and induces a trophic effect on visual responses. *Invest Ophthalmol Vis. Sci.* 48, 916–926 (2007)
10. Zrenner, E., et al.: Subretinal microelectrode arrays implanted into blind retinitis pigmentosa patients allow recognition of letters and direction of thin stripes. In: IFMBE Proceedings World Congress on Medical Physics and Biomedical Engineering, vol. 25, pp. 444–447 (2009)
11. Wilke, R., et al.: Visual acuity determined by landolt c test in a blind patient provided with a subretinal electronic implant. *Invest. Ophthalmol. Vis. Sci.* 2009 : E-Abstract (2009)
12. Wilke, R., Zrenner, E.: Clinical results: Thresholds and visual sensations elicited by subretinal implants in 8 patients. In: The Eye and the Chip World Congress on Artificial Vision (Detroit, 2008) (2008)
13. Zrenner, E.: Will retinal implants restore vision? *Science* 295, 1022–1025 (2002)
14. Dommel, N.B., et al.: A CMOS retinal neurostimulator capable of focussed, simultaneous stimulation. *J. Neural Eng.* 6, 35006 (2009)
15. Benav, H., Wilke, R., Stett, A., Zrenner, E.: Modeling advantages of subretinal microphotodiode-arrays utilizing sequential electrode activation. *Invest. Ophthalmol. Vis. Sci.* 2009 : E-Abstract (2009)
16. Addi, M.M., et al.: Charge recovery during concurrent stimulation for a vision prosthesis. In: Conf. Proc. IEEE Eng. Med. Biol. Soc. 2008, pp. 1797–1800 (2008)

Spatial Feature Extraction by Spike Timing Dependent Synaptic Modification

Kazuhisa Fujita

Department of Computer and Information Engineering, Tsuyama National Collage of Technology, 654-1 Numa , Tsuyama, Okayama, Japan 708-8506
kazu@spikingneuron.net

Abstract. Spike timing dependent synaptic plasticity (STDP) is found in various areas of the brain, visual cortex, hippocampus and hindbrain of electric fish, etc. The synaptic modification by STDP depends on time difference between pre- and postsynaptic firing time. If presynaptic neuron fires earlier than postsynaptic neuron dose, synaptic weight is strengthened. If postsynaptic neuron fires earlier than presynaptic neuron dose, synaptic weight is weakened. This learning rule is one example of various rules (hippocampal type). The learning rule of electric fish type is reversed to the rule of hippocampal type. Changes of synaptic efficiency precisely depend on timing of pre- and postsynaptic spikes under STDP. Because of this precise dependence, it is thought that STDP plays the important role in temporal processing. Temporal processing by STDP is well known. However, the role of STDP in spatial processing is not enough understood. In present study, we propose two type spatial filter by STDP on interconnected network. One is high-pass filter when the learning rule is hippocampal type. Another is low-pass filter when the learning rule is electric fish type. We show that synaptic modification based on STDP may play important role in spatial processing.

1 Introduction

Spike timing dependent synaptic plasticity (STDP) is found in various areas of the brain. The synaptic modification by STDP precisely depends on timing between pre- and postsynaptic spike firing. According to precise dependence on spike timing, many researchers have focused the processing of temporal information by STDP. We, however, focused the processing of spatial information of stimulus by STDP. In this study, we propose that spike timing dependent synaptic modification provide processing of spatial information.

Spike timing dependent synaptic modification is found in various areas of various neural systems [1]. For example, the STDP is found in the visual cortex of a cat [2] and a rat [3], the hindbrain of a weakly electric fish [4] etc. Intensity of the modification depends on time difference between pre- and postsynaptic firing. For example, if presynaptic neuron fires more earlier than postsynaptic neuron dose, synaptic weight is strengthened. If postsynaptic neuron fires more earlier than presynaptic neuron dose, synaptic weight is weakened. This learning

rule of STDP is found in hippocampus [5]. The learning rule that is reverse of rule of hippocampal type is found in hindbrain of electric fish.

Many previous studies have focused the function of STDP in temporal coding. Learning based on STDP like rules plays the various role in temporal information processing, for example, coincident detection [6], storage of temporal sequence [7,8], direction selectivity in visual cortex [9]. The recurrent connection with STDP make stabilize selectivity of clusters of neurons [10]. However the role of timing based learning in spatial information processing is not well understood. In present study, we propose that STPD provides the function of spatial filter on interconnected network. We show two type spatial filter by STDP on interconnected network. One is high-pass filter using the learning window of hippocampus type. Another is low-pass filter using the learning window of electric fish type.

2 Model

Figure 1 shows the neural network used in this study. The structure of the network is one dimensional array. The network has 200 neurons. A neuron of the network has connection to neighbor neurons. The neuron i connects with neurons from $i - RF$ to $i + RF$. RF is the size of connecting area. We imposed periodic boundary condition on the network to avoid edge effect. One connection has 10 excitatory synapses. Synaptic weight is modified by the rule of STDP.

2.1 Neuronal Model

The Leaky integrate and fire neuron [11] was used as a single neuron. The membrane potential $V(t)$ of the neuron is determined by

$$\tau_m \frac{dV(t)}{dt} = V_0 - V(t) + I_{in}(t), \quad (1)$$

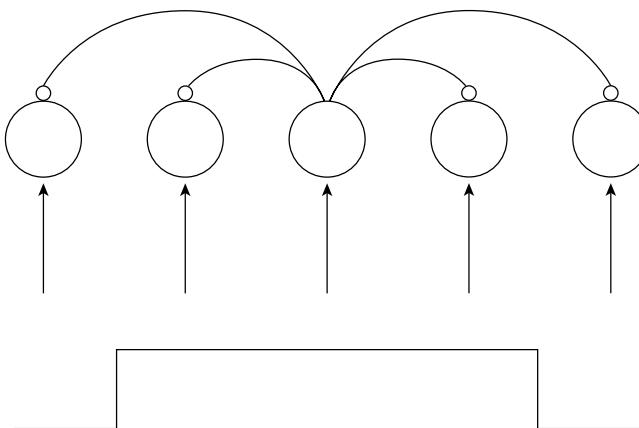


Fig. 1. The network is one dimensional array. The neuron connects neighbor neurons. A neuron receives afferent input each other (allow).

where τ_m is time constant, $In(t)$ is input of the neuron. The input consists of the stimuli and synaptic current from connected neurons in the network. V_0 indicates resting potential. The neuron generates a spike and resets the membrane potential to V_0 when the membrane potential reaches threshold ϑ . Spike $S(t)$ is determined by

$$S(t) = \begin{cases} 1 & \text{if } V \geq \vartheta \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

The neuron is stimulated by input current and synaptic current evoked by connected neurons. $In(t)$ is summed up input current and synaptic current. $In(t)$ is determined by

$$In(t) = I_c + g(t)(E_V - V), \quad (3)$$

where I_c is input current, $g(t)$ is synaptic conductance. In present study, input current is direct current.

2.2 Synaptic Model

We used the synaptic model proposed by Song and Abbott [10]. The synaptic conductance $g(t)$ is determined by

$$\tau_g \frac{dg(t)}{dt} = -g(t), \quad (4)$$

where τ_g is time constant. This model is leaky integrator, thus synaptic conductance only decreases and converges into 0. Synaptic conductance $g(t)$ increases when the presynaptic neuron generates a spike. If the presynaptic neuron generates a spike at t , the synaptic conductance at $t + \Delta t$ is determined by

$$g(t + \Delta t) = g(t) + W(t), \quad (5)$$

where $W(t)$ is synaptic weight. On STDP, synaptic weight changes when the spikes from presynaptic neuron and postsynaptic neuron reaches the synapse. Synaptic weight $W(t)$ is given by

$$W(t + \Delta t) = W(t) + F(t_{\text{pre}} - t_{\text{post}}), \quad (6)$$

where F is synaptic change rate, t_{pre} is time of presynaptic spike arrival, t_{post} is time of postsynaptic firing. Initial value of $W(t)$ is chosen W_0 . Synaptic changes F is determined by

$$F(t_{\text{pre}} - t_{\text{post}}) = \begin{cases} A_+ \exp((t_{\text{pre}} - t_{\text{post}})/\tau_+) & \text{if } \Delta t > 0 \\ A_- \exp((t_{\text{pre}} - t_{\text{post}})/\tau_-) & \text{otherwise} \end{cases}. \quad (7)$$

Figure 2B and C shows the form of F . When $t_{\text{pre}} - t_{\text{post}}$ is negative and F is positive, learning window of STDP is shown in Fig. 2B. This learning rule is shown in hippocampus. When $t_{\text{pre}} - t_{\text{post}}$ is negative and F is negative, learning window of STDP is shown in Fig. 2C. This learning rule is shown in electrosensory lateral line lobe of electric fish.

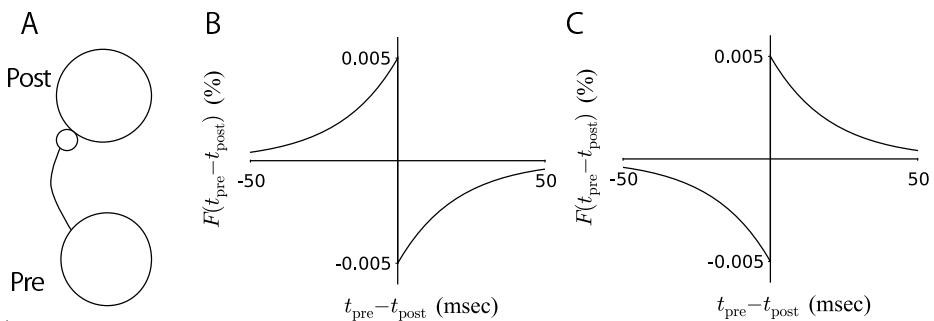


Fig. 2. A shows relation of pre- and postsynaptic neuron. The learning windows of STDP. B and C shows hippocampal and electric fish type of learning window, respectively. Horizontal line of B and C indicates the time difference $t_{\text{pre}} - t_{\text{post}}$. Vertical line indicates the rate of synaptic modification.

When learning window is hippocampal type, the values of parameters are: $RF = 8$, $\tau_m = 20\text{msec}$, $V_0 = -70\text{mV}$, $\vartheta = -54\text{mV}$, $E_V = 0\text{mV}$, $\tau_g = 5\text{msec}$, $A_+ = -0.005$, $A_- = -A_+$, $\tau_{\text{pre}} = 20\text{msec}$, $\tau_{\text{post}} = 20\text{msec}$, $W_{\text{max}} = 0.005$, $W_0 = W_{\text{max}}/2$.

3 Results

3.1 Spatial High-Pass Filter by Hippocampal Type Rule

In this subsection, rectangularly distributed direct current stimulated the network (shown in Fig. 1). Neuron 0 – 49 and 150 – 199 received direct current 20 mV. 50 – 149 neuron received direct current 35 mV. Firings of the neuron were evoked by direct current and postsynaptic potential.

The response of neurons from 9.5sec to 10.0sec is shown in Fig. 3 A without learning. In this case, the spatial form of input was represented on the response of network. On the other hand, Fig. 3B shows the response of network with learning based on hippocampal type rule (learning window shown in Fig. 2B) from 9.5sec to 10.0sec. The response went to equilibrium and synapses was almost sufficiently modified with learning. The response of neuron 50 and 149 was higher than one of other neurons receiving same stimulus. There were these neuron on the edge of spatial distribution of intensity of stimulus. In other words, the higher firing rate of neuron on the edge of intensity of stimulus was evoked by synaptic modification. The network with synaptic modification provided the function of edge extraction of stimulus.

Using hippocampal type rule, we achieved that the interconnected network with STDP provided the function of edge extraction. In this case, spike timing dependent synaptic modification caused to strengthen synaptic weight from the neuron stimulated by weaker input to the neuron stimulated by higher input. On the other hand, synaptic weight from the neuron stimulated by higher input

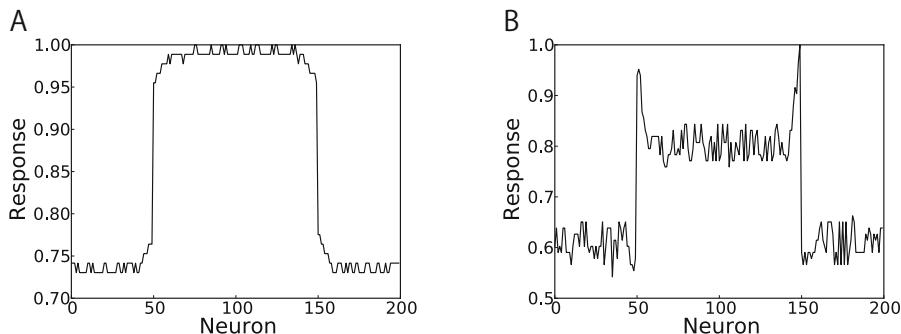


Fig. 3. Response of the network. A shows response of the network without learning. B shows response of the network with learning based on hippocampal type rule.

to the neuron stimulated by lower input was weakened. Thus the firing rate of the neuron stimulated by higher and lower inputs became higher and lower respectively. Therefor the edge that was the changing point of stimulus was extracted by learning according to timing learning based on hippocampal type rule. This function was regarded as spatial high-pass filtering.

3.2 Spatial Low-Pass Filter by Electric Fish Type Rule

In this subsection, direct current rectangularly distributed with noise stimulated the network (Fig. 1). Neuron 0 – 49 and 150 – 199 received direct current 20 mV with noise. 50 – 149 neuron received direct current 35 mV with noise. Noise was spatial white noise from 1 Hz to 100 Hz. One Hz as spatial frequency in present study had a spatial periodic interval of 200 neurons.

When learning rule is electric fish type, the network provided the function of low-pass filter. The network was stimulated by rectangularly distributed direct current with noise. We examined two cases. In one case, synaptic modification did not occur (Fig. 1A). In another case, synaptic modification occurred (Fig. 1B). In the first case, noise presented in response of the network and covered the rectangular form of stimulus. In the second case, high frequency noise was eliminated and the rectangular form of stimulus was represented on the response of the network.

In this result, high frequency component was reduced by the network with timing learning based on electric fish type rule (learning window shown in Fig. 2B). When firing rate of neuron A was higher than a neighbor neuron B, the synaptic weight from B to A was weakened while synaptic weight from A to B was strengthened. As the result, difference in firing rate of neuron A and neighbor neuron B was reduced. Thus the network with timing learning based on electric fish rule reduced high spatial frequency component.

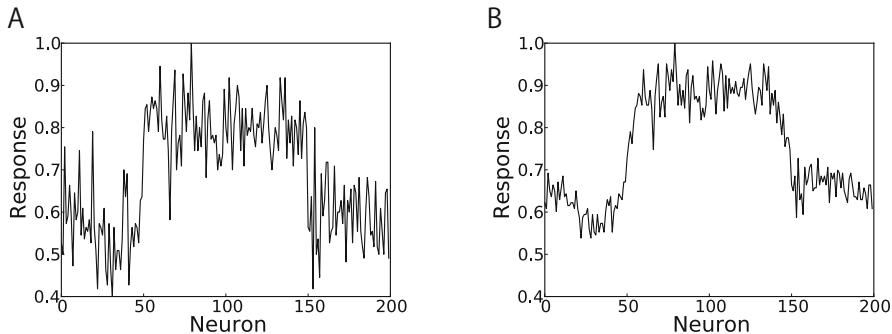


Fig. 4. Response of the network. A shows response of the network without learning. B shows response of the network with learning based on electric fish type rule.

3.3 Mechanism for Spacial Filter by Spike Timing Dependent Modification

We considered here how spike timing modification provide the function of spatial filter of stimulus. Figure 5 shows the schematic image of firing of presynaptic and postsynaptic neuron. In this figure, firing rate of postsynaptic neuron is more than one of presynaptic neuron. In this case, the number of spike pairs $\Delta t < 0$ (solid line shown in Fig. 5) is more than one of spike pares $\Delta t \leq 0$ (broken line shown in Fig. 5). Here, learning windows is hippocampal type. If firing rate of postsynaptic neuron is higher, spike pair $\Delta t < 0$ appears more frequently and summation of F decided by a spike pair is positive. Namely, higher firing rate of postsynaptic neuron forces to strengthen synaptic weight and to make firing rate of postsynaptic neuron higher. Thus, the synaptic weight from the neuron firing less frequently to the neuron firing more frequently is strengthened. Additionally, firing rate of the postsynaptic neuron firing more

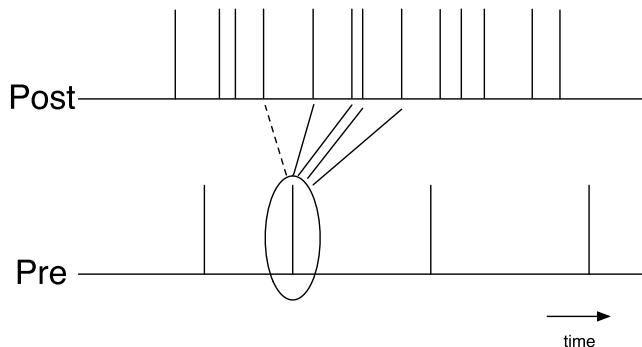


Fig. 5. The schematic image of spikes of pre- and post- synaptic neuron. Solid line indicates pre and post spike pare for $\Delta t < 0$. Broke line indicate pre and post spike pare for $\Delta t \leq 0$. $\Delta(t) = t_{\text{pre}} - t_{\text{post}}$.

frequently increases. On the other hand, if firing rate of postsynaptic neuron is less than one of presynaptic neuron, synaptic weight is weakened and firing rate of postsynaptic neuron becomes lower. Thus, difference of firing rate of post- and pre synaptic neuron occurs at the changing point of intensity of stimulus and here synaptic weight is strengthened or weakened. In summary, difference of firing rate of neurons near changing point of intensity of stimulus is enhanced by synaptic modification and high frequency components of stimulus is extracted. When synaptic modification obeys spike timing dependent plasticity using learning window of electric fish type, reverse phenomenon occurs.

4 Concluding Remarks

We have presented here two spatial filter by spike timing dependent plasticity. One is spatial high-pass filter that plays the role of edge extraction using hippocampal type learning rule. The other is spatial low-pass filter that plays the role of reduction of spatial high frequency noise using electric fish type learning rule. The results of present study suggest that STDP may play important role of not only temporal information processing but also spatial information processing.

References

1. Dan, Y., Poo, M.M.: Spike timing-dependent plasticity: From synapse to perception. *Physiol. Rev.* 86(3), 1033–1048 (2006)
2. Fu, Y.X., Djupsund, K., Gao, H., Hayden, B., Shen, K., Dan, Y.: Temporal specificity in the cortical plasticity of visual space representation. *Science* 296(5575), 1999–2003 (2002)
3. Sjostrom, P.J., Turrigiano, G.G., Nelson, S.B.: Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* 32, 1149–1164 (2001)
4. Bell, C.C., Han, V.Z., Sugawara, Y., Grant, K.: Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature* 387(6630), 278–281 (1997)
5. Bi, G.Q., Poo, M.M.: Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18, 10464–10472 (1998)
6. Gerstner, W., Kempter, R., van Hemmen, J.L., Wagner, H.: A neuronal learning rule for sub-millisecond temporal coding. *Nature* 383(6595), 76–81 (1996)
7. Roberts, P.D.: Computational consequences of temporally asymmetric learning rules: I. differential hebbian learning. *J. Comput. Neurosci.* 7, 235–246 (1999)
8. Roberts, P.D., Bell, C.C.: Computational consequences of temporally asymmetric learning rules: II. *J. Comput. Neurosci.* 9, 67–83 (2000)
9. Rao, R.P., Sejnowski, T.J.: Predictive learning of temporal sequences in recurrent neocortical circuits. *Novartis Found Symp.* 239, 208–229 (2001) discussion 229–240
10. Song, S., Abbott, L.F.: Cortical development and remapping through spike timing-dependent plasticity. *Neuron* 32(2), 339–350 (2001)
11. Gerstner, W., Kistler, W.M.: Spiking Neuron Models. Cambridge University Press, Cambridge (2002)

Learning Shapes Bifurcations of Neural Dynamics upon External Stimuli

Tomoki Kurikawa and Kunihiko Kaneko

University of Tokyo, Department of Basic Science
Komaba 3-8-1, Meguro-ku, Tokyo, Japan
kurikawa@complex.c.u-tokyo.ac.jp

Abstract. Memory is often considered to be embedded into one of the attractors in neural dynamical systems, which provides an appropriate output depending on the initial state specified by an input. However, memory is recalled only under the presence of external inputs. Without such inputs, neural states do not provide such memorized outputs. Hence, each of memories do not necessarily correspond to an attractor of the dynamical system without input and do correspond to an attractor of the dynamics system with input. With this background, we propose that memory recall occurs when the neural activity changes to an appropriate output activity upon the application of an input. We introduce a neural network model that enables learning of such memories. After the learning process is complete, the neural dynamics is shaped so that it changes to the desired target with each input. This change is analyzed as bifurcation in a dynamical system. Conditions on timescales for synaptic plasticity are obtained to achieve the maximal memory capacity.

Keywords: neural network, bifurcation, associative reward-penalty.

1 Introduction

One of the most important features of the brain is the ability to learn and generate an appropriate response to external stimuli. Output responses to the input stimuli are memorized as a result of synaptic plasticity. A wide variety of neural network models have been proposed to study how synaptic strength pattern is formed to memorize given input-output (I/O) relationships. In most of the studies [1], inputs are supplied as the initial conditions for neural activity, whose temporal evolution results in the generation of the desired outputs. However, recent experimental studies have shown that there exist structured spontaneous neural activity even in the absence of external stimuli [2]. Upon external stimuli, such neural activity is modified to provide appropriate response neural activities [3]. Considering these studies, we propose a novel perspective of the memorization of I/O relationships: If a neural system memorizes an I/O relationship, the spontaneous dynamics is modulated by a given input to provide a required output. In other words, the input changes the flow structure of the neural dynamical system to generate the corresponding output.

On the basis of this idea, we address the following questions: Can we construct an appropriate neural network model to demonstrate the learning process under biologically plausible assumptions? If so, under what conditions would learning be possible? What changes in the neural activity brings about the output when an input is applied?

2 Model

In modeling learning, we postulate the following two conditions that satisfy the biological requirements of the brain: (i) The learning process should not require detailed error information. In other words, the number of error information should be considerably smaller than the number of neurons. For example, the error back-propagation algorithm requires error information corresponding to each of the output neurons. In the case of biological learning with neural network, however, it is difficult to transmit such large amounts of information specifically to each neuron. (ii) I/O relationships should be learned one by one, i.e., the next novel I/O relationship is learned after one relationship has been learned with preserving the previously learned relationships. In contrast, in most neural network algorithms, many relationships are simultaneously learned by gradually changing the synaptic strength until all the relationships are memorized.

We introduce a layered network model consisting of input, hidden, and output layers along with a reinforcement learning such algorithm as the associative reward-penalty (ARP)(Fig.1A) [4], so that this model satisfy the above-mentioned conditions. In this model, several I/O relationships are learned one by one with only a single error signal which is given as the distance between the activity pattern of the output neurons and a prescribed target pattern. During the learning process, the synaptic strength varies in accordance with the Hebbian and anti-Hebbian rules, switched depending on the magnitude of the error signal.

To be specific, we adopt the following model with N neurons in each layer. Three types of synapses are considered: forward synapses (FSs), backward synapses (BSs), and mutually inhibiting intralayer synapses (ISs). FSs connect the neurons in the input layer to those in the hidden layer and the neurons in the hidden layer to those in the output layer. BSs connect the neurons in the output layer to those in the hidden layer, while ISs connect the neurons within a given layer (hidden layer or output layer).

The neural activity in the input layer is determined by an input pattern I , a vector whose element takes the value 0 or 1; the magnitude of the vector (input strength) is η (Eq.1). The neural activities in the other layers change, as shown by the rate coding model (Eq.2):

$$x_i = \eta I_i \quad (I \in \{0, 1\}) \quad (\text{input layer}) \quad (1)$$

$$\tau^{\text{NA}} \dot{x}_i = 1/(1 + \exp(-\beta u_i + \theta)) - x_i \quad (\text{the other layers}) \quad (2)$$

where x_i is the firing rate of a neuron i , and u_i is the input current applied to each neuron i . The input current is given by $u_i^{\text{hid}} = \sum_{j=1}^N J_{ij}^{\text{FS}} x_j^{\text{in}} + \sum_{j=1}^N J_{ij}^{\text{BS}} x_j^{\text{out}} +$

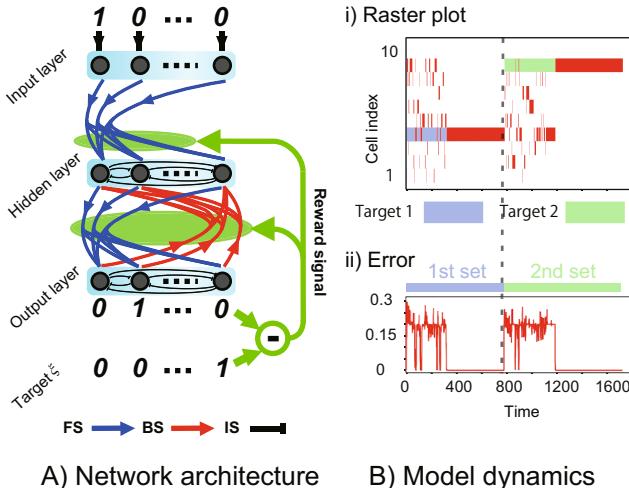


Fig. 1. A) Schematic representation of the network architecture of our model. B) Dynamics of neural activities during the learning of two input-output (I/O) relationships. I/O relationships are learned in the search phase by the anti-Hebbian rule ($0 < t < 350, 800 < t < 1200$) and in the stabilization phase by the Hebbian rule ($350 < t < 800, 1200 < t < 1700$). As initial conditions for the network, we set $\tau^{\text{NA}} = 1$, $\tau^{\text{BS}} = 8$, and $\tau^{\text{FS}} = 64$ and assign the synaptic strength a random value between 0 and 1, except in the case of the ISs. (i) Raster plot of neurons in the output layer. Red bar represents the high activity of each neuron ($x_i > 0.9$). (ii) Time series of amplitude of the error signal E between the output and the target patterns. Color bar above the time series represents each set of (I/O) relationships.

$\sum_{j \neq i} J_j^{\text{IS}} x_j^{\text{hid}}$ for the neurons in the hidden layer and $u_i^{\text{out}} = \sum_{j=1}^N J_{ij}^{\text{FS}} x_j^{\text{hid}} + \sum_{j \neq i} J_j^{\text{IS}} x_j^{\text{out}}$ for the neurons in the output layer. Here, $J_{ij}^{\text{FS}}(J_{ij}^{\text{BS}})$ is the strength of the forward (backward) synapse from a presynaptic neuron j to a postsynaptic neuron i . J^{IS} is the strength parameter for the mutually inhibiting IS; this parameter assumes a fixed and identical value for all synapses. The parameters are set at $\tau^{\text{NA}} = 1$, $\beta = 42$, $\theta = 2.5$, $\eta = 1.0$, $J^{\text{IS}} = -1.0$, and $N = 10$.

For each input pattern, we prescribe a target pattern ξ as an N -dimensional vector whose element takes the value 0 or 1. Sparse input and target patterns, in which only one neuron is activated, are chosen. By describing the neural activity in the output layer as the N -dimensional vector X^{out} , the learning task moves the error $E = |X^{\text{out}} - \xi|^2/N$ closer to zero, as measured by the Euclidean norm.

Synaptic plasticity is necessary for achieving the learning in a neural network. For simplicity, we maintain the strength of the ISs constant and vary the strengths of the FSs and BSs. In accordance with the Hebb scheme and ARP [4], we assume that the synaptic dynamics depends on the activities of the pre- and postsynaptic neurons as well as on the reward signal R determined by the error signal E , as

$$\tau^p \dot{J}_{ij}^p = R^p(x_i - r)x_j \quad (J \geq 0) \quad (p = \text{FS or BS}) \quad (3)$$

Here, r is the spontaneous firing rate (r is set at 0.1). The synaptic plasticity in this model has two characteristic features as follows.

(i) *Plasticity switched by the error:*

As mentioned earlier, the sign of R determines the synaptic plasticity. The sign of R changes with the magnitude of E such that

$$R^{\text{FS}} = \begin{cases} 1 & \text{for } E \leq \epsilon \\ -1 & \text{for } E > \epsilon \end{cases} \quad R^{\text{BS}} = \begin{cases} 0 & \text{for } E \leq \epsilon \\ -1 & \text{for } E > \epsilon \end{cases} \quad (4)$$

When the output pattern is close to (distance from) the target pattern, i.e., $E \leq (>)\epsilon$, the synaptic plasticity follows the Hebbian (anti-Hebbian) rule, as derived by substituting Eq 4 to Eq 3. The Hebbian (anti-Hebbian) rule stabilizes (destabilizes) the ongoing neural activity. In this manner, the error switches the synaptic plasticity between the Hebbian and anti-Hebbian rules. Note that under Hebbian rule, only the strength of the FS varies so that memories of the I/O relationships are embedded in the FSs.

(ii) *Multiple time scales:*

In most neural network studies, only two time scales are considered: One for neural activities and the other for the synaptic plasticity. Considering variety of timescales of the synaptic plasticity, we introduce two time scales for the plasticity of the FSs and BSs, τ^{FS} and τ^{BS} in Eq 3, respectively. As will be shown later, I/O relations are successfully memorized when the difference between the time scales is appropriate.

3 Results and Analysis

3.1 Learning Process

We first show that this model can be used to learn I/O relationships; the neural activity is varied when searching the target and stabilized when matching the target (Fig. 1B). When the error is large, the present neural activity becomes unstable by the anti-Hebbian rule and hence the neural dynamics iterate among different patterns ($0 < t < 350$). The target pattern is searched during this iteration. We term this period the search phase in what follows. At $t \sim 350$, the magnitude of the error reduces to a sufficient extent when the output dynamics are within the neighborhood of ϵ . Once this occurs, the neural activity is stabilized as per the Hebbian rule, and the output activity remains close to the target ($350 < t < 800$). Thus, the activated synapses are continuously strengthened until a new target is presented. This period is called the stabilization phase. At $t \sim 800$, we switch a new input and the corresponding target pair to learn this. At that time, the distance between the output pattern and the target pattern increases again, and therefore, the searching process progresses on the basis of the anti-Hebbian rule ($800 < t < 1200$). In this manner, the neural activity

can reach the target by switching between anti-Hebbian and Hebbian rules alternately for synaptic plasticity, depending on the error. Now, we successively provide new input-target pairs after the time interval of stabilization phase, which is sufficiently long for learning an I/O relationship.

3.2 Memories as Bifurcations

The learning process changes the flow structure in neural dynamical system both in the presence and the absence of external inputs. We examine the typical orbits of neural activity without any input and changes of such orbits with inputs in a dynamical system with a fixed synaptic strength after each learning step. Figure 2 shows examples of typical orbit in the attractor, as shown by the results of learning four and seven I/O relations. After four I/O relations are learned (Fig. 2A(i)), the neural activity in the output layer in the absence of any input is itinerant over three patterns that are close to three of the target patterns until

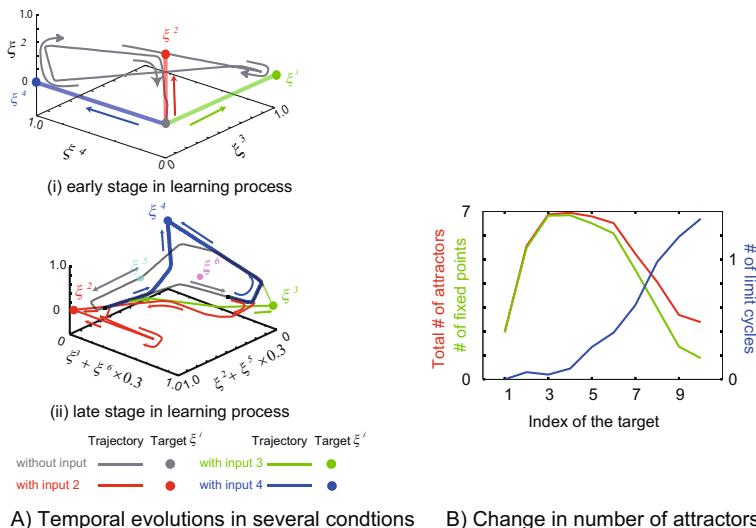


Fig. 2. Analysis of changes in the flow structure of the spontaneous neural activity during the learning process. A) Temporal evolution of neural activity in the output layer. Neural activity in the output layer is projected from the N -dimensional space consisting of neural activities in the output layer to three-dimensional space by obtaining the product of the output activity and the target pattern(s). (i) Neural dynamics after learning four I/O relationships. Each axis represents the product of neural activity and the corresponding target pattern. (ii) Neural dynamics after learning seven I/O relationships. Each axis represents the product of the neural activity and corresponding combined target patterns. B) Change in the number of attractors during the learning process. Numbers of fixed-point attractors (green line) and limit-cycle attractors (blue line) and the total number of attractors (red line) in the absence of inputs as a function of the number of learning steps, i.e., number of learned targets, are plotted.

the neural activity converges to a fixed point. Figure 2A(ii) shows an example of a limit-cycle attractor in the absence of inputs after seven I/O relations are learned, such that the orbit in the attractor (not a transient orbit as shown in Fig. 2A(i)) itinerates over the targets in the cyclic order 2, 6, 3, 4, 5. In both cases, these target patterns do not exist as fixed-point attractors without inputs. In application of the input, the fixed point and/or the limit cycle collapse and the corresponding target pattern change to a stable fixed-point attractor. Hence, this network memorizes these I/O relations as bifurcations. The number of memories varies through the learning process. Generally, as the learning progresses further, the number of memories increases in adequate time scales and that of fixed-point attractors begins to decrease, as these attractors are replaced by one or more limit-cycle attractors (Fig. 2B).

3.3 Memory Capacity: Dependence on Time Scales for Synaptic Plasticity

In this manner, our model can memorize I/O relationships successively up to some limit. The maximal number of memorized I/O relationships through learning process gives a memory capacity. This capacity depends on three time scales: τ^{NA} for changes in the neural activity, τ^{BS} and τ^{FS} for the plasticity of the BSs and FSs, respectively. From extensive numerical simulations, we found that the memory capacity reaches the maximal possible number N under the sparse coding case, when the condition $\tau^{NA} \ll \tau^{BS} \ll \tau^{FS}$ is satisfied. This implies that with a single synaptic time scale ($\tau^{FS} = \tau^{BS}$) as in the case of usual learning models, the capacity is not high.

To explain this time scale relationship, we study the synaptic dynamics during the search phase. In the phase, the output activity may come close to one of the previously learned target patterns. Since this pattern differs from the current target pattern, the attraction to it may be destroyed by the anti-Hebbian rule. In general, the longer the output pattern stays close to a state corresponding to a previously learned target pattern, the stronger is the destabilization of the state. Hence, the degree of destabilization of the previous memory increases with the residence time of the pattern on the corresponding state. As shown in Fig. 3, the residence time as a function of τ^{BS} decreases to a minimum at the τ^{BS} of the maximum memory capacity and increases as τ^{BS} approaches τ^{NA} or τ^{FS} . This can explain the dependence of the memory capacity on τ^{BS} .

Now we discuss the origin of this behavior of residence time. First, τ^{FS} determines the time scale of the memory decay, because the memory information is embedded in the FSs (Eq. 4). Second, a smaller value of either τ^{FS} or τ^{BS} determines the time scale of the search phase, because the search for the target is possible only on the basis of the change in the flow structure by the anti-Hebbian rule. Since the search phase should be sufficiently smaller than the memory decay time, $\tau^{BS} \ll \tau^{FS}$ is required to preserve the previous memory during the search phase. Third, τ^{NA} determines the time scale for the neural dynamics under given fixed flow structure. If this value were larger than or of the same order of τ^{BS} , the phase structure would be modified before the neural activity change,

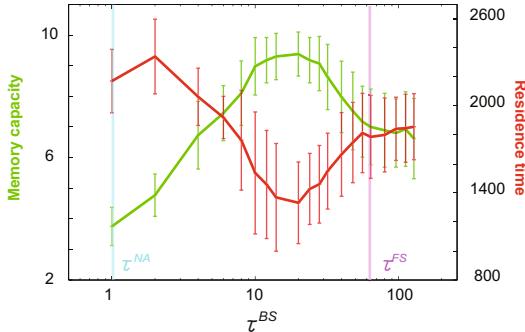


Fig. 3. Memory capacity (green) and residence time (red) as a function of the time scale of backward synapse τ^{BS} . The other two time scales τ^{NA} and τ^{FS} are fixed at $\tau^{NA} = 1$ and $\tau^{FS} = 64$. See text for the definition of capacity and residence time. Computed from the average over 100 learning processes for each τ^{BS} . Error bars indicate standard deviations.

so that the approach to the target pattern would be hindered. Hence $\tau^{NA} \ll \tau^{BS}$ is required. Accordingly the relationship $\tau^{NA} \ll \tau^{BS} \ll \tau^{FS}$ is required for the effective search for a new target without destroying the previous memory.

4 Discussion

In the present paper, we have proposed a novel dynamical systems model for the memory, in which the learning process shapes the “appropriate” flow structure of spontaneous neural dynamics, through successive presentations of inputs and the corresponding outputs. Memory recall is achieved as a result of the bifurcation of neural dynamics from spontaneous activity to an attractor that matches the target pattern induced by the external input. This bifurcation viewpoint is supported by, for example, a recent experimental study [3], in which the neural dynamics of the olfactory system of insects were studied in the presence and absence of odor stimuli. Here, we discuss how two above-mentioned features in our model can be implemented in our brain.

(i) Plasticity switched by the error:

In our brain, there exist several neural modulators such as dopamine, serotonin, norepinephrine, and acetylcholine. In particular, dopamine modulates the synaptic plasticity at the hetero-synaptic [5] and is projected to the cerebral cortex broadly. These natures correspond to the synaptic dynamics determined by the product between the reward, the activities of pre- and postsynaptic neurons and that regulated by the error signal globally. In addition, the activity of dopamine neuron producing dopamine is related to evaluation how consistent the response is with the request [6]. Hence, in our brain, the switch between positive and negative plasticity corresponding to that between Hebbian and anti-Hebbian rules can be regulated by value evaluation through the concentration of dopamine.

(ii) *Multiple time scales:*

Because the synaptic plasticity is proportional to the product between the reward, the activities of pre- and postsynaptic neurons, the difference in the sensitivity of the reward signal or the activities of pre- and postsynaptic neurons controls that in the time scale of the synaptic plasticity effectively. In our brain, this sensitivity can be interpreted as the number and/or the type of receptors of neural modulators and neurotransmitters.

To achieve the maximum number of memorized patterns, a proper relationship has to be satisfied among the time scales of the changes in the neural activity and of the plasticity of the FSs and the BSs. By above discussion, the requested relationships for successful learning can be implemented by the distribution of neurons with adequate numbers and/or type of receptors in our brain.

According to our idea “memories as bifurcations”, the neural dynamics in the presence and absence of different inputs are distinct and separated because of bifurcation, which stabilizes distinct memorized patterns. In contrast to the view “memories as attractors”, our interpretation allows for the coherent discussion of diverse dynamics depending on the applied inputs. In fact, spontaneous neural activity is recently reported, which shows itinerancy over several states that are stabilized by inputs [7]. This gives remarkable agreement with our results. This bifurcation against the input strength will be experimentally confirmed by measuring the neural activity depending on the external stimuli.

References

1. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79, 2554–2558 (1982)
2. Luczak, A., Bartho, P., Marguet, S.L., Buzsaki, G., Harris, K.D.: Sequential structure of neocortical spontaneous activity in vivo. Proc. Natl. Acad. Sci. USA 104, 347–352 (2007)
3. Ofer, M., Laurent, G.: Transient Dynamics versus Fixed Points in Odor Representations by Locust Antennal Lobe Projection Neurons. Neuron 48, 661–673 (2005)
4. Barto, A.G., Sutton, R.S., Brouwer, P.S.: Associative Search Network. Biol. Cybern. 40, 201–211 (1981)
5. Jay, T.M.: Dopamine: a potential substrate for synaptic plasticity and memory mechanism. Prog. Neurobiol. 69, 375–390 (2003)
6. Reynolds, J.N.J., Hyland, B.I., Wickens, J.R.: A cellular mechanism of reward-related learning. Nature 413, 67–70 (2001)
7. Kenet, T., Bibitchkov, D., Tsodyks, M., Grinvald, A., Arieli, A.: Spontaneously emerging cortical representations of visual attributes. Nature 425, 954–956 (2003)

Towards Spatio-Temporal Pattern Recognition Using Evolving Spiking Neural Networks

Stefan Schliebs, Nuttapod Nuntalid, and Nikola Kasabov

Auckland University of Technology, KEDRI, New Zealand

sschlieb@aut.ac.nz

www.kedri.info

Abstract. An extension of an evolving spiking neural network (eSNN) is proposed that enables the method to process spatio-temporal information. In this extension, an additional layer is added to the network architecture that transforms a spatio-temporal input pattern into a single intermediate high-dimensional network state which in turn is mapped into a desired class label using a fast one-pass learning algorithm. The intermediate state is represented by a novel probabilistic reservoir computing approach in which a stochastic neural model introduces a non-deterministic component into a liquid state machine. A proof of concept is presented demonstrating an improved separation capability of the reservoir and consequently its suitability for an eSNN extension.

1 Introduction

The desire to better understand the remarkable information processing capabilities of the mammalian brain has recently led to the development of more complex and biologically plausible connectionist models, namely spiking neural networks (SNN). See *e.g.* [4] for a comprehensive standard text on the material. These models use trains of spikes as internal information representation rather than continuous variables. Nowadays, many studies attempt to use SNN for practical applications, some of them demonstrating very promising results in solving complex real world problems. An evolving spiking neural network (eSNN) architecture was proposed in [16]. The eSNN belongs to the family of Evolving Connectionist Systems (ECoS), which was first introduced in [2]. ECoS based methods represent a class of constructive ANN algorithms that modify both the structure and connection weights of the network as part of the training process. Due to the evolving nature of the network and the employed fast one-pass learning algorithm, the method is able to accumulate information as it becomes available, without the requirement of retraining the network with previously presented training data.

The eSNN classifier learns the mapping from a single data vector to a specified class label. This behavior is very suitable for the classification of time-invariant data. However, many of today's data volumes are continuously updated adding an additional time dimension to the data sets. The classification of spatio-temporal patterns is a great challenge for data mining methods. Many data vectors are sequentially presented to an algorithm which in turn learns the mapping of this sequence to a given class label. In its current form, eSNN does not allow the classification of spatio-temporal data.

In this paper an extension of eSNN is proposed that enables the method to process spatio-temporal information. The general idea is to add an additional layer to the network architecture that transforms the spatio-temporal input pattern into a single high-dimensional network state. The mapping of this intermediate state into a desired class label can then be learned by the one-pass learning algorithm of eSNN. We present here an initial experimental analysis demonstrating the feasibility of the proposed extension in principle and leave a more comprehensive analysis for future studies.

2 Extending eSNN for Spatio-Temporal Pattern Recognition

The eSNN classification method is built upon a simplified integrate-and-fire neural model first introduced in [14] that mimics the information processing of the human eye. We refer to [12][11] for a comprehensive description and analysis of the method. The proposed extension of eSNN is illustrated in Figure 1. The novel parts in the architecture are indicated by the highlighted boxes. We outline the working of the method by explaining the diagram from left to right.

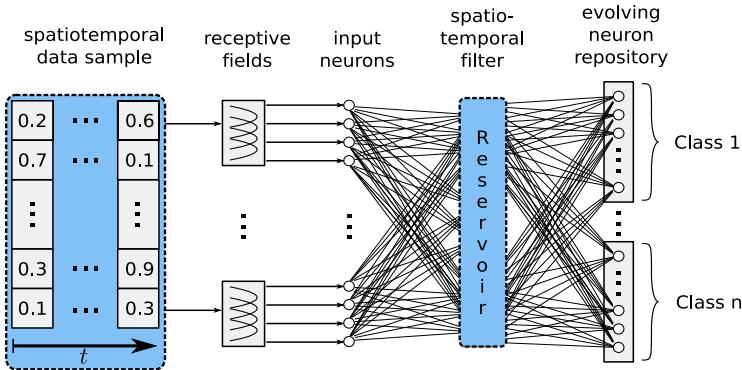


Fig. 1. Architecture of the extended eSNN capable for processing spatio-temporal data. The dashed boxes indicate novel parts in the original eSNN architecture.

Spatio-temporal data patterns are presented to the system in form of an ordered sequence of real-valued data vectors. In the first step, each real-value of a data vector is transformed into a spike train using a population encoding. This encoding distributes a single input value to multiple neurons. Our implementation is based on arrays of receptive fields as described in [1]. Receptive fields allow the encoding of continuous values by using a collection of neurons with overlapping sensitivity profiles. In [1] the role of the encoding was investigated and suitable parameter configurations were suggested.

As a result of the encoding, input neurons spike at predefined times according to the presented data vectors. The input spike trains are then fed into a *spatio-temporal filter* which accumulates the temporal information of all input signals into a single high-dimensional intermediate state. We elaborate on the specifics of this filter in the next section. The one-pass learning algorithm of eSNN is able to learn the mapping of the

intermediate state into a desired class label. The learning process successively creates a repository of trained output neurons during the presentation of training samples. For each training sample a new neuron is trained and then compared to the ones already stored in the repository. If a trained neuron is considered to be too similar (in terms of its weight vector) to the ones in the repository (according to a specified similarity threshold), the neuron will be merged with the most similar one. Otherwise the trained neuron is added to the repository as a new output neuron. The merging is implemented as the (running) average of the connection weights, and the (running) average of the two firing threshold. We refer to [12] for a more detailed description of the employed learning in eSNN and to [11] for a comprehensive analysis of all parameters involved in the training process.

In this study, the implementation of the spatio-temporal filter employs concepts of the reservoir computing paradigm [15]. The reservoir is represented by a large recurrent neural network whose topology and connection weight matrix is fixed during simulation. The function of the reservoir is to project the network inputs into a high-dimensional space in order to enhance their separability. Then, an external classification method, *i.e.* a readout function, maps the reservoir state into a class label. Due to the use of recurrent networks, the state of the reservoir can incorporate temporal information present in the input signals. Thus, the reservoir approach is very suitable to process spatio-temporal data.

A reservoir based on leaky integrate-and-fire (LIF) neurons was proposed in [9] and is commonly referred to as a Liquid State Machine (LSM). LSMs have attracted a lot of research interest; see [13] for a review. In [3][7] it was argued that the LSM approach is biologically very plausible, since some parts of the mammalian brain might act as a liquid generator while other brain areas learn how to interpret the perturbations of the liquid caused by external sensory stimuli. From this viewpoint, LSM models mimic brain-like information processing and their analysis might not only lead to very powerful computational tools, but may also provide further insights into the functioning of the mammalian brain.

2.1 A Spatio-Temporal Filter Based on a Probabilistic Liquid State Machine

For the extension of eSNN, we propose a *probabilistic* LSM which replaces the deterministic LIF neurons of a traditional LSM with stochastic neural models. This approach is motivated by the fact that biological neurons exhibit significant stochastic characteristics which, we believe, have to be taken into account when modelling a brain-like information processing system. Probabilistic neural models have been proposed in many studies, *e.g.* in the form of dynamic synapses [10], the stochastic integration of the post-synaptic potential [4] and stochastic firing thresholds [2], but also in [8] where the spike propagation and generation are defined as stochastic processes.

As an initial study, we employ some very simple probabilistic extensions of the LIF model in the probabilistic LSM. These stochastic models are well-known and are comprehensively described in [4]. Based on a brief summary of the LIF neural model, we explain the probabilistic extensions in the next paragraphs.

The LIF neuron is arguably the best known model for simulating spiking networks. It is based on the idea of an electrical circuit containing a capacitor with capacitance C

and a resistor with a resistance R , where both C and R are assumed to be constant. The model dynamics are then described by the following differential equation:

$$\tau_m \frac{du}{dt} = -u(t) + R I(t) \quad (1)$$

The constant τ_m is called the membrane time constant of the neuron. Whenever the membrane potential u crosses a threshold ϑ from below, the neuron fires a spike and its potential is reset to a resting potential u_r . It is noteworthy that the shape of the spike itself is not explicitly described in the traditional LIF model. Only the firing times are considered to be relevant.

We define a *stochastic reset* (SR) model that replaces the deterministic reset of the potential after spike generation with a stochastic one. Let $t^{(f)} : u(t^{(f)}) = \vartheta$ be the firing time of a LIF neuron, then

$$\lim_{t \rightarrow t^{(f)}, t > t^{(f)}} u(t) = \mathcal{N}(u_r, \sigma_{SR}) \quad (2)$$

defines the reset of the post-synaptic potential. $\mathcal{N}(\mu, \sigma)$ is a Gaussian distributed random variable with mean μ and standard deviation σ . Variable σ_{SR} represents a parameter of the model.

We define two stochastic threshold models that replace the constant firing threshold ϑ of the LIF model with a stochastic one. Once more, let $t^{(f)}$ be the firing time of a LIF neuron. In the *step-wise stochastic threshold* (ST) model, the dynamics of the threshold update are defined as

$$\lim_{t \rightarrow t^{(f)}, t > t^{(f)}} \vartheta(t) = \mathcal{N}(\vartheta_0, \sigma_{ST}) \quad (3)$$

Variable σ_{ST} represents the standard deviation of $\vartheta(t)$ and is a parameter of the model. According to Eq. [3] the threshold is the outcome of a ϑ_0 -centered Gaussian random variable which is sampled whenever the neuron fires. We note that this model does not allow spontaneous spike activity. More specifically, the neuron can only spike at time $t^{(f)}$ when also receiving a pre-synaptic input spike at $t^{(f)}$. Without such a stimulus a spike output is not possible.

The *continuous stochastic threshold* (CT) model updates the threshold $\vartheta(t)$ continuously over time. Consequently, this model allows spontaneous spike activity, *i.e.* a neuron may spike at time $t^{(f)}$ even in the absence of a pre-synaptic input spike at $t^{(f)}$. The threshold is defined as an Ornstein-Uhlenbeck process [6]:

$$\tau_\vartheta \frac{d\vartheta}{dt} = \vartheta_0 - \vartheta(t) + \sigma_{CT} \sqrt{2\tau_\vartheta} \xi(t) \quad (4)$$

where the noise term ξ corresponds to Gaussian white noise with zero mean and unit standard deviation. Variable σ_{CT} represents the standard deviation of the fluctuations of $\vartheta(t)$ and is a parameter of the model. We note that $\vartheta(t)$ has an overall drift to a mean value ϑ_0 , *i.e.* $\vartheta(t)$ reverts to ϑ_0 exponentially with rate τ_ϑ , the magnitude being in direct proportion to the distance $\vartheta_0 - \vartheta(t)$.

The following parameters were used to configure the neural models. We set the membrane time constant $\tau_m = 10\text{ms}$, the resting potential $u_r = 0\text{mV}$, the firing threshold

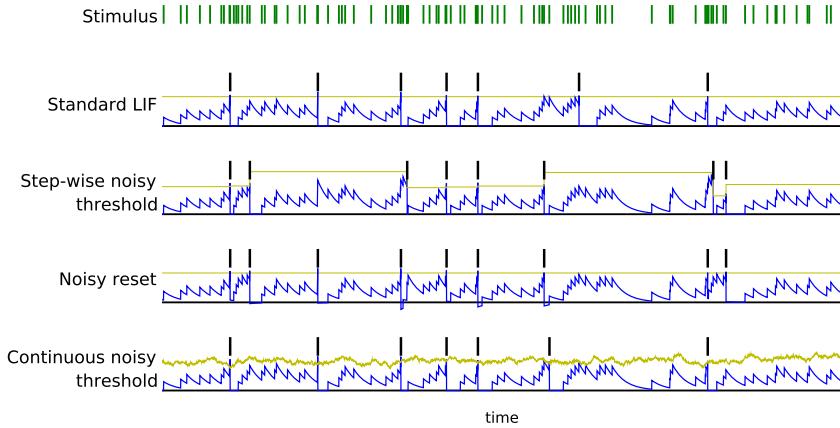


Fig. 2. Evolution of the post-synaptic potential $u(t)$ and the firing threshold $v(t)$ over time (blue (dark) and yellow (light) curves respectively) recorded from a single neuron of each neural model. The input stimulus for each neuron is shown at the top of the diagram. The output spikes of each neuron are shown as thick vertical lines above the corresponding threshold curve.

$\vartheta_0 = 10\text{mV}$, the after-spike refractory period $\Delta^{\text{abs}} = 5\text{ms}$, the standard deviation of reset fluctuations $\sigma_{SR} = 3\text{mV}$, the standard deviation of step-wise firing threshold $\sigma_{ST} = 2\text{mV}$ and the standard deviation of continuous firing threshold $\sigma_{CT} = 1\text{mV}$.

The dynamics of the four models are presented in Figure 2. For each model a single neuron is shown that is stimulated by a random spike train generated by a Poisson process with mean rate 150Hz. Both the evolution of the post-synaptic potential $u(t)$ and the evolution of the firing threshold $v(t)$ are recorded and shown in the figure. We note the step-wise and the continuous update of the two threshold models and the stochastic reset of the reset model. Due to the stochastic dynamics each probabilistic model displays a different spike output pattern compared to the deterministic LIF neuron.

In order to investigate the variation of the neural response of each stochastic model, we repeat the previous scenario 1000 times for each model and compute the corresponding peristimulus time histograms (PSTH). A PSTH is a common neuro-scientific tool that creates a histogram of spikes occurring in a raster plot. A frequency vector is computed which is normalized by dividing each vector element by the number of repetitions and by the size of a time bin (1ms here). Additionally, the raw PSTH is smoothed using Gaussian smoothing with a window width of 10ms. Figure 3 presents a raster plot of the neural response for each stochastic model. The impact of the non-deterministic neural dynamics is clearly visible in the diagrams. We also note that some of the spikes occur in every single repetition resulting in sharp peaks in the PSTH. In this paper, we call these spikes *reliable*. Reliable spikes are of particular interest in the context of a training algorithm that learns to map a reservoir response to a desired class label. We elaborate on this point as part of the experiments presented below.

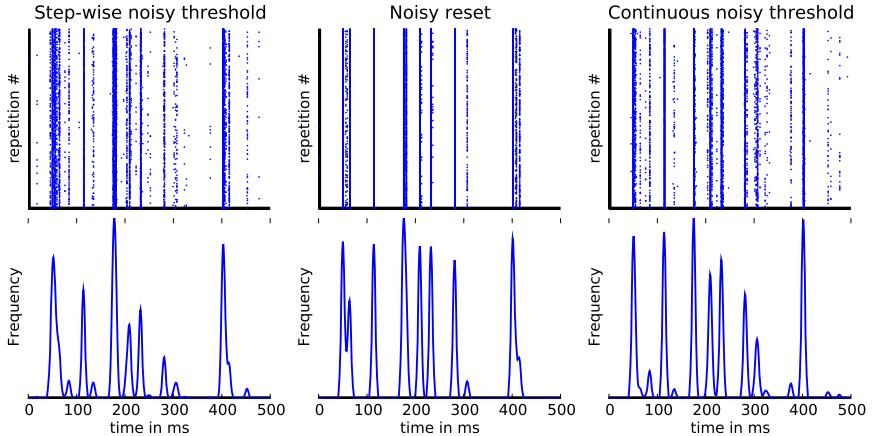


Fig. 3. Raster plot of the neural response of a single stochastic neuron recorded in 1000 repetitions (top row). The bottom row presents the corresponding smoothed PSTH for each raster plot. Each column corresponds to a different stochastic neural model as indicated by the plot titles.

3 Experiments

As a proof of concept of the suitability of the proposed probabilistic spatio-temporal filter, we demonstrate here its separation ability. Our experiments are inspired by the study presented in [5] where the authors investigate different neural models in the context of a LSM. It was shown that the choice of a particular neural model impacts the separation ability of the liquid significantly.

In our experiments we construct a liquid having a small-world inter-connectivity pattern as described in [9]. A recurrent SNN is generated by aligning 1000 neurons in a three-dimensional grid of size $10 \times 10 \times 10$. Two neurons A and B in this grid are connected with a connection probability

$$P(A, B) = C \times e^{\frac{-d(A, B)}{\lambda^2}} \quad (5)$$

where $d(A, B)$ denotes the Euclidean distance between two neurons and λ corresponds to the density of connections which was set to $\lambda = 2$ in all simulations. Parameter C depends on the type of the neurons. We discriminate into excitatory (ex) and inhibitory (inh) neural types resulting in the following parameters for C : $C_{ex-ex} = 0.3$, $C_{ex-inh} = 0.2$, $C_{inh-ex} = 0.4$ and $C_{inh-inh} = 0.1$. The network contained 80% excitatory and 20% inhibitory neurons. All parameter values are directly adopted from [5].

Four recurrent SNN are generated each employing a different neural model. All networks have the same network topology and the same connection weight matrix. The networks are then stimulated by two input spike trains independently generated by a Poisson process with a mean rate of 200Hz. The response of each network was recorded in 25 independent repetitions.

The averaged response of the networks is presented in Figure 4. The two top rows of diagrams show the average raster plots of the spike activity for both stimuli. The

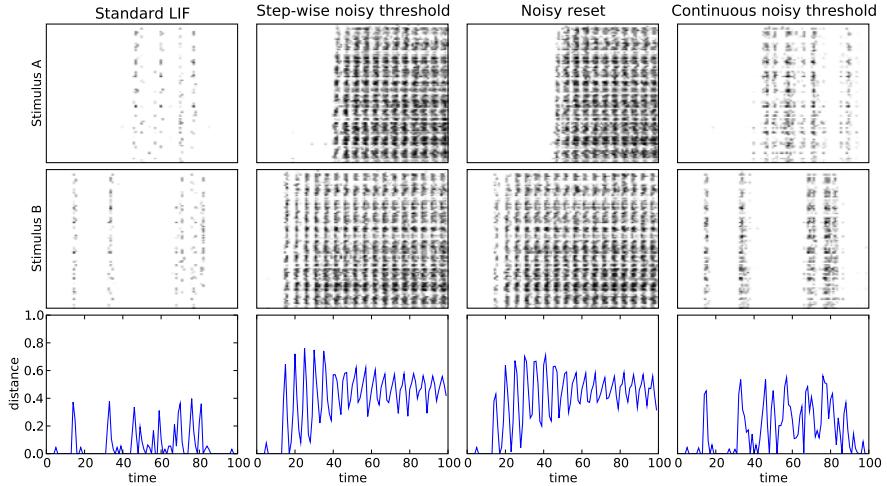


Fig. 4. Averaged spike response of the reservoirs using different neural models to two independent input stimuli recorded in 25 independent runs (top two rows). The averaged normalized Euclidean differences in each time bin (bin size 1ms) between the responses are presented in the bottom row.

darker the shade in these plots, the more likely the corresponding neuron was observed to spike at the given time bin in 25 runs. The shade is white for time bins in which no neural activity was observed in any run. Each time bin has a size of 1ms. Similar to the raster plots in Figure 3 we notice some reliable spikes in the response corresponding to the very dark shades in the plots.

The bottom row of diagrams in Figure 4 presents the averaged normalized Euclidean distances between the two responses for each time bin. The very same distance measure was used in [5] to evaluate differences in response patterns. We note the comparably low separation ability of the deterministic LIF model which confirms the findings in [5]. The results indicate that stochastic models have the potential to significantly increase the separation ability of the reservoir. However, additional experimental analysis is required to provide strong statistical evidence for this claim.

4 Conclusion and Future Directions

This study has proposed an extension of the eSNN architecture that enables the method to process spatio-temporal data. The extension projects a spatio-temporal signal into a single high-dimensional network state that can be learned by the eSNN training algorithm. We have proposed a novel probabilistic LSM approach that was subjected to an initial feasibility analysis. It was experimentally demonstrated that probabilistic neural models are principally suitable reservoirs that have furthermore the potential to enhance the separation ability of the system.

Future studies will investigate the characteristics of the extended eSNN on typical benchmark problems. Furthermore, the method is intended to be used to classify

real-world EEG data sets. Particularly interesting are also different stochastic neural models, such as the dynamic stochastic synapses discussed in [10].

References

1. Bohte, S.M., Kok, J.N., Poutré, J.A.L.: Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* 48(1-4), 17–37 (2002)
2. Clopath, C., Jolivet, R., Rauch, A., Lüscher, H.R., Gerstner, W.: Predicting neuronal activity with simple models of the threshold type: Adaptive exponential integrate-and-fire model with two compartments. *Neurocomput.* 70(10-12), 1668–1673 (2007)
3. Destexhe, A., Contreras, D.: Neuronal computations with stochastic network states. *Science* 314(5796), 85–90 (2006)
4. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (2002)
5. Grzyb, B.J., Chinellato, E., Wojcik, G.M., Kaminski, W.A.: Which model to use for the liquid state machine? In: IJCNN 2009: Proceedings of the 2009 international joint conference on Neural Networks, pp. 1692–1698. IEEE Press, Piscataway (2009)
6. van Kampen, N.G.: Stochastic Processes in Physics and Chemistry. North-Holland, Amsterdam (2007)
7. Kasabov, N.: The ECOS framework and the ECO learning method for evolving connectionist systems. *JACIII* 2(6), 195–202 (1998)
8. Kasabov, N.: To spike or not to spike: A probabilistic spiking neuron model. *Neural Networks* 23(1), 16–19 (2010)
9. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation* 14(11), 2531–2560 (2002)
10. Maass, W., Zador, A.: Dynamic stochastic synapses as computational units. In: Advances in Neural Information Processing Systems, pp. 903–917. MIT Press, Cambridge (1998)
11. Schliebs, S., Defoin-Platel, M., Kasabov, N.: Analyzing the dynamics of the simultaneous feature and parameter optimization of an evolving spiking neural network. In: International Joint Conference on Neural Networks, IEEE - INNS - ENNS. IEEE Computer Society Press, Barcelona (2010)
12. Schliebs, S., Defoin-Platel, M., Worner, S., Kasabov, N.: Integrated feature and parameter optimization for an evolving spiking neural network: Exploring heterogeneous probabilistic models. *Neural Networks* 22(5-6), 623–632 (2009)
13. Schrauwen, B., Verstraeten, D., Campenhout, J.V.: An overview of reservoir computing: theory, applications and implementations. In: Proceedings of the 15th European Symposium on Artificial Neural Networks, pp. 471–482 (2007)
14. Thorpe, S.J.: How can the human visual system process a natural scene in under 150ms? On the role of asynchronous spike propagation. In: ESANN. D-Facto public (1997)
15. Verstraeten, D., Schrauwen, B., D’Haene, M., Stroobandt, D.: An experimental unification of reservoir computing methods. *Neural Networks* 20(3), 391–403 (2007)
16. Wysoski, S.G., Benuskova, L., Kasabov, N.K.: Adaptive learning procedure for a network of spiking neurons and visual pattern recognition. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 1133–1142. Springer, Heidelberg (2006)
17. Yamazaki, T., Tanaka, S.: 2007 special issue: The cerebellum as a liquid state machine. *Neural Netw.* 20(3), 290–297 (2007)

Real-Time Simulation of Phosphene Images Evoked by Electrical Stimulation of the Visual Cortex

Tamas Fehervari, Masaru Matsuoka, Hirotsugu Okuno, and Tetsuya Yagi

Graduate School of Engineering, Osaka University,
2-1 Yamadaoka, Suita, Osaka, 565-0871 Japan
tamas@neuron.eei.eng.osaka-u.ac.jp,
{h-okuno,yagi}@eei.eng.osaka-u.ac.jp

Abstract. Applying electrical stimulation to the visual cortex has been shown to produce dot-like visual perceptions called phosphenes. Artificial prosthetic vision is based on the concept that patterns of phosphenes can be used to convey visual information to blind patients. We designed a system that performs real-time simulation of phosphene perceptions evoked by cortical electrical stimulation. Phosphenes are displayed as Gaussian circular and ellipsoid spots on a randomised grid based on existing neurophysiological models of cortical retinotopy and magnification factor. The system consists of a silicon retina camera (analogue integrated vision sensor), desktop computer and head-mounted display.

Keywords: visual prosthesis, phosphene, artificial vision, simulation, visual cortex, silicon retina.

1 Introduction

A number of studies have shown that electrical stimulation applied to the human visual cortex evokes spot-like visual perceptions, called phosphenes [1], [2], [3]. An implanted electrode array – a cortical prosthesis – is thought to induce a pattern of phosphenes – a phosphene image – in the visual field of a blind patient according to the spatiotemporal pattern of electrical stimulations. The feasibility of such cortical prostheses is supported by previous clinical research [2]. It is important to study simulated phosphene vision in order to understand what level of visual function, e.g. object recognition, navigation, can be restored to blind patients, and what methods should be used to achieve that. In the last two decades, several studies have been conducted to simulate artificial vision using conventional digital image processing systems. Many of these studies, however, employed simplified models for phosphene generation, such as uniform phosphene shape and size over the image and regular hexagonal or rectangular phosphene lattice [4].

Here, we aim to design an emulation system of phosphene image based on neurophysiological and clinical observations, employing a neuromorphic image sensor that is a good candidate to be incorporated in visual prostheses because of its real-time and low power computational ability to mimic neuronal representation of visual information in retinal circuits.

2 System Setup

The simulation system we present in this paper consists of the following components: image acquisition, image processing and image display. Image acquisition and pre-processing is carried out by the Intelligent Vision Sensor (Neural Image Co.), which is composed of a silicon retina and a field programmable gate array (FPGA) circuit. The original architecture of the silicon retina developed by Kameda and Yagi [5] mimics the sustained and transient channels of the vertebrate retinal circuit. The sustained output possesses a Laplacian-Gaussian-like receptive field and carries out a smoothing and contrast enhancement on the input. The transient output is sensitive to moving objects [5]. Due to its real-time processing ability, low power consumption and small size, this chip is ideal for the image sensor and pre-processing role of an artificial vision system. A 50 mm lens was used on the camera, which provided an angle of view of approximately 26°. Images were recorded at 128x128 pixel resolution and 30 Hz frame rate, and transmitted to a personal computer via USB connection (ASPECTUS-U, Neuralimage Co. Ltd., Osaka, Japan).

Image processing and phosphene image generation is performed partially on-chip by the silicon retina and then on personal computer by software developed at our laboratory in Microsoft Visual Studio 2005 (Microsoft, Redmond, WA, USA). Simulated phosphene image is generated for the right visual hemi-field and is displayed either on a computer monitor or head-mounted display (iWear VR920, VUZIX Co., NY, USA, 640x480 pixel resolution, 32° viewing angle) (Fig. 1).

Image output frequency is currently limited by software to the image acquisition frequency, thus phosphene time courses are displayed at 30 Hz frame rate.

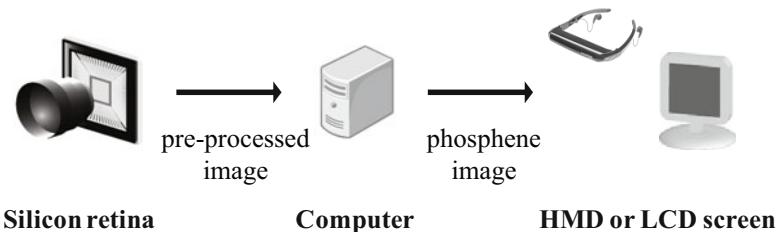


Fig. 1. Simulation System: image acquisition and pre-processing by a silicon retina-based camera, image processing done on digital computer, and image display on either computer screen or head-mounted display (HMD)

3 Emulation of Phosphene Images

The algorithm of simulated phosphene image generation described below follows steps similar to what would happen in the case of the implantation and configuration of a real cortical visual prosthesis. In the first step, a phosphene database is generated from the location of electrodes on the virtual visual cortex, describing location, size and shape of phosphenes. A similar phosphene topology map – describing the phosphenes that are elicited by cortical electrical stimulation – would have to be created for real prosthesis patients too, after the electrodes have been implanted.

The phosphene database is calculated once, prior to any simulation of cortical stimulation. The second step is responsible for real-time image processing and phosphene image generation.

3.1 Phosphene Database Calculation

3.1.1 Wedge-Dipole Model and Cortical Magnification Ratio

The Wedge-Dipole model introduced by Polimeni, Balasubramanian and Schwartz [6] provides a mapping between visual field polar coordinates and the two-dimensional visuotopic structure of several visual cortical areas in macaque monkeys and humans. This model was reduced to the visuotopic mapping of V1 area to fit our simulation system. The simplifications are discussed in detail by Matsuoka [7] and are reproduced here for convenience.

Points in the right visual hemi-field can be represented by the complex variable

$$z = re^{i\Theta} \in \mathbb{C}, 0 \leq r \leq r_{max}, -\frac{\pi}{2} \leq \Theta \leq +\frac{\pi}{2} \quad (1)$$

where r represents eccentricity and Θ represents azimuth. The *wedge map* of V1 within the left cerebral hemisphere is given by

$$z_w = re^{i\alpha\Theta} \in \mathbb{C}, \quad (2)$$

where z_w is the wedge map complex coordinate corresponding to z , and α is a parameter controlling the amount of azimuthal shear within V1. The wedge map coordinates are transformed into dipole coordinates using

$$w = \log\left(\frac{z_w+a}{z_w+b}\right) \in \mathbb{C}, \quad (3)$$

where w signifies the complex coordinate for V1 area in the contralateral visual cortex. Following values were used for the parameters: $a=0.9$, $b=180$, $\alpha=0.95$.

Phosphene size is calculated using a cortical magnification ratio that is location dependent but uniform in all directions with this formula:

$$M = \left| \frac{dw}{dz} \right| = \left| \frac{k(b-a)}{(z+a)(z+b)} \right|, \quad (4)$$

where M indicates the cortical magnification factor for a location specified by the complex coordinate z . Further details regarding this model can be found in [7].

3.1.2 Phosphene Location

Virtual electrodes are placed on a virtual representation of the visual cortex (Fig. 2A). The inverse of the Wedge-Dipole mapping is used to determine the corresponding visual field locations for each electrode. (Fig. 2B) shows the visual field location of an array of electrodes covering the entire area that corresponds to the visual hemi-field in the visual cortex seen on (Fig. 2A). Whereas rectangular or hexagonal phosphene lattices are regularly used in psychophysical studies, cortical visuotopic mapping based on neurophysiological data suggests that for electrodes arranged in a regular grid the elicited phosphenes closer to the centre of the visual field are located closer to each other and those on periphery are located sparsely [8]. This topological correspondence is supported by electrophysiological experiments on monkeys [9], [6]

and [11]. The above model is, however, rather optimistic, since it does not take into account irregularities in the cortical map and in the cortical surface and other factors that have an effect on phosphene locations. In order to represent these factors in our model, we randomly shifted phosphene locations (Fig. 2D). The formula used for calculation of visual field coordinates was described in section 3.1.1 .

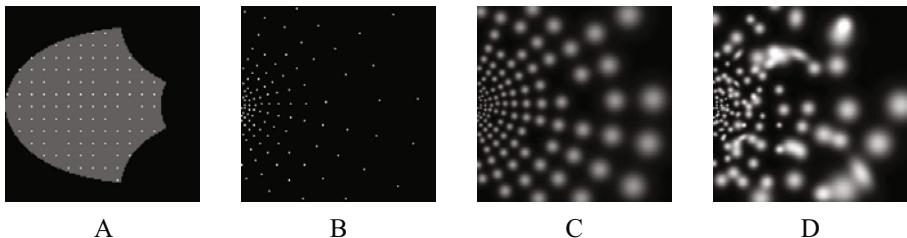


Fig. 2. Phosphene map generation steps. A) Left visual cortex flattened map. Grey: Wedge-Dipole map area corresponding to the visual hemi-field; lower eccentricities are to the left. White dots: electrode locations. Absent grid dots indicate electrode defects. B) Locations in right visual hemi-field corresponding to electrode locations. C) Phosphene sizes in the visual hemi-field. Phosphenes with larger eccentricity are larger in size. D) Phosphene locations randomised.

3.1.3 Electrode Defects

In order to account for electrodes that do not elicit phosphenes, some stimulation points are removed from further calculations. Absent grid points on (Fig. 2A-C) indicate defects. Random shift of phosphene locations has a mitigating effect on the noticeability of defects (Fig. 2D).

3.1.4 Phosphene Size

Neurophysiological studies have demonstrated that receptive field sizes are smaller at smaller eccentricities and larger on the periphery [9]. Our model assumes that the inverse is also true, namely that the same stimulation would evoke a response of smaller size when applied at a small eccentricity location than at the periphery (Fig. 2C). Calculation method was described in section 3.1.1 . Phosphene sizes were also randomised (Fig. 3A and 3B).

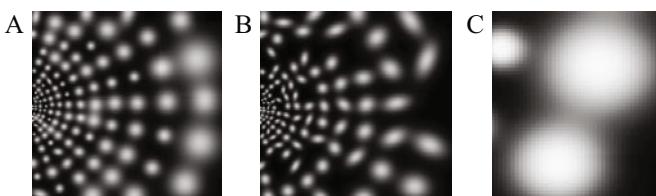


Fig. 3. Phosphene appearance. A) Circular phosphenes of randomised sizes. B) Elliptical phosphenes with random sizes and rotational angles. C) Additive drawing of overlapping phosphenes.

3.1.5 Phosphene Appearance

Phosphenes are reported to appear in a variety of shapes [4], [1], [2] and [3]. Our system displays phosphenes as circular or elliptical Gaussian spots. Very small phosphenes appear as dots. Rotation angle and axis lengths are randomised for elliptical phosphenes. Overlapping phosphene regions are rendered additively to avoid sharp edges at borders. (Fig. 3).

3.1.6 Temporal Properties of Phosphenes

In a cortical micro-stimulation study Schmidt et al. [1] reported that when the interval between two stimulation trains was above ~25 ms, two phosphenes were perceived separately. Intervals below this would produce a phosphene that appeared to be uninterrupted. In addition, they found that during long stimulation trains of one second and longer, phosphenes seemed to be becoming smaller in size but not in intensity. We implemented this latter phenomenon in a way that each phosphene's intensity is reduced by a small and randomised percentage each frame. A simplified model of flicker fusion was also realised by prolonging each phosphene's duration by one extra frame (33.3 ms) after the end of stimulation. If a new stimulation train starts on the second frame after the previous one has ended, the phosphene display is not interrupted.

4 Real-Time Calculation of Phosphene Images

Output from the silicon retina (Fig. 4D) is spatially filtered with one of the implemented filters, which produces an edge or surface detection image (Fig. 4E). This image is compared to each phosphene's detection area (Fig. 4C) to determine if a

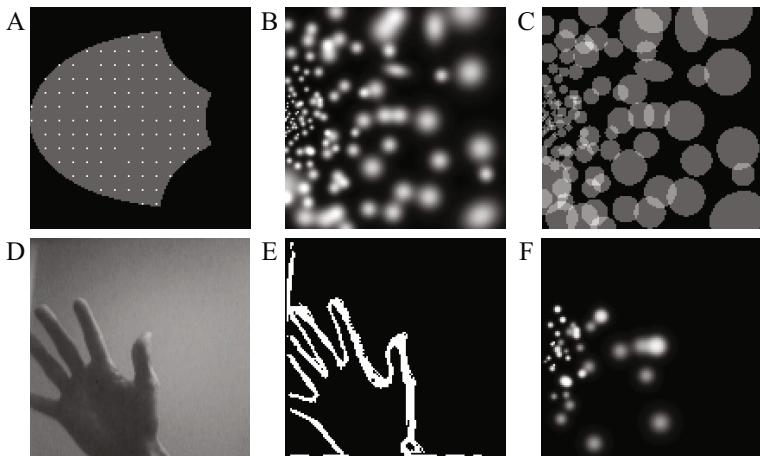


Fig. 4. Image processing steps. The upper row represents items that are calculated once, the bottom row is calculated in real time. A) Cortical map with electrode locations. B) Phosphene map. C) Detection masks. D) Input from first resistive layer. E) Sobel filter output. F) Overlaps between the spatial filter output E) and individual detection masks C) are used to calculate intensity for activated phosphenes, which are here displayed at 4 different greyscale levels.

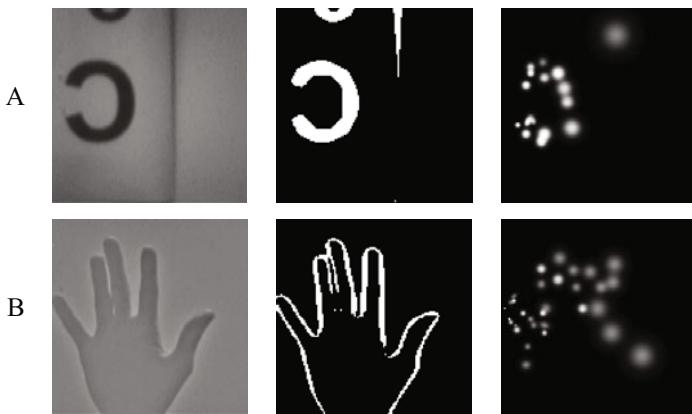


Fig. 5. Spatial filtering methods. A) Brightness threshold on first resistive layer output might be useful for reading tasks. B) Laplacian-Gaussian-like (LG-like) image thresholding filter. C) Zero-cross filter on LG-like image. Meaning of images in each row from left to right: input image, spatial filtering output, simulated phosphene image.

phosphene is active (rendered on the output) or not. Phosphene intensity is determined by how much its detection area is covered by pixels from the spatially filtered image. Intensity is converted to 2, 3, 4 or 8 levels of greyscale brightness on the output image (Fig. 4F). The following filters are implemented: Sobel (Fig. 4E), brightness thresholding (Fig. 5A), zero-cross detection on Laplacian-Gaussian (LG)-like silicon retina image (Fig. 5B) and thresholding on LG-like image. (Fig. 4F) and the rightmost images on (Fig. 5) show just a frame of the real-time output. The hand was much more recognisable in motion than on these static images.

5 Discussion

5.1 Limitations of the Model

Unlike the flattened out two-dimensional Wedge-Dipole representation, the real visual cortex has a complex topography and large part of it is hidden from easy access. This can be emulated by not placing virtual electrodes on inaccessible regions.

Without an eye-tracking device, the displayed phosphene image is not fixed on a retina position, allowing eye scanning across the entire screen. This effect is probably not very pronounced when most of the important visual information is located near the highly concentrated central visual field area, but it means that our system currently cannot reliably simulate a situation when all electrodes are in the periphery.

5.2 Future Improvements

We will conduct further tests in order to compare the performance of different filters and image processing techniques.

5.3 Similarities Observed between Phosphene Behaviour and Our Voltage-Sensitive Dye recordings

Our research group also conducts voltage-sensitive dye experiments on mouse visual cortex to investigate the effect of intracortical electrical stimulation in V1 and surrounding visual areas. Our experiment protocol is described in [10]. Observations of V1 response to train stimuli show interesting similarities with some aspects of phosphene behaviour in clinical trials.

Stimulation (cathodic-first biphasic unipolar pulse train with pulse duration 200 µm, inter-pulse interval 0, train length 300 ms, current intensity 10 µA) at 50 Hz and lower produces individual activation peaks, while 100 and above stimulation is continuous (Fig. 6). A cortical fusion frequency around 50 Hz agrees with visual prosthesis study findings in [1] and [4]. The same recording also illustrates that cortical activity can remain hundreds of milliseconds even after the simulation has ended. Similarly, sometimes phosphenes are perceived longer than stimulus duration [1], [2] and [3]. Also noticeable is a gradual decrease in signal amplitude over time, which may correspond to the perception that some phosphenes appear to recede into the distance, as reported in [1].

We will further investigate these phenomena and integrate results into our model.

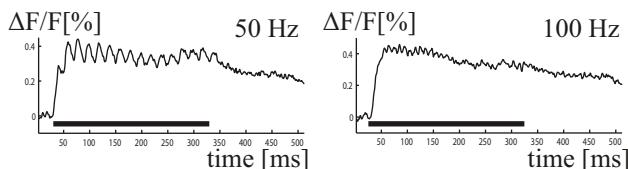


Fig. 6. Voltage-sensitive dye recordings of mouse visual cortex V1 area near the stimulation site after electrical stimulation. Individual peaks are present at 50 Hz stimulation, and the signal appears continuous at 100 Hz. Cortical activity lasts longer than the stimulus and is steadily decreasing in amplitude. Black bars indicate stimulus train length.

References

1. Schmidt, E.M., Bak, M.J., Hambrecht, F.T., Kufta, C.V., O'Rourke, D.K., Vallabhanath, P.: Feasibility of a visual prosthesis for the blind based on intracortical micro stimulation of the visual cortex. *Brain* 119, 507–522 (1996)
2. Brindley, G.S., Lewin, W.S.: The sensations produced by electrical stimulation of the visual cortex. *J. Physiol.* 196, 479–493 (1968)
3. Dobelle, W.H., Mladejovsky, M.G., Girvin, J.P.: Artificial Vision for the Blind: Electrical Stimulation of Visual Cortex Offers Hope for a Functional Prosthesis. *Science* 183, 440–444 (1974)
4. Chen, S.C., Suaning, G.J., Morley, J.W., Lovell, N.H.: Simulating prosthetic vision: I. Visual models of phosphenes. *Vision Research* 49, 1493–1506 (2009)
5. Kameda, S., Yagi, T.: An analog VLSI chip emulating sustained and transient response channels of the vertebrate retina. *IEEE Transactions on Neural Networks* 14(5), 1405–1412 (2003)
6. Polimeni, J.R., Balasubramanian, M., Schwartz, E.L.: Multi-area visuotopic map complexes in macaque striate and extra-striate cortex. *Vision Research* 46, 3336–3359 (2006)

7. Matsuoka, M., Fehervari, T., Okuno, H., Yagi, T.: Simulation of Perception Induced by Electrical Stimulation on Visual Cortex. In: Proceedings of the 3rd Brain Biocommunication Kenkyukai, pp. 5–8 (2009)
8. Horton, J.C., Hoyt, W.F.: The Representation of the Visual Field in Human Striate Cortex. *Arch. Ophthalmol.* 109, 816–824 (1991)
9. Daniel, P.M., Whitteridge, D.: The Representation of the Visual Field on the Cerebral Cortex in Monkeys. *J. Physiol.* 159, 203–221 (1961)
10. Okazaki, Y., Fehervari, T., Sawai, H., Yagi, T.: In vivo optical measurements of the activities of mouse striate and extrastriate cortices evoked by intracortical electrical stimulation. In: Okazaki, Y., Fehervari, T., Sawai, H., Yagi, T. (eds.) BPES 2010, 25th Symposium on Biological and Physiological Engineering (in Press 2010) (in Japanese)
11. Adams, D.L., Horton, J.C.: A Precise Retinotopic Map of Primate Striate Cortex Generated from the Representation of Angioscotomas. *J. Neurosci.* 23(9), 3771–3789 (2003)

An Effect of Inhibitory Connections on Synchronous Firing Assembly in the Inhibitory Connected Pulse Coupled Neural Network

Hiroaki Kurokawa, Masahiro Yoshihara, and Masato Yonekawa

Tokyo University of Technology, 1404-1 Katakura Hachioji Tokyo, Japan
hkuro@bs.teu.ac.jp

Abstract. The Pulse Coupled Neural Network (PCNN) had been proposed as a model of visual cortex and a lot of applications to the image processing have been proposed recently. Authors also have been proposed Inhibitory Connected PCNN (IC-PCNN) which shows good performances for the color image processing. In our recent study, we had been shown that the IC-PCNN can obtain successful results for the color image segmentation. In this study, we show the effect of the inhibitory connections to the characteristics of synchronous firing assembly. Here we consider that the results will be a key to find appropriate values of inhibitory connections for the image processing using IC-PCNN. In simulations, we show that the valid domains of inhibitory connections for the color image segmentation exists.

1 Introduction

The Pulse Coupled Neural Network (PCNN) was originally proposed as a model which can explain synchronous dynamics of neurons' activity in cat visual cortex [1]. The PCNN model consists of neurons and two kind of inputs, namely feeding inputs and linking inputs with leaky integrators. The linking and feeding inputs are received from neurons' receptive fields which is defined by synaptic weight and directly from the environment.

In the engineering point of view, one of the remarkable features of the PCNN is its temporal pulse coding and a synchronous assembly of firing neurons will be observed in the network in some conditions. Using these characteristics of PCNN, a lot of studies showed the applications to the problem of image processing, *e.g.* segmentation, edge enhancement, and so on [2] [3] [4].

In the studies on image processing using PCNN, almost of conventional studies had not been concerned color image. However, some of the research mentioned a color image processing [5] [6]. Also, in our previous study, we proposed the Inhibitory Connected PCNN (IC-PCNN) which is an extended model of the PCNN and showed an application to color image segmentation [7]. Our proposed IC-PCNN has three types of neuron, which can detect RGB component of the color image as observed in vivo retinal cone cell. The IC-PCNN model consists of full components which are in the PCNN and inhibitory connection among

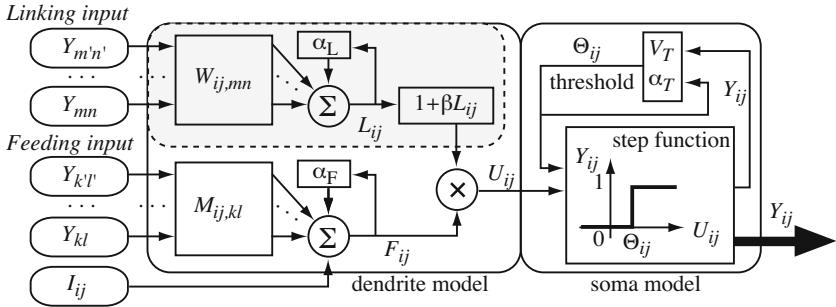


Fig. 1. The schematic model of Pulse Coupled Neural Network (PCNN) [7]

RGB neurons. To apply the IC-PCNN to the color image segmentation, set of the RGB neuron will be arranged to each pixel in the image. We had shown that IC-PCNN achieves plausible segmentation of color images in previous study [7].

On the other hand, the PCNN has a lot of parameters and the parameters in PCNN is used to be defined empirically for the engineering use. Therefore the method of parameter setting in the PCNN is still remaining as a problem to be solved. Corresponding to that problem, parameters design methods have been studied [8] [9] [10].

In this study, we describe the characteristics of the inhibitory connections to the synchronous firing assembly in the IC-PCNN. We consider that the characteristics will be a key to reveal a requirement to set up the parameters in IC-PCNN for the color image processing. In the simulation, characteristics of the inhibitory connections related to its average strength and variance are shown. The results suggest that the optimization of the inhibitory connections in the IC-PCNN is valid for the color image segmentation using IC-PCNN.

2 Color Image Processing Using IC-PCNN

In this section, we describe a configuration of IC-PCNN model [7]. The IC-PCNN is a model which is introduced the inhibitory connections to the conventional PCNN. Figure 1 shows a schematic of PCNN. The model consists of the dendrite model and the soma model. In general, the PCNN for the image processing has two dimensional structure with lattice connection among neurons and each neuron in the PCNN receives information from each corresponding pixel via feeding input. The two dimensional PCNN model is mathematically described as follows.

The internal state of the neuron ij , namely membrane potential in biological model, is given by,

$$U_{ij}(t) = F_{ij}(t)(1 + \beta_{ij}L_{ij}(t)). \quad (1)$$

Note that the indices ij denote neuron number in the two dimensional PCNN. In the PCNN model, there are two kinds of inputs which are propagated via

different connections. One is a *feeding input* and the other is a *linking input*. Each input is described as follows, respectively.

$$F_{ij}(t+1) = F_{ij}(t) \exp(-1/\alpha_F) + V_F \sum_k \sum_l M_{ij,kl} Y_{kl}(t) + I_{ij}, \quad (2)$$

$$L_{ij}(t+1) = L_{ij}(t) \exp(-1/\alpha_L) + V_L \sum_m \sum_n W_{ij,mn} Y_{mn}(t). \quad (3)$$

Where $M_{ij,kl}$ and $W_{ij,mn}$ are synaptic weights which define a receptive field of the neuron, I_{ij} is constant inputs to the neuron, and $Y_{kl}(t)$ and $Y_{mn}(t)$ are spike output of neuron kl and mn , respectively. This spike output is defined as a step function which is given by,

$$Y_{kl}(t) = \begin{cases} 1 & \text{if } U_{kl}(t) > \Theta_{kl}(t) \\ 0 & \text{else} \end{cases}. \quad (4)$$

In Eq.(4), $\Theta_{kl}(t)$ is a threshold of the action potential of the neuron kl which is given by,

$$\Theta_{kl}(t+1) = \Theta_{kl}(t) \exp(-1/\alpha_T) + V_T Y_{kl}(t) \quad (5)$$

Through Eq.(1)–Eq.(5), parameters, β_{ij} , α_F , α_L , α_T , V_F , V_L , and V_T are decided appropriately.

Based on this basic configuration of PCNN, we proposed the IC-PCNN. The IC-PCNN has three neurons per pixel of the image, and each neuron detect each RGB component of the pixel color. Here, in the IC-PCNN, there exists inhibitory connections among three R-, G-, and B- neurons which correspond to the same pixel. Where inhibitory connections are defined as follows.

$$\boldsymbol{\Gamma}_{ij} = \begin{pmatrix} 0 & \gamma_{GR} & \gamma_{BR} \\ \gamma_{RB} & 0 & \gamma_{GB} \\ \gamma_{RG} & \gamma_{BG} & 0 \end{pmatrix}. \quad (6)$$

Where the parameters γ has negative value and all the parameters of γ is defined appropriately to lead effective characteristics of the color PCNN for the application, such as segmentation, edge detection, and so on.

The schematic model of proposed IC-PCNN is illustrated in Figure 2. The feeding input and linking input of proposed IC-PCNN is given by,

$$\begin{pmatrix} F_{ij,R}(t+1) \\ F_{ij,G}(t+1) \\ F_{ij,B}(t+1) \end{pmatrix} = \begin{pmatrix} F_{ij,R}(t) \\ F_{ij,G}(t) \\ F_{ij,B}(t) \end{pmatrix} \exp(-1/\alpha_F) + V_F \sum_{k,l} M_{ij,kl} \begin{pmatrix} Y_{kl,R}(t) \\ Y_{kl,G}(t) \\ Y_{kl,B}(t) \end{pmatrix} + V_F \boldsymbol{\Gamma}_{ij} \begin{pmatrix} Y_{ij,R}(t) \\ Y_{ij,G}(t) \\ Y_{ij,B}(t) \end{pmatrix} + \begin{pmatrix} I_{ij,R} \\ I_{ij,G} \\ I_{ij,B} \end{pmatrix}, \quad (7)$$

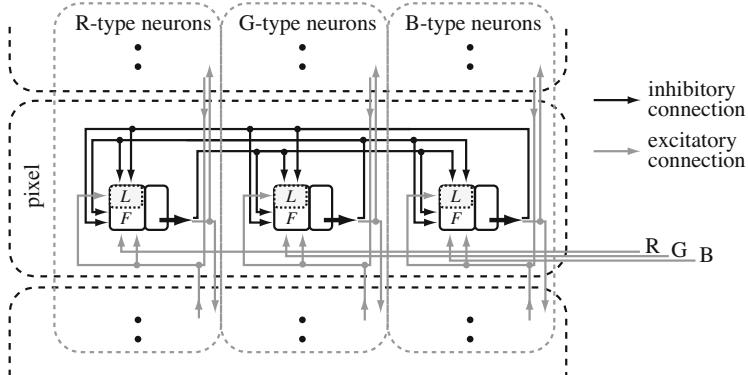


Fig. 2. The schematic model of Inhibitory Connected Pulse Coupled Neural Network (IC-PCNN) is illustrated^[7]. The IC-PCNN has neurons for each RGB component and inhibitory connections are defined among neurons in same pixel.

$$\begin{pmatrix} L_{ij,R}(t+1) \\ L_{ij,G}(t+1) \\ L_{ij,B}(t+1) \end{pmatrix} = \begin{pmatrix} L_{ij,R}(t) \\ L_{ij,G}(t) \\ L_{ij,B}(t) \end{pmatrix} \exp(-1/\alpha_L) + V_L \sum_{m,n} W_{ij,mn} \begin{pmatrix} Y_{mn,R}(t) \\ Y_{mn,G}(t) \\ Y_{mn,B}(t) \end{pmatrix} + V_L \boldsymbol{\Gamma}_{ij} \begin{pmatrix} Y_{ij,R}(t) \\ Y_{ij,G}(t) \\ Y_{ij,B}(t) \end{pmatrix}. \quad (8)$$

Where $F_{ij,R}$, $F_{ij,G}$, $F_{ij,B}$, $L_{ij,R}$, $L_{ij,G}$, and $L_{ij,B}$ is feeding or linking input of R-type, G-type, and B-type neuron of pixel ij .

In this study, the feeding input of the neuron accept only an external input, *i.e.*, a normalized intensity of any color component, as assumed in conventional studies^{[3][10]}. This assumption is widely used in an application of the image processing. For example, the feeding input of the neuron for the red component is defined as follows instead of Eq.(7).

$$F_{ij,R}(t+1) = I_{ij,R} = \frac{X_{ij,R}}{255}. \quad (9)$$

Where $X_{ij,R}$ denotes 8-bit intensity of red component of the pixel P_{ij} .

An example of the color image segmentation using IC-PCNN is shown in the Figure 3. Here, we used an Ishihara color test^{[1][11][12]} that is used in the test of color blindness. A input image is shown in Figure 3(a) which indicates a character "74" in the circle. As shown in Figure 3(a), we cannot recognize a character "74" in the circle from the gray-scaled image because the intensity difference of each pixel in the circle of the image does not form the character "74". From this peculiar characteristics of the color test image, it is clearly considered that conventional "gray-scale PCNN" cannot achieve segmentation of this character. On

¹ Here we cannot show an actual color image of the color test because this manuscript will be published in monochrome. However, we can easily find this color test on webpage^[12].

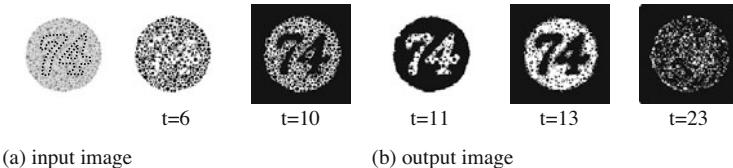


Fig. 3. (a) The input image to be segmented. images are color images in reality. Dotted line shows boundary of the object in the image. (b) The output image from IC-PCNN.

the other hand, using IC-PCNN, successful results of segmentation is obtained as shown in Figure 3(b). This results shows that the IC-PCNN has the qualities for the color image segmentation essentially.

3 The Characteristics of the Inhibitory Connections to the Results of Image Segmentation

First, we show the simulation results to describe an effect of the strength of the inhibitory connections to the results of the image segmentation using IC-PCNN. The images to be processed in these simulations are shown in Figure 4. These images are 64×64 pixel color image and consist of 2 colors area. The color of the object is R,G,B=(93,205,67) and that of background is R,G,B=(201,119,65). Also the Gaussian noise is added to both images in the simulation.

In the simulations, all the parameters of IC-PCNN is assumed appropriately defined. However the inhibitory connections are varied depending on the following procedure. To define the inhibitory connections, namely γ_{XX} in Eq. (6), we first choose the average and the variance of the inhibitory connections and generate 6 values randomly. Whole 720 patterns of inhibitory connections using these 6 values are tested and the mean square error (MSE) to the ideal segmentation for every results are calculated. Thus 720 MSEs are obtained for a set of the average and the variance. Then the strength of the inhibitory connection is changed and this procedure is repeated. Here the ideal segmentation is assumed that the object is exactly segmented from the image.

From the results in Figure 5, we can observe a wide region of the strength of the inhibitory connections in which the small MSE to the ideal segmentation is obtained. Moreover, it is observed that each of regions are continuous. Figure

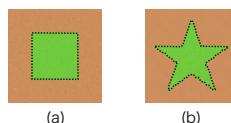


Fig. 4. The test images to be used in simulations. The images are color images and dotted lines show boundary of the object.

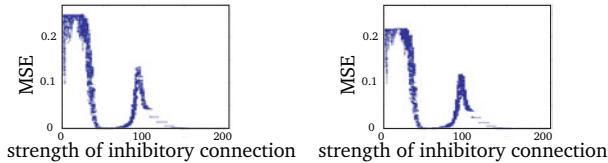


Fig. 5. Characteristics of MSEs corresponding to the average of the inhibitory connections. The variance of the inhibitory connections is 2.98.(a) Figure 4(a) is used for the test image. (b) Figure 4(b) is used for the test image.

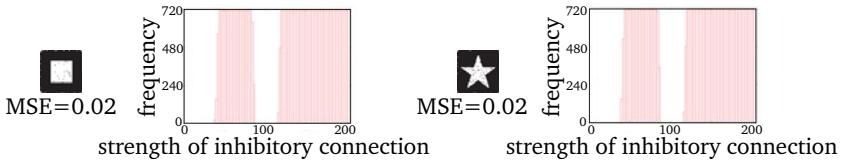


Fig. 6. The relationship between the frequency to achieve $MSE \leq 0.02$ and the average of the inhibitory connections. The variance of the inhibitory connections is 2.98. (a) Figure 4(a) is used for the test image.(b) Figure 4(b) is used for the test image.

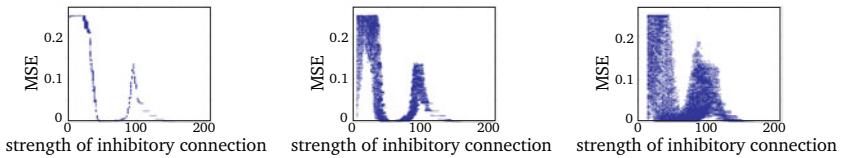


Fig. 7. Characteristics of MSEs corresponding to the average of the inhibitory connections. Figure 4(a) is used for the simulation. (a)The variance of the inhibitory connections is 0.18. (b)18.6. (c)146.

Figure 6 shows the relationship between the frequency to achieve $MSE \leq 0.02$ and the average strength of the inhibitory connections. From the results, the regions of $MSE \leq 0.02$ are also continuous and the value replacing among γ_{XX} does not affect to the results of the segmentation. Also, we can find that these characteristics does not depend on the form of the object. Therefore we can presume that the valid region of the average of strength in inhibitory connections exists.

From the results in above, the combinations of mutual inhibitions among R- G- B- neurons does not affect to the results of the segmentation. However the inhibitions among neurons corresponds to the firing timing of the neuron and it must affect synchronous firing assembly in the IC-PCNN. Therefore, we investigate an effect of the variance of inhibitory connections to the results of the image segmentation. Here the variance of inhibitory connections corresponds to the deviation of neuron's inhibitory effect to other neurons in the pixel. In all simulations, the conditions of simulation is same as the simulations in above.

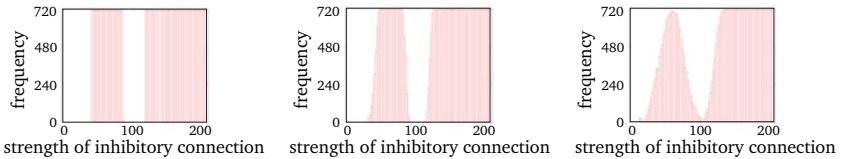


Fig. 8. The relationship between the frequency to achieve $MSE \leq 0.02$ and the average of the inhibitory connections. Figure 4(a) is used for the simulation. (a)The variance of the inhibitory connections is 0.18. (b)18.6.(c)146.

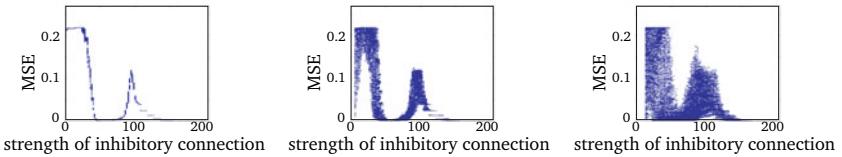


Fig. 9. Characteristics of MSEs corresponding to the average of the inhibitory connections. Figure 4(b) is used for the simulation. (a)The variance of the inhibitory connections is 0.18. (b)18.6.(c)146.

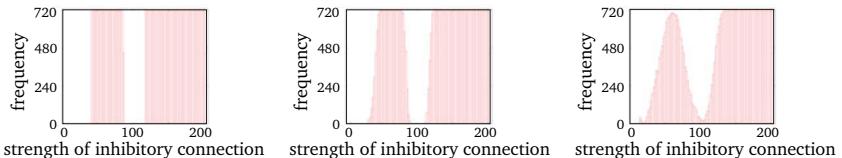


Fig. 10. The relationship between the frequency to achieve $MSE \leq 0.02$ and the average of the inhibitory connections. Figure 4(b) is used for the simulation. (a)The variance of the inhibitory connections is 0.18. (b)18.6.(c)146.

Figure 7 shows the characteristics of MSEs corresponding to the average of the inhibitory connections. Each figure (a), (b) and (c) shows the result in the case that the variance of inhibitory connection is 0.18, 18.6 and 146, respectively. Also corresponding the results in Figure 7, the relationship between the frequency to achieve $MSE \leq 0.02$ and the average of the inhibitory connections are shown in Figure 8. In these results, Figure 4(a) is used as an image to be segmented. Furthermore, in Figure 9 and 10, we show the similar results of the simulation in the case that Figure 4(b) is used.

From these results, we can find that the region of the inhibitory connection to achieve $MSE \leq 0.02$ is relatively wider in the case that the variance has large value. On the other hand, in large variance case, performance of segmentation becomes to depend on the combination of the value in γ_{XX} . This results suggest that an effect of the mutual inhibition among neurons become to appear in large variance of the inhibitory connections.

Also, in conventional studies, an effect of another parameters of IC-PCNN is studied. For example, it is known that the PCNN achieve "perfect segmentation" in specific condition of some parameters[10]. These parameters characteristics are also corresponds to the dynamics of IC-PCNN and they must be clarified in the future work. In any case, characteristics of inhibitory connections in the IC-PCNN which are shown in this study is considered as a key issue to reveal the synchronous firing assembly in the IC-PCNN and is valid to engineering application of IC-PCNN, *e.g.* image segmentation and others.

4 Conclusion

In this study, we described the effect of the inhibitory connections to the characteristics of synchronous firing assembly in the IC-PCNN. In the simulations, IC-PCNN was applied to the color image segmentation and the characteristics of the inhibitory connections related to its average strength and variance were described. The results showed that the valid domains of inhibitory connections for the color image segmentation exists. Also it is suggested that the results lead a key to find appropriate values of inhibitory connections for the image processing using IC-PCNN.

References

1. Echorn, R., Reitboeck, H.J., Arndt, M., Dicke, P.: Feature linking via synchronization among distributed assemblies: Simulations of results from cat visual cortex. *Neural Computation* 2, 293–307 (1990)
2. Echorn, R.: Neural Mechanisms of Scene Segmentation: Recording from the Visual Cortex Suggest Basic Circuits for Liking Field Model. *IEEE Trans. Neural Network* 10(3), 464–479 (1999)
3. Johnson, J.L., Padgett, M.L.: PCNN Models and Applications. *IEEE Trans. Neural Network* 10(3), 480–498 (1999)
4. Lindblad, T., Kinser, J.M.: Image processing using Pulse-Coupled Neural Networks, 2nd edn. Springer, Heidelberg (2005)
5. Zhou, L., Sun, Y., Zheng, J.: Automated Color Image Edge Detection using Improved PCNN Model. *WSEAS Transactions on Computers* 7(4), 184–189 (2008)
6. Xiong, X., Wang, Y., Zhang, X.: Color Image Segmentation using Pulse-Coupled Neural Network for Locusts Detection. In: Proc. of the International Conference on Data Mining, pp. 410–413 (2006)
7. Kurokawa, H., Kaneko, S., Yonekawa, M.: A color image segmentation using inhibitory connected pulse coupled neural network. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) *ICONIP 2008. LNCS*, vol. 5507, pp. 776–783. Springer, Heidelberg (2009)
8. Yonekawa, M., Kurokawa, H.: An automatic parameter adjustment method of Pulse Coupled Neural Network for image segmentation. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) *ICANN 2009. LNCS*, vol. 5768, pp. 834–843. Springer, Heidelberg (2009)
9. Yonekawa, M., Kurokawa, H.: The parameter optimization of the pulse coupled neural network for the pattern recognition. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) *ICANN 2010. LNCS*, vol. 6354, Springer, Heidelberg (2010)

10. Ranganth, H.S., Kuntimad, G.: Perfect image segmentation using pulse coupled neural networks. *IEEE Trans. Neural Networks* 10(3), 591–598 (1999)
11. Ishihara, S.: Tests for colour-blindness, Handaya (1917)
12. Images of the Ishihara color test are able to be found on web pages, for example,
http://en.wikipedia.org/wiki/Ishihara_color_test
<http://www.toledo-bend.com/colorblind/Ishihara.htm> etc

Array-Enhanced Stochastic Resonance in a Network of Noisy Neuromorphic Circuits

Gessyca Maria Tovar, Tetsuya Asai, and Yoshihito Amemiya

Graduate School of Information Science and Technology, Hokkaido University,
Kita 14, Nishi 9, Kita-ku, Sapporo, 060-0814, Japan
gessyca@lalsie.ist.hokudai.ac.jp
<http://lalsie.ist.hokudai.ac.jp/en/>

Abstract. Stacy *et.al.* demonstrated that an array of simulated hippocampal CA1 neurons exhibited stochastic resonance-like behaviors where an optimal correlation value between the sub-threshold input and the output was obtained by tuning both the noise intensity and the connection strength between the CA1 neurons. Based on this model, we proposed a simple neural network model for semiconductor devices. We carried out simulations using internal noise sources and confirmed that the correlation value between input and output in the network increased as the coupling strength increased.

Keywords: Neuromorphic LSI, Stochastic Resonance, Oscillators.

1 Introduction

Stochastic Resonance (SR) [1], refers to a situation where the response of a system can be optimized by the addition of optimal level of noise. Noises are usually considered obstacles in the operation of both analog and digital information processing systems, and most strategies to deal with them are focused on the suppression. On the other hand, neural systems may employ different strategies to exploit the properties of noise improving the efficiency of neural operations.

It is well known that there are numerous noise sources in nervous systems. Chaos in neural systems might work as fluctuations sources for the systems [2]; in the membrane of neurons, the ion channels are known to be stochastic [3]; receptor cells in sensory systems receive signals from the outer world, thus they are exposed to the fluctuations of the environment [4]. Thus, suggesting that SR may play a significant role in neural systems. It has been observed that SR gets enhanced if a coupled array of similar nonlinear elements responds to the same signal. This phenomenon, known as array enhanced stochastic resonance (AESR), was first observed in chains of nonlinear oscillators [5], and latter in ion channels [6], arrays of FitzHugh-Nagumo model neurons [7], and in a globally coupled network of Hodgkin-Huxley models neurons [8].

Regarding the applications of AESR, in [9] Inchiosa *et.al.* suggested that for practical systems, the coupling between nonlinear elements, such as nonlinear

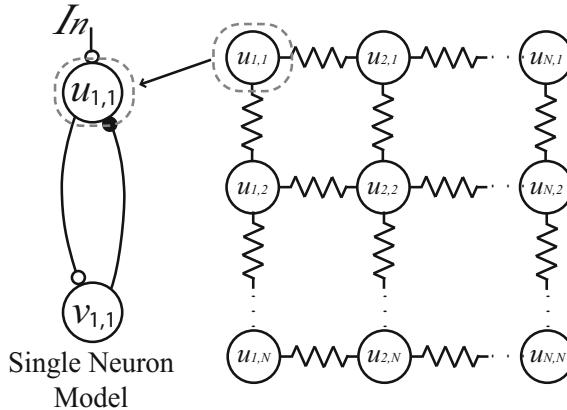


Fig. 1. Network model

detector of weak signal embedded in background noise, may enhance their input-output correlation. Also, Lindner *et.al.* [10] understand AESR as a matching of time and space scales through the tuning of noise and coupling that enhances synchronization and the signal processing ability of a locally and linearly coupled array of bistable elements.

Recently, Schweihofer *et.al.* [11] reported that the spiking behavior of a network of coupled Inferior Olive cells became chaotic for moderate electrical coupling, under these circumstances the input-output information transmission increased. In addition, in [12] Stacy *et.al.* demonstrated that an array of simulated hippocampal CA1 neurons exhibit SR-like behavior where an optimal correlation value between the sub-threshold input and the output was obtained by tuning both the noise intensity and the connection strength between the CA1 neurons, and, the correlation was further increased as the number of neurons increased.

Motivated by these findings, we proposed a neural network model composed of Wilson-Cowan neuron models [13] that is suitable for implementing with semiconductor devices, to exploit noise and fluctuations in coupled electronic systems for constructing brain-inspired computer architectures. In the network each neuron device is electrically coupled to its four neighbors to form a 2D grid network. All neurons accept a common sub-threshold input. The output of the network is defined by the sum of all the neurons. We carried out numerical and circuit simulations using a simulation program with integrated circuit emphasis (SPICE), and confirmed that the correlation value between the input and the output of the network can be increased by adjusting the coupling strength.

2 Network Model and Numerical Simulations

Our network model is illustrated in Fig. 1. The network has $N \times N$ neural oscillators consisting of the Wilson-Cowan neural oscillator model [13]. In the

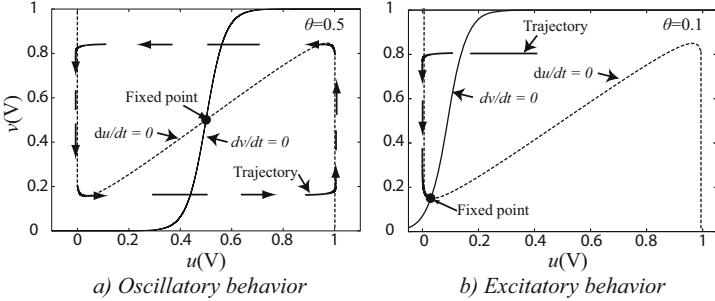


Fig. 2. Nullclines of a single oscillator for different θ s

network each neural oscillator is electrically coupled to its four neighbors to form a 2D grid network. The dynamics are defined by

$$\tau \frac{du_{i,j}}{dt} = -u_{i,j} + \frac{\exp(\beta_1 u)}{\exp(\beta_1 u) + \exp(\beta_1 v)} + In + \sum_{j \neq i} g_{i,j} U_{j,j}, \quad (1)$$

$$\frac{dv_{i,j}}{dt} = -v_{i,j} + \frac{\exp(\beta_2 u)}{\exp(\beta_2 u) + \exp(\beta_2 \theta)}, \quad (2)$$

where $U_{i,j}$ is given by

$$U_{i,j} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}, \quad (3)$$

In Eqs. (1) and (2), τ represents the time constant, N represents the size of the matrix ($N \times N$), In is the common input to the neurons, and $g_{i,j}$ is the connection strength between oscillators. The constant θ determines the state (behavior) of the neuron. Figure 2 shows the nullclines and trajectory of a single oscillator for different θ 's (0.1 and 0.5). The remaining parameters were set at $\tau = 0.1$, $\beta_1 = 5$ and $\beta_2 = 10$. As shown in the figure, depending on the position of the fixed point, the neuron exhibits oscillatory or excitatory behavior. When θ is 0.5, the fixed point is located on nullcline u at which $du/dv > 0$. In this case, the neuron exhibits limit-cycle oscillations (see Fig. 2(a)). On the other hand, when θ is 0.1, the fixed point is located on nullcline u at which $du/dv < 0$. In this case, the neuron exhibits excitatory behavior (see Fig. 2(b)) and it is stable at the fixed point as long as an external stimulus is not applied.

Figure 3 shows the numerical solution of Eqs. (1) and (2) with 1000×1000 neurons, where the values of $u_{i,j}$ are represented in a black/white scale ($u_{i,j} < 0.5 \rightarrow$ black and $u_{i,j} \geq 0.5 \rightarrow$ white). The value of the rest of the parameters were set at $\tau = 0.01$, $\theta = 0.1$ (excitatory behavior), $\beta_1 = 5$, $\beta_2 = 10$, $In = 0$ and the gap junction $g_{i,j} = 0.035$. The solution was numerically obtained by solving the ODEs with the fourth-order Runge-Kutta method. At each corner of the network, the values of represented by $[i, j] \rightarrow [0, j]$ and $[i, j] \rightarrow [N + 1, j]$ are treated as $[i, j] \rightarrow [N, j]$ and $[i, j] \rightarrow [1, j]$, respectively. The initial conditions of the neurons were set as follow: most neurons were initially set to $u_{i,j} = 0.1$ and

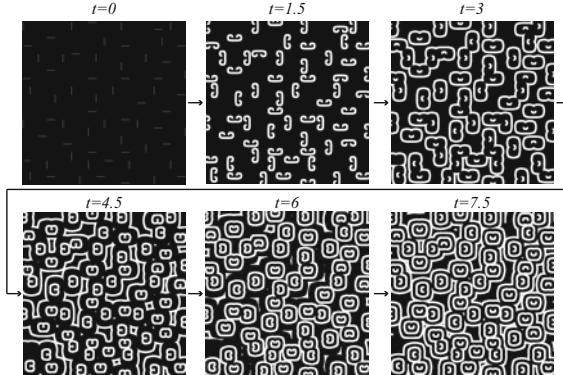


Fig. 3. Numerical results of 1000×1000 network with $\theta = 0.1$ (excitatory behavior)

$v_{i,j} = 0.3$ (inactive or excitatory mode), the neurons represented by white lines in the figure ($t = 0$) were set to $u_{i,j} = 0.9$ and $v_{i,j} = 0.6$ (active mode) and the neurons adjacent to each white line were set to $u_{i,j} = 0.0001$ and $v_{i,j} = 0.68$ (refractory mode). The inactive neurons next to the active neurons (white line) were excited (activated) through their connection with the active neurons, then went back to the inactive mode following the pattern showed in the figure ($t = 1.5$ to $t = 7.5$). We used this continuous pattern as an internal noise source for the network.

We conducted numerical simulations with In set as a periodic sub-threshold pulse. The rest of the parameters were set as previously described. The firing of each neuron was recorded and it was converted into a series of pulses of amplitude 1 and 0 corresponding to the firing and non-firing states respectively. The output (out) of the network was then defined by the sum of all the pulses divide by the number of neurons. To evaluate the performance of the network, we calculated correlation values C between converted sub-threshold input pulse (in) ($in = 0$ for $In = 0$, $in = 1$ for $In > 0$) and the output (out) given by:

$$C = \frac{\langle in.out \rangle - \langle in \rangle \langle out \rangle}{\sqrt{\langle in^2 \rangle - \langle in \rangle^2} \sqrt{\langle out^2 \rangle - \langle out \rangle^2}}. \quad (4)$$

Figure 4 shows the simulations results, as shown the correlation value between input and output increased as the coupling strength increased reaching a maximum peak when the coupling was around 0.12 and then decreasing again. In addition, we varied the noise levels according to the number of spirals (number of white lines set in the initial conditions; see Fig. 3). When the number of spirals consists of just a few lines the noise pattern becomes almost periodic and it is no longer considered noise, for too many spirals the neuron cancel each other and the pattern disappears. However, for a moderated number of spirals, (≈ 30) the noise pattern is similar to that shown in Fig. 3 at $t = 7.5$ resulting in a random pattern as the time increased. As shown in Fig. 4, for medium levels of noise (about 30 spirals) the correlation values reached a maximum. These results indicate that, if

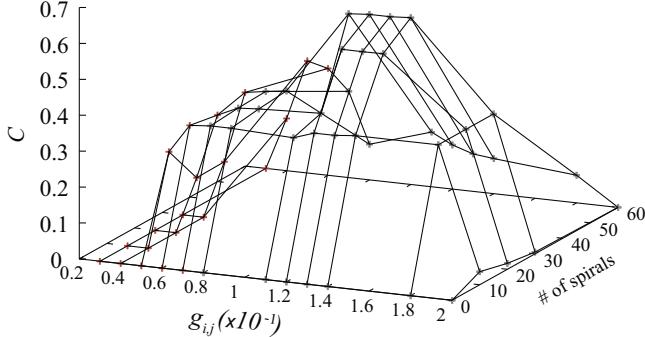


Fig. 4. Numerical results showing the Correlation value vs coupling strength and noise levels (number of spirals)

we postulate signal transmission via the array of the neuron devices under a noisy environment where the noise strength is fixed, the transmission error rate could be tuned by the coupling strength.

3 CMOS Circuit Implementation

Our Wilson-Cowan based neural oscillators have been implemented in [14]. The oscillator consists of two pMOS differential pairs ($m_1 - m_2$ and $m_3 - m_4$), two capacitors (C_1 and C_2) and two resistors (g) as shown in Fig. 5(a). The differential-pairs sub-threshold currents, I_1 and I_2 , are given by:

$$I_1 = I_b \frac{\exp(\kappa u/V_T)}{\exp(\kappa u/V_T) + \exp(\kappa v/V_T)}, \quad (5)$$

$$I_2 = I_b \frac{\exp(\kappa u/V_T)}{\exp(\kappa u/V_T) + \exp(\kappa \theta/V_T)}, \quad (6)$$

where I_b represents the differential pairs bias current, V_T is the thermal voltage, k is the Boltzmann's constant. The circuit dynamics can be determined by applying Kirchhoff's current law to both differential pairs, as follows:

$$C_1 \dot{u} = -gu + \frac{I_b \exp(\kappa u/V_T)}{\exp(\kappa u/V_T) + \exp(\kappa v/V_T)}, \quad (7)$$

$$C_2 \dot{v} = -gv + \frac{I_b \exp(\kappa u/V_T)}{\exp(\kappa u/V_T) + \exp(\kappa \theta/V_T)}, \quad (8)$$

where κ is the subthreshold slope, C_1 and C_2 are the capacitances representing the time constants, and θ is bias voltage. Note that Eqs. (7) and (8) correspond to the dynamics of the network explained above (Eqs. (11) and (12)) for $In = 0$ and $g_{i,j} = 0$. The simulated nullclines and the trajectory of a single neuron circuit for $\theta = 0.11$ are shown in Fig. 5(b).

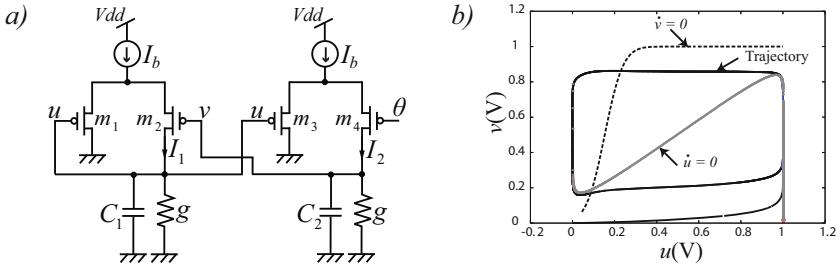


Fig. 5. Single neural oscillator circuit and circuit's nullclines

4 Simulations Results

We carried out circuit simulations of a 10×10 circuit network. The parameter sets we used for the transistors were obtained from MOSIS 1.5- μm CMOS process. Transistor sizes of the differential pairs (Fig. 5) were fixed at $L = 1.6 \mu\text{m}$ and $W = 4 \mu\text{m}$. The supply voltage was set at 3 V. The neurons were locally connected by pass transistors instead of linear resistors. The connection strength was controlled by the common gate voltage (V_g) of these transistors. The values of the bias current I_b and the resistors g were set so that $g \times I_b = 1$. The capacitors C_1 and C_2 were set at 0.1 pF and 10 pF respectively. Figure 6 shows the wave propagation of the circuit network with $V_g = 0.4$ V. For the simulations the initial condition first neuron ($u_{i,j}$ and $v_{i,j}$ for $i, j = 1, 1$) were set to be in the active mode ($u_{1,1} = 0.99$ V and $v_{1,1} = 0.18$ V; white dot in the figure), the rest of the neurons were set to be in the inactive mode ($u_{i,j} = 0.01$ V and $v_{i,j} = 0.18$ V). As previously explained, the inactive neurons located next to the active neuron (white dot) were activated through their connection with the active neuron, then went back to the inactive mode as the neuron's pulse finish its course following the pattern showed in the figure.

We conducted circuit simulations of a network of 100×100 neurons. For the simulations, each neuron was excited with a periodic sub-threshold current pulse. The simulations results are shown in Fig. 7. The figure shows the correlation value as a function of the coupling strength. With a low coupling strength the correlation values were almost 0, however, as the coupling strength increased the correlation value also increased. It is important to note that, as the network size is small when compare to that of the numerical simulations, the number of spiral set in the initial conditions is also small (about 7 spirals). Due to this, the maximum correlation value between the input and the output that could be obtained was around $C = 0.32$, less than half the value obtained by numerical simulations. However, these results proved that effectively an increase of the correlation value can be attained by tuning the coupling strength, suggesting that as the size of the network and the number of spiral increases, the performance of the network also increases.

These results indicate that, under a noisy environment the performance of nonlinear systems can be improved by coupling the devices. As suggested in [9]

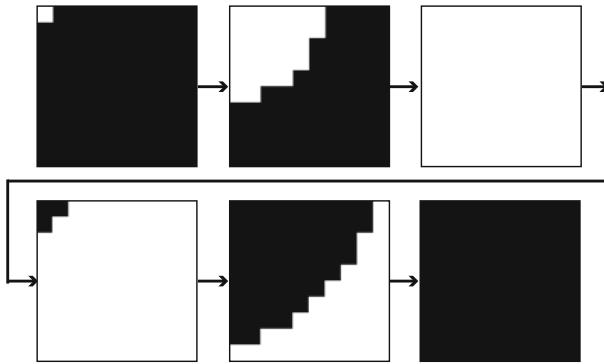


Fig. 6. Circuit simulations results showing the wave propagation of the circuit with 10×10 neurons

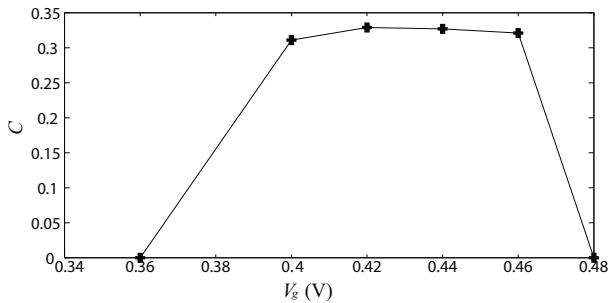


Fig. 7. Circuit simulations results showing the correlation vs the coupling strength with 100×100 neurons

the coupling can be applied in practical systems like nonlinear detectors of weak signal or for optimizing the information transmission of the network. In the neuromorphic point of view, these results may be an important step toward the construction of robust brain-inspired computer systems.

5 Conclusion

Neuromorphic devices exhibiting array-enhanced stochastic resonance were introduced. The model consisted of $N \times N$ neural oscillators. Each oscillator was connected to its four neighbors to form a 2D grid. We used the wave propagation characteristic of the network as internal noise sources. We carried out numerical simulations of 1000×1000 network and proved that the correlation value between input and output can be increased by tuning the coupling strength. Moreover, we designed a circuit network of the model. We simulated a circuit network of 10×10 neurons to show the wave propagation of the network and carried out circuit simulations of a larger circuit network 100×100 . The results

showed that the correlation value was increased by tuning the coupling strength, however, due to the size of the network the correlation value was lower than that obtained in numerical simulations, suggesting that in much larger network the circuit the correlation increased.

Acknowledgments. This study was supported by a Grant-in-Aid for Scientific Research on Innovative Areas [20111004] from the Ministry of Education, Culture Sports, Science and Technology (MEXT) of Japan.

References

1. Gammaitoni, L., Hanggi, P., Jung, P., Marchesoni, F.: Stochastic resonance. *Reviews of Modern Physics* 70(1), 223–287 (1998)
2. Freeman, W.J.: Simulation of chaotic EEG patterns with a dynamic model of the olfactory system. *Biol. Cybern.* 56, 139–150 (1989)
3. DeFelice, L.J., Isaac, A.: Chaotic states in a random world. *J. Stat. Phys.* 70, 339–354 (1992)
4. Smith, C.U.M.: *Biology of Sensory Systems*. John Wiley & Sons Ltd., West Sussex (2000)
5. Lindner, J.F., Meadows, B.K., Ditto, W.L., Inchiosa, M.E., Bulsara, A.R.: Array Enhanced Stochastic Resonance and Sapatiotemporal Synchronization. *Phys. Rev. Lett.* 75, 3–6 (1995)
6. Bezrukov, S.M., Vodyanoy, I.: Noise-induced enhancement of signal transduction across voltage-dependent ion channels. *Nature (London)* 378, 362–364 (1995)
7. Collins, J.J., Chow, C.C., Imhoff, T.T.: Aperiodic stochastic resonance in excitable systems. *Phys. Rev. E.* 52(4), 3321–3324 (1995)
8. Liu, F., Hu, B., Wang, W.: Effects of correlated and independent noise on signal processing in neuronal systems. *Phys. Rev. E.* 63, 31907 (2001)
9. Inchiosa, M.E., Bulsara, A.R.: Coupling enhances stochastic resonance in nonlinear dynamic elements driven by a sinusoid plus noise. *Physics Letters A* 200, 283–288 (1995)
10. Lindner, J.F., Meadows, B.K., Ditto, W.L.: Scaling laws for spatiotemporal synchronization and array enhanced stochastic resonance. *Physical Rev. E.* 53(3), 2081–2086 (1996)
11. Schweighofer, N., Doya, K., Fukai, H., Chiron, J.V., Furukawa, T., Kawato, M.: Chaos enhance information transmission in the inferior olive. *PNAS* 101(13), 4655–4660 (2004)
12. Stacy, W.C., Durand, D.M.: Noise and coupling affect signal detection and bursting in a simulated physiological neural network. *J. Neurophysiol.* 88, 2598–2611 (2002)
13. Wilson, H.R., Cowan, J.D.: Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.* 12, 1–24 (1972)
14. Gessyca, T.N., Asai, T., Hirose, T., Amemiya, Y.: Critical Temperature Sensor Based on Oscillatory Neuron Models. *J. of Signal Processing.* 12(1), 17–24 (2008)

Modelling the Interplay of Emotions, Beliefs and Intentions within Collective Decision Making Based on Insights from Social Neuroscience

Mark Hoogendoorn, Jan Treur, C. Natalie van der Wal, and Arlette van Wissen

VU University Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
[{mhoogen,treur,cn.van.der.wal,wissen}](mailto:{mhoogen,treur,cn.van.der.wal,wissen}@few.vu.nl)
<http://www.few.vu.nl/~{mhoogen,treur,cn.van.der.wal,wissen}>

Abstract. Collective decision making involves on the one hand individual mental states such as beliefs, emotions and intentions, and on the other hand interaction with others with possibly different mental states. Achieving a satisfactory common group decision on which all agree requires that such mental states are adapted to each other by social interaction. Recent developments in Social Neuroscience have revealed neural mechanisms by which such mutual adaptation can be realised. These mechanisms not only enable intentions to converge to an emerging common decision, but at the same time enable to achieve shared underlying individual beliefs and emotions. This paper presents a computational model for such processes.

Keywords: computational modeling, collective decision making, social neuroscience, mirroring, belief, emotion, intention.

1 Introduction

When it comes to group decision making versus individual decision making, it is often said that ‘two heads are better than one’, and ‘the more the merrier’. Combining the individual capabilities in a group setting is often perceived as a benefit for all parties involved. However, deciding as a group comes with substantial challenges, as each group member has autonomous neurological processes, carrying, for example, private mental states such as emotions, beliefs, and intentions, which may seem hard to combine within a group. So, viewed from a distance, group decision making by reaching mutual agreement could be very hard. Yet, quite often coherent decisions are made by groups, and group members even seem to feel good with these decisions. In recent years, this seeming paradox has been resolved by developments in the new area called Social Neuroscience; e.g., [2], [3], [10], [11], [14], [25].

The crux is that after all these private mental states are not so static and isolated as they may seem; they often show high extents of dynamics due to social interaction. In Social Neuroscience neural mechanisms have been discovered that indeed - often in unconscious manners - account for mutual *mirroring* effects between mental states of different persons; e.g., [17], [23], [24]. For example, an emotion expresses itself in a

smile which, when observed by another person, automatically triggers certain preparation neurons (also called *mirror neurons*) for smiling within this other person, and consequently generates the same emotion. Similarly, mirroring of intentions and beliefs can be considered.

In this paper group decision making in stressful circumstances (with emergency evacuations as an example) is addressed. Here emotions have an important interaction with the beliefs and intentions involved in a decision making process. A computational model is introduced that not only incorporates mechanisms for mirroring emotions, intentions and beliefs between different persons, but also addresses how within a person beliefs and emotions affect each other, and how they both affect the person's intentions.

2 Background from Social Neuroscience

Within Neuroscience it has been discovered that certain neurons have a *mirroring function* of (e.g., [9], [17], [18], [20], [21], [22], [23], [24]). In the context of the neural circuits in which they are embedded, these neurons show both a function in preparation for certain actions or bodily changes and a function to mirror similar states of other persons: they are active also when the person observes somebody else intending or performing the action or body change. This includes expressing emotions in body states, such as facial expressions. These neurons and the neural circuits in which they are embedded play an important role in social functioning; (e.g., [9], [17], [23], [24]). When mental states of other persons are mirrored by some of the person's own states, which at the same time play a role in generating their own behaviour, then this provides an effective basic mechanism for how in a social context persons fundamentally affect each other's mental states and behaviour. These discoveries are the basis for an exciting new research area, called Social Neuroscience.

A person's cognitive states usually induce emotions, as described by neurologist Damasio, [6], [7]; for example: 'Even when we somewhat misuse the notion of feeling – as in "I feel I am right about this" or "I feel I cannot agree with you" – we are referring, at least vaguely, to the feeling that accompanies the idea of believing a certain fact or endorsing a certain view. This is because believing and endorsing *cause* a certain emotion to happen.' ([7], p. 93). Damasio's *Somatic Marker Hypothesis*; cf. [1], [5], [7], [8], 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 ([5], pp. 173-174).

In Figure 1 an overview of the interplay of the different states within the model for collective decision making is shown. It is assumed that at the individual level the strength of an intention for a certain decision option depends on the person's beliefs (*cognitive responding*) and emotions (*somatic marking*) in relation to that option. Moreover, it is assumed that beliefs may generate certain emotions (*affective responding*), for example of fear, that in turn may affect the strength of beliefs (*affective biasing*). Note that it is assumed that these latter emotions are independent of the different decision options.

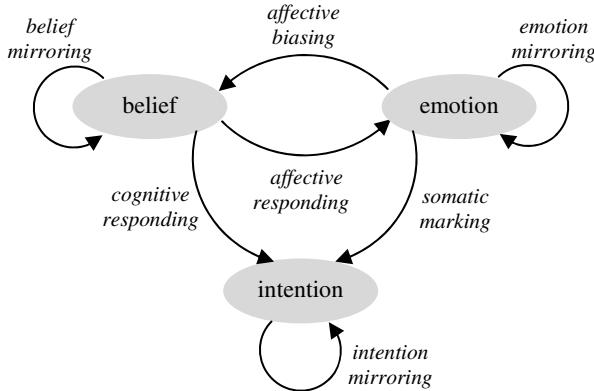


Fig. 1. The interplay of beliefs, emotions and intentions in social context

Given this, to obtain collectiveness of the decision making a mirroring mechanism as briefly described above is used in three different ways; see also Figure 1:

- *mirroring of emotions* is a mechanism for how fear and emotions felt in different individuals about a certain considered decision option mutually affect each other,
- *mirroring of beliefs* is a mechanism transferring information on the extent to which different individuals believe certain information
- *mirroring of intentions* is a mechanism transferring information between individuals on the strength of action tendencies (e.g., [13], p.70) for certain decision options

These mechanisms describe not only how over time the individual decision intentions of group members may converge to a common group intention, but also how this relates to a basis of shared beliefs and shared emotions developed within the group. Indeed, the computational model introduced in Sections 3 and 4 shows these types of patterns, as illustrated in Section 5.

3 A Computational Model for Mirroring of Mental States

A main building block of the computational model is a general model describing at an abstract level the mirroring of a given mental state S (for example, an emotion, belief or intention). This is based upon the model that was also used as a generic building block in [15], [16]. An important element is the contagion strength γ_{SBA} from person B to person A in a group. This denotes how much the state S of A is influenced by the state S of B . It is defined by

$$\gamma_{SBA} = \varepsilon_{SB} \alpha_{SBA} \delta_{SA} \quad (1)$$

Here, ε_{SB} is the personal characteristic *expressiveness* of the sender B for S , δ_{SA} the personal characteristic *openness* of the receiver A for S , and α_{SBA} the interaction characteristic *channel strength* for S from sender B to receiver A . In order to determine the level q_{SA} of state S in an agent A , the following calculations are performed. First, the overall contagion strength γ_A from the group towards agent A is calculated:

$$\gamma_{SA} = \sum_{B \neq A} \gamma_{SBA} \quad (2)$$

This value is used to determine the weighed impact q_{SA}^* of all the other agents upon state S of agent A :

$$q_{SA}^* = \sum_{B \neq A} \gamma_{SBA} q_{SB} / \gamma_{SA} \quad (3)$$

Two additional personal characteristics determine how much this external influence actually changes state S of agent A , namely the tendency η_{SA} to absorb or to amplify the level of a state and the bias β_{SA} towards increasing or reducing impact for the value of the state.

$$q_{SA}(t + \Delta t) = q_{SA}(t) + \gamma_{SA} [f(q_{SA}^*(t), q_{SA}(t)) - q_{SA}(t)] \Delta t \quad (4)$$

where the combination function $f(q_{SA}^*(t), q_{SA}(t))$ used was taken as:

$$f(q_{SA}^*(t), q_{SA}(t)) = \eta_{SA} [\beta_{SA} (1 - (1 - q_{SA}^*(t))(1 - q_{SA}(t))) + (1 - \beta_{SA}) q_{SA}^*(t) q_{SA}(t)] + (1 - \eta_{SA}) q_{SA}^*(t)$$

By (4) the new value for the state S at time $t + \Delta t$ is calculated from the old value at t , plus the change of the value based upon the transfer by mirroring. This change is defined as the multiplication of the overall contagion strength γ_{SA} times the difference of a combination function of q_{SA}^* and q_{SA} with q_{SA} . The combination function used has a component for amplification (after $\eta_{SA}(t)$) and one for absorption. The amplification component depends on the tendency of the person towards more positive (part multiplied by $\beta_{SA}(t)$) or negative (part of equation multiplied by $1 - \beta_{SA}(t)$ side). Table 1 summarizes the most important parameters and states within this general model.

Table 1. Parameters and states

q_{SA}	level for state S for person A
ϵ_{SA}	extent to which person A expresses state S
δ_{SA}	extent to which person A is open to state S
η_{SA}	tendency of person A to absorb or amplify state S
β_{SA}	positive or negative bias of person A on state S
α_{SBA}	channel strength for state S from sender B to receiver A
γ_{SBA}	contagion strength for S from sender B to receiver A

4 Modelling the Interplay of Beliefs, Emotions and Intentions

This section describes a computational model for the interplay of emotions, beliefs and intentions in a group of persons in the context of collective decision making. In this model the general model described in Section 3 is specialised for three different types of mental states S , namely beliefs, emotions, and intentions. In principle this a large number of variants of equation (4) above for all persons A in a group and all states S , indicated by *belief*(X), *fear*, *emotion*(O), *intention*(O) for information X and options O . However, in addition, at the individual level interactions between these different states are modelled, as depicted in Figure 1; see also Table 2 for a brief explanation of all interactions in the model. This means that the model obtained by forming specialisations of the generic model from Section 3 is modified in order to

incorporate the internal interactions between the different types of states. For example, as can be seen in Table 2, the effect of beliefs on fear of a person has to be combined with the effect of fear of other group members on the own fear. This will be explained in more detail in the remainder of this section.

Table 2. The different types of processes in the model

from S	to S'	type	description
$belief(X)$	$fear$	internal	affective response on information; for example, on threads and possibilities to escape
$emotion(O)$ $fear$	$emotion(O)$ $fear$	interaction	emotion mirroring by nonverbal and verbal interaction; for example, fear contagion
$fear$	$belief(X)$	internal	affective biasing; for example, adapting openness, amplification extent and orientation
$belief(X)$	$belief(X)$	interaction	belief mirroring by nonverbal and verbal interaction; for example, of information on threads and options to escape
$belief(X)$	$intention(O)$	internal	cognitive response on information; for example, aiming for an exit that is believed to be reachable
$emotion(O)$	$intention(O)$	internal	somatic marking of intention options; for example, giving options that feel bad a low valuation
$intention(O)$	$intention(O)$	interaction	intention mirroring by nonverbal and verbal interaction; for example, of tendency to go in a certain direction

4.1 The Effect of Emotions on Beliefs

To model the effect of emotions on information diffusion, below the personal characteristics δ_{SA} , η_{SA} and β_{SA} for a belief state $S = belief(X)$ are not assumed constant, but are instead modeled in a dynamic manner, depending on emotions. Personal characteristics $\varepsilon_{belief(X)A}$, $\delta_{belief(X)A}$, $\eta_{belief(X)A}$, $\beta_{belief(X)A}$ and interaction characteristic $\alpha_{belief(X)BA}$ are parameters in the model as described in Section 3. One additional category is introduced here, namely informational state characteristics r_{XA} denoting how relevant, and p_{XA} denoting how positive information X is for person A. An assumption made for the model is that the intensity of the fear state of a person will affect his ability to receive information, by affecting the value of the individual person characteristics; in particular, a high level of fear affects $\beta_{belief(X)A}$, $\eta_{belief(X)A}$ and $\delta_{belief(X)A}$. First the effect of fear upon the openness for a belief $belief(X)$ (characterized by a relevance r_{XA} of information X for A) is expressed:

$$\delta_{belief(X)A}(t+\Delta t) = \delta_{belief(X)A}(t) + \mu \cdot (1/(1+e^{-\sigma q_{fear,A}^{(t)} - \tau})) \cdot [(1 - (1-r_{XA}) q_{fear,A}(t)) - \delta_{belief(X)A}(t)] \Delta t \quad (5)$$

If $q_{fear,A}$ is lower than threshold τ (on the interval [0,1]), it will not contribute to the value of $\delta_{belief(X)A}$. If $q_{fear,A}$ has a value above τ , the openness will depend on the relevance of the information: when the relevance is high, openness will increase, while if the relevance is low, openness will decrease. In all formulae, μ is an adaptation parameter. This proposed model corresponds to theories of emotions as frames for selective processing, as described in [11], [19]. A distinction between amplification values

for different types of information is also made, depending on the emotional state fear. The dynamics for the characteristic $\eta_{\text{belief}(X)A}(t)$ modeling the amplification or absorption of $\text{belief}(X)$ are described as follows:

$$\eta_{\text{belief}(X)A}(t+\Delta t) = \eta_{\text{belief}(X)A}(t) + \mu \cdot (1/(1+e^{-\sigma(q_{\text{fear},A}(t) - \tau)}) \cdot [r_{XA} \cdot (1-p_{XA}) \cdot (q_{\text{fear},A}(t) - \eta_{\text{belief}(X)A}(t))] \cdot \Delta t \quad (6)$$

The emotion of fear only has an influence when it is above the threshold. In that case the parameter only changes for relevant, non-positive information for which the parameter starts to move towards the value for the emotion of fear (meaning this type of information will be amplified). This property represents an interpretation of [4] on how emotion can result in selective processing of emotion-relevant information.

The bias of a person is also influenced by its emotion, but in addition depends on the content of the information, which can be either positive or negative:

$$\beta_{\text{belief}(X)A}(t+\Delta t) = \beta_{\text{belief}(X)A}(t) + \mu \cdot (1/(1+e^{\sigma(q_{\text{fear},A}(t) - \tau)}) \cdot (1-q_{\text{belief}(X)A}(t)) \cdot [(z_A \cdot p_{XA} + (1-z_A) \cdot (1-p_{XA})) - \beta_{\text{belief}(X)A}(t)] \cdot \Delta t \quad (7)$$

Parameter τ is a number between 0 and 1 and represents a threshold for q_{fear} : when $q_{\text{fear}} > \tau$, then $q_{\text{fear},A}$ has an influence on the bias $\beta_{\text{belief}(X)A}(t)$. Parameter z_A is a personality characteristic; if $z_A = 1$, represents a person who is optimistic when he/she has a lot of fear: positive information will be strengthened more and negative information will be weakened more. The reverse happens when $z_A = 0$, this represents a person who is more ‘pessimistic’ when experiencing fear: negative information will be strengthened and positive information will be weakened. Both personality characteristics seem to exist in people: a bias towards the negative side of information in case of experiencing a high level of fear, corresponds with the narrowing hypothesis from Frederickson’s broaden-and-build theory in [12]. The reverse personality characteristic of being able to ‘stay optimistic’ under pressure is a personality characteristic that is found in leaders. These dynamically changing ‘parameters’ $\delta_{\text{belief}(X)A}(t)$, $\eta_{\text{belief}(X)A}(t)$, $\beta_{\text{belief}(X)A}(t)$ are used in the equation describing the dynamics of the belief state $\text{belief}(X)$:

$$q_{\text{belief}(X)A}(t+\Delta t) = q_{\text{belief}(X)A}(t) + \gamma_{\text{belief}(X)A}(t) [f(q_{\text{belief}(X)A}^*(t), q_{\text{belief}(X)A}(t)) - q_{\text{belief}(X)A}(t)] \Delta t \quad (8)$$

where the combination function $f(q_{SA}^*(t), q_{SA}(t))$ used is taken in a dynamic manner as:

$$f(q_{\text{belief}(X)A}^*(t), q_{\text{belief}(X)A}(t)) = \eta_{\text{belief}(X)A}(t) [\beta_{\text{belief}(X)A}(t) (1 - (1 - q_{\text{belief}(X)A}^*(t))(1 - q_{\text{belief}(X)A}(t))) + (1 - \beta_{\text{belief}(X)A}(t)) q_{\text{belief}(X)A}^*(t) q_{\text{belief}(X)A}(t)] + (1 - \eta_{\text{belief}(X)A}(t)) q_{\text{belief}(X)A}^*(t) \quad (9)$$

Note that since it depends on $\delta_{\text{belief}(X)A}(t)$, also $\gamma_{\text{belief}(X)A}(t)$ becomes dynamic.

4.2 The Effect of Beliefs on Emotions in the Dynamics of Fear

Besides modeling the influence of emotion upon the information contagion in the previous Section, the opposite direction is investigated in this Section: emotions being influenced by information. This influence is modeled by altering the overall weighed impact of the contagion of the emotional state for fear. This is expressed as follows:

$$q_{\text{fear},A}^*(t) = v_A \cdot (\sum_{B \neq A} \gamma_{\text{fear},BA} \cdot q_{\text{fear},B} / \gamma_{\text{fear},A}) + (1 - v_A) \cdot (\sum_X \omega_{X,\text{fear},A} \cdot (1 - p_{XA}) \cdot r_{XA} \cdot q_{\text{belief}(X)A}) \quad (9)$$

Here the influence depends on the impact from the emotion fear by others (the first factor, with weight v_A) in combination with the influence of the belief present within

the person. In this case, information has an increasing effect on fear if it is relevant and non positive. This $q_{fearA}^*(t)$ is used in the equation describing the dynamics of fear:

$$q_{fearA}(t+\Delta t) = q_{fearA}(t) + \gamma_{fearA} [f(q_{fearA}^*(t), q_{fearA}(t)) - q_{fearA}(t)] \Delta t$$

with

$$f(q_{fearA}^*(t), q_{fearA}(t)) = \eta_{fearA} [\beta_{fearA} (I - (I - q_{fearA}^*(t))(I - q_{fearA}(t))) + (1 - \beta_{fearA}) q_{SA}^*(t) q_{SA}(t)] + (1 - \eta_{fearA}) q_{fearA}^*(t)$$

4.3 The Effects of Beliefs and Emotions on Intentions

The abstract model for mirroring described above applies to emotion, belief and intention states S for an option O or the situation in general, but does not describe any interplay for intentions yet. Taking the Somatic Marker Hypothesis on decision making as a point of departure, not only intentions of others, but also own emotions affect the own intentions. To incorporate such an interaction, the basic model is extended as follows: to update $q_{intention(O)A}$ for an intention state S relating to an option O , both the intention states of others for O and the $q_{emotion(O)A}(t)$ values for the emotion state S' for O are taken into account. These intention and emotion states S and S' for option O are denoted by OI and OE , respectively:

Level of fear of person A :	$q_{fearA}(t)$
Level of emotion for option O of person A :	$q_{emotion(O)A}(t)$
Level of intention indication for option O of person A :	$q_{intention(O)A}(t)$
Level of belief supporting option O of person A :	$q_{beliefsfor(O)A}(t)$

Here $q_{beliefsfor(O)A}(t)$ denotes to aggregated support for option O by beliefs of A ; it is defined as

$$q_{beliefsfor(O)A}(t) = \sum_X \omega_{XOA} q_{belief(X)A} / \sum_X \omega_{XOA}$$

where ω_{XOA} indicates how supportive information X is for option O . The combination of the own (positive) emotion level and the rest of the group's aggregated intention is made by a weighted average of the two:

$$q_{intention(O)A}^{**}(t) = (\omega_{OIA1}/\omega_{OIEBA}) q_{intention(O)A}^*(t) + (\omega_{OEA2}/\omega_{OIEBA}) q_{emotion(O)A}(t) + (\omega_{OBA2}/\omega_{OIEBA}) q_{beliefsfor(O)A}(t)$$

$$\gamma_{intention(O)A}^* = \omega_{OIEBA} \gamma_{intention(O)A}$$

where ω_{OIA1} , ω_{OBA2} and ω_{OEA2} are the weights for the contributions of the group intention impact (by mirroring), the own emotion impact (by somatic marking), and the own belief impact on the intention of A for O , respectively, and

$$\omega_{OIEBA} = \omega_{OIA1} + \omega_{OEA2} + \omega_{OBA2}$$

The combination of the own belief level and the rest of the group's aggregated emotion for a certain option O is made by a weighted average of the two

$$q_{emotion(O)A}^{**}(t) = (\omega_{OEA1}/\omega_{OEBA}) q_{emotion(O)A}^*(t) + (\omega_{OBA1}/\omega_{OEBA}) q_{beliefsfor(O)A}(t) \quad (10)$$

$$\gamma_{emotion(O)A}^* = \omega_{OEBA} \gamma_{emotion(O)A} \quad (11)$$

where ω_{OEA1} and ω_{OBA1} are the weights for the contributions of the group emotion impact (by mirroring), the own belief impact on the emotion of A for O , respectively, and $\omega_{OEBA} = \omega_{OEA1} + \omega_{OBA1}$. Then the overall model for the dynamics of emotions and intentions for options becomes:

$$\begin{aligned}
 q_{emotion(O)A}(t + \Delta t) &= q_{emotion(O)A}(t) \\
 &+ \gamma_{intention(O)A} * [\eta_{emotion(O)A} (\beta_{emotion(O)A} (I - (1-q_{emotion(O)A}^{**}(t))(1-q_{emotion(O)A}(t))) \\
 &\quad + (1-\beta_{emotion(O)A}) q_{emotion(O)A}^{**}(t) q_{emotion(O)A}(t)) \\
 &\quad + (1 - \eta_{emotion(O)A}) q_{emotion(O)A}^{**}(t) - q_{emotion(O)A}(t)] \cdot \Delta t \\
 q_{intention(O)A}(t + \Delta t) &= q_{intention(O)A}(t) \\
 &+ \gamma_{intention(O)A} * [\eta_{intention(O)A} (\beta_{intention(O)A} (I - (1-q_{intention(O)A}^{**}(t))(1-q_{intention(O)A}(t))) \\
 &\quad + (1-\beta_{intention(O)A}) q_{intention(O)A}^{**}(t) q_{intention(O)A}(t)) \\
 &\quad + (1 - \eta_{intention(O)A}) q_{intention(O)A}^{**}(t) - q_{intention(O)A}(t)] \cdot \Delta t
 \end{aligned}$$

5 Simulation Results

In this section, the results of a case study will be presented. The goal of the case study was to investigate if the computational model can simulate the interplay of emotions, intentions and beliefs, as described in neuroscientific, social and psychological literature. The computational model was implemented in Matlab in the context of an evacuation scenario (see Appendix A¹ for the complete Matlab specification).

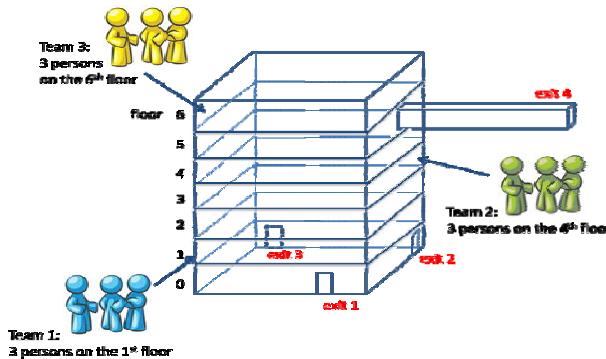


Fig. 2. The location of 3 teams in a building of 6 floors with 4 exits

The example scenario is expressed as follows: at the end of a working day in an office, the fire alarm goes off and all the persons that are in the building need to evacuate immediately. At the time of the alarm, 3 teams of each 3 people are present on different floors, as can be seen in Figure 2. Persons can communicate with each other when they are on the same floor, or they can communicate to each other through their personal device. Communication through such personal devices can only occur in case the distance is 3 floors or less. The building has 4 emergency exits. If an exit is accessible, the information is rated as ‘positive’ information, if not accessible then the information is rated ‘not positive’. According to the model, $p = 1$ and $p = 0$ are given as values to these two messages. The messages are always modeled as relevant for survival, $r = 1$.

¹ <http://www.cs.vu.nl/~mhoogen/social-diffusion/AppendixA-ICONIP10.pdf>

In the scenario, the three persons located at the top initially know that exit 4 is available (with a belief of 1), whereas the three persons on the middle floor do not have any strong beliefs about an emergency exit. The three at the first floor have beliefs of strength 1 concerning exit 1 and 2 (whereby the first one concerns a negative piece of information, namely that the exit is blocked and the second concerns positive information: the exit is accessible). Furthermore, a belief of strength 0 is present concerning exit 3. Besides these values, all other values are set to 0.5 with respect to the beliefs to indicate that they know the exits are there but do not know specifically whether the exit is accessible or not. Moreover, the intentions of all agents are initially set to 0 and the emotions to 0, 1, 0, and 1 for exit 1, 2, 3, and 4 respectively (since exit 1 and exit 3 represent negative information, the emotion for that option is not positive). Finally, for the emotion of fear the agents at the first floor have no fear, at the middle floor they have maximum fear, and at the top floor medium fear is present. Furthermore, the initial belief about the situation itself is 0.5. Regarding all the parameter setting as described before: each agent has the same initial set of parameters, and these can be found in the Matlab specification as shown in appendix A.

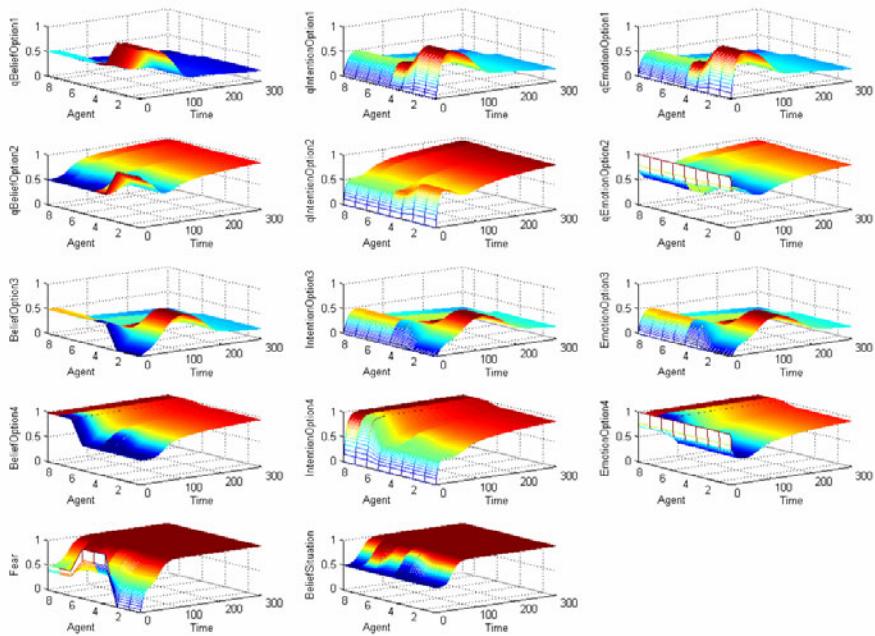


Fig. 3. Simulation results for the example scenario

Figure 3 shows the change of the values of the beliefs, intentions, and emotions. The top four rows represent the values related to the four exits. Here, the values for the agents during the simulation runs are shown. Furthermore, at the bottom row the amount of fear and the judgment of the entire situation are shown. It can be seen that fear spreads quickly, resulting in a very negative judgment of the situation by all

agents. For exit 1 the belief about the exit being congested eventually stabilizes at a relatively low value due to the fact that no human has a good feeling for that option (although in the beginning the emotions are slightly pulled upwards as well as the intention, due to the very strong belief of the three agents at the first floor). For exits 2 and 4 a very strong belief occurs rapidly for all agents as well as a very strong intention and the positive emotions also remain high. Finally, for exit 3 the agents at the first floor get a slightly stronger belief, intention, and emotion due to the fact that the other agents have a belief with value 0.5 about the exit. Eventually however, the values return to a rather low value again due to the fact that the others have lowered their value again.

6 Discussion

This paper has presented a computational model for collective decision making based on neural mechanisms revealed by recent developments in Social Neuroscience; e.g., [2], [3], [10], [11], [14], [25]. These mechanisms explain how mutual adaptation of individual mental states can be realised by social interaction. They not only enable intentions to converge to an emerging common decision, but at the same time enable to achieve shared underlying individual beliefs and emotions. Therefore a situation can be achieved in which a common decision is made that for each individual is considered in agreement with the own beliefs and feelings. More specifically, this model for collective design making involves on the one hand individual beliefs, emotions and intentions, and on the other hand interaction with others involving mirroring of such mental states; e.g., [17], [23], [24]. As shown in Figure 1 and in Table 2, the model involves seven types of interactions: three types of mirroring interactions between different persons, and within each person four types of interactions between the individual mental states.

In earlier work presented in [15] a simpler model for decision making was introduced in which only decision options and emotions associated to them, and their mutual interaction play a role, and no fear, nor interactions with beliefs. This model covers only three of the seven types of interaction of the currently presented model. The overlap is mainly in the somatic marking of intentions for decision options. In [16] a model was introduced in which only emotions and information and their mutual interaction play a role, and no decision making. The equations for the dynamics of δ , η , and β were adopted from this paper.

Acknowledgements. This research has partly been conducted as part of the FP7 ICT Future Enabling Technologies program of the European Commission under grant agreement No 231288 (SOCIONICAL).

References

1. Bechara, A., Damasio, A.: The Somatic Marker Hypothesis: a neural theory of economic decision. *Games and Economic Behavior* 52, 336–372 (2004)
2. Cacioppo, J.T., Berntson, G.G.: Social neuroscience. Psychology Press, New York (2005)

3. Cacioppo, J.T., Visser, P.S., Pickett, C.L.: Social neuroscience: People thinking about thinking people. MIT Press, Cambridge (2006)
4. Côté, S.: Reconciling the feelings-as-information and hedonic contingency models of how mood influences systematic information processing. *Journal of Applied Social Psychology* 35, 1656–1679 (2005)
5. Damasio, A.: *Descartes' Error: Emotion, Reason and the Human Brain*. Papermac, London (1994)
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 (2003)
8. 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)
9. Damasio, A., Meyer, K.: Behind the looking-glass. *Nature* 454, 167–168 (2008)
10. Decety, J., Cacioppo, J.T. (eds.): *The Handbook of Social Neuroscience*. Oxford University Press, Oxford (2010)
11. Decety, J., Ickes, W.: *The Social Neuroscience of Empathy*. MIT Press, Cambridge (2009)
12. Frederickson, B.L., Branigan, C.: Positive Emotions broaden the scope of attention and thought-action repertoires. *Cognition and Emotion* 19, 313–332 (2005)
13. Frijda, N.H.: *The Emotions*. Cambridge University Press, Cambridge (1987)
14. Harmon-Jones, E., Winkielman, P. (eds.): *Social neuroscience: Integrating biological and psychological explanations of social behavior*. Guilford, New York (2007)
15. Hoogendoorn, M., Treur, J., van der Wal, C.N., van Wissen, A.: Modelling the Emergence of Group Decisions Based on Mirroring and Somatic Marking. In: Yao, Y., Sun, R., Poggio, T., Liu, J., Zhong, N., Huang, J. (eds.) *Brain Informatics. LNCS (LNAI)*, vol. 6334, pp. 29–41. Springer, Heidelberg (2010)
16. Hoogendoorn, M., Treur, J., van der Wal, C.N., van Wissen, A.: An Agent-Based Model for the Interplay of Information and Emotion in Social Diffusion. In: Proc. of the 10th IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2010, pp. 439–444. IEEE Computer Society Press, Los Alamitos (2010)
17. Iacoboni, M.: *Mirroring People*. Farrar, Straus & Giroux, New York (2008)
18. Iacoboni, M.: Understanding others: imitation, language, empathy. In: Hurley, S., Chater, N. (eds.) *Perspectives on imitation: from cognitive neuroscience to social science* vol. 1, pp. 77–100. MIT Press, Cambridge (2005)
19. Pan, X., Han, C., Dauber, K., Law, K.: Human and social behaviour in computational modeling and analysis of egress. *Automation in Construction* 15, 448–461 (2006)
20. Rizzolatti, G.: The mirror-neuron system and imitation. In: Hurley, S., Chater, N. (eds.) *Perspectives on imitation: from cognitive neuroscience to social science*, vol. 1, pp. 55–76. MIT Press, Cambridge (2005)
21. Rizzolatti, G., Craighero, L.: The mirror-neuron system. *Annu. Rev. Neurosci.* 27, 169–192 (2004)
22. Rizzolatti, G., Fogassi, L., Gallese, V.: Neuro-physiological mechanisms underlying the understanding and imitation of action. *Nature Rev. Neurosci.* 2, 661–670 (2001)
23. Rizzolatti, G., Sinigaglia, C.: *Mirrors in the Brain: How Our Minds Share Actions and Emotions*. Oxford University Press, Oxford (2008)
24. Pineda, J.A. (ed.): *Mirror Neuron Systems: the Role of Mirroring Processes in Social Cognition*. Humana Press Inc., Totowa (2009)
25. Striano, T., Reid, V.: *Social Cognition: Development, Neuroscience, and Autism*. Wiley-Blackwell (2009)

Visual Selective Attention Model Considering Bottom-Up Saliency and Psychological Distance

Young-Min Jang¹, Sang-Woo Ban², and Minho Lee¹

¹ School of Electronics 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
ymjang@ee.knu.ac.kr, swban@dongguk.ac.kr, mholee@knu.ac.kr

Abstract. Congruency or incongruity between a psychological distance and a spatial distance for given visual stimuli can affect reaction time to corresponding visual stimuli in the course of visual perception. Human reacts more rapidly for congruent stimuli than incongruent one. More rapid reaction to visual stimuli is related with higher selectivity property in terms of visual selective attention. Based on this psychological evidence, we propose a new visual selective attention model reflecting the congruency or incongruity between a psychological distance and a spatial distance of visual stimuli as well as bottom-up saliency generated by spatial relativity of primitive visual features. The proposed visual selective attention model can generate more human like visual scan path by considering a psychological factor, which can focus on congruent visual stimuli with higher priority than incongruent one.

Keywords: Affective saliency map, psychological distance, bottom-up saliency map, selective attention.

1 Introduction

Selective processing mechanism of human vision system is one of essential functions for efficiently perceiving very complex visual scene. Visual selective attention can provide important function of human vision system with such an efficient selective processing function, through which visual stimuli has different priority in the course of visual perception. Many different cues are involved in visual selective attention. Usually, visual selective attention may be categorized as bottom-up feature based attention and top-down goal oriented attention initiated by visual target object search or more complex perception functions such as human's preference, interesting, curiosity, intention, etc.. Most of proposed visual selective attention models are based on bottom-up feature based attention and top-down attention for finding a specific target object by mimicking the human-like selective attention functions.

Itti et al. introduced a brain-like model in order to generate a saliency map (SM) [1]. Lee et al. have also proposed a bottom-up SM model by using symmetry with an independent component analysis (ICA) filter [2].

On the other hand, according to Picard, human affective factors are involved in many aspects including decision and behavior as well as attention [3]. The human vision system includes an affective computing process to pay attention to a desired area by a preference and also to skip an unwanted area by refusal mechanism, which is mediated by the pulvinar [4]. Moreover, according to Bar-Anan et al., psychological distance, which is related with the way people think about stimuli, can affect stimuli perception and reaction [5]. They showed that a human would respond faster to congruent visual stimuli than incongruent one between psychological distance and spatial distance of corresponding visual stimuli. Congruency is occurred when the psychological distance and the spatial distance of visual stimuli are matched. And incongruency is occurred when the psychological distance and the spatial distance from the human are mismatched. Preferable visual stimuli such as friend and preferable objects are psychological proximal stimuli. On the other hand, non preferable stimuli such as enemy and disgust objects are psychological distal stimuli. Thus, in this paper, we propose a new visual selective attention model that reflects the psychological distance as well as the bottom-up feature based saliency. We verified the psychological distance evidence about visual objects stimuli by human vision reaction time and human eye scan path to the congruent and incongruent visual stimuli, which are measured by an Eye-Track system. This verification is the most important part of this paper in order to reflect psychological distance into visual selective attention model. The fuzzy adaptive resonant theory (ART) network is applied for learning the characteristics of visual objects, in which 4 features such as intensity, orientation, color and symmetry extracted from an input image are used as input feature representing an object. Fuzzy ART plays a role for categorizing trained objects into preferable (psychological proximal) one or disgust (psychological distal) one. Then, congruency between psychological distance and spatial distance is integrated with bottom-up saliency. Consequently, final saliency map reflecting psychological distance is generated. Based on this final saliency map, selective attention considering both the bottom-up features and the psychological effects is occurred. Such a visual selective attention model considering psychological distance can generate more human like attention, which can be utilized for a new human computer interface system reflecting human's psychological factors as well as an advanced intelligent robot vision system.

In Section 2, the proposed visual attention model is described in detail. And the experiments and conclusion follows in Section 3 and Section 4, respectively.

2 Visual Selective Attention Model

Figure 1 illustrates the proposed visual attention model, which is partly inspired by biological visual pathway from the retina to the visual cortex through the lateral geniculate nucleus (LGN) for bottom-up processing, which is extended to the inferior temporal area (IT) and prefrontal cortex (PFC) for top-down processing. In order to implement a visual selective attention function, three processes are combined to generate an affective SM. One generates a bottom-up SM. Second considers object categorizing and memory of the psychological proximal objects and distal objects.

Finally, an affective SM is constructed by considering the psychological distance.

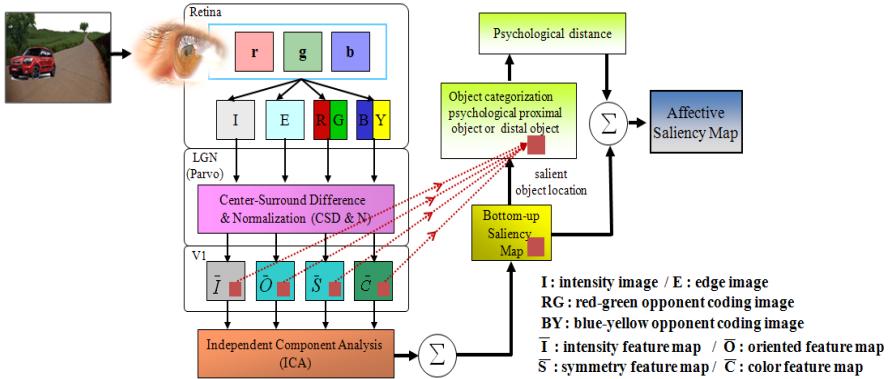


Fig. 1. The proposed visual selective attention model considering a psychological distance as well as primitive visual features

2.1 Bottom-Up Saliency Map (SM) Generation and Object Categorization

Based on Treisman's feature integration theory, Itti et al. used three basis feature maps based on intensity, orientation and color information for generating a bottom-up SM model [1,6]. By extending Itti et al.'s SM model, Lee et al. proposed SM models which included the neural network approach of Fukushima to construct the symmetry feature and an ICA filter designed to integrate the different feature information, which properly reflected the importance of the symmetry feature and an ICA filter in constructing an object preferred attention model [1,2,7,8]. In Fig. 1, the process to generate a SM from an input image is a bottom-up SM generation part, which follows Lee et al.'s bottom-up SM model [7]. Based on Barlow's hypothesis that human visual cortical feature detectors might be the end result of a redundancy reduction process, and Sejnowski's results showing that ICA is the best alternative to reduce redundancy [9,10]. The four constructed FMs (\bar{I} , \bar{O} , \bar{S} , and \bar{C}) are then integrated by means of an ICA algorithm based on maximization of the entropy. To obtain ICA filters, we derived the four conspicuity maps for intensity, color, orientation and symmetry information which are used for input patches of the ICA. An SM, $S(x,y)$, is obtained summing the convolution between each patch, $P(x,y)$, and the i -th filter (ICs_i) obtained by ICA learning, as shown in Eq. (1) [1,2].

$$S(x,y) = \sum_i P_{(x,y)} * ICs_i \quad (1)$$

In order to implement the visual selective attention model reflecting a psychological distance, we need a visual object categorization module, which can indicate whether a visual object is psychological proximal or distal. For this purpose, we implemented the object categorization models using the fuzzy ART network in conjunction with the bottom-up SM model. It is well known that the fuzzy ART model maintains the plasticity required to learn new patterns, while preventing the modification of patterns that have been learned previously [11]. Thus, the four conspicuity maps of a salient object selected by the bottom-up SM model are used as an input data of fuzzy ART

model that learns features of preferable objects (psychological proximal objects) and disgust objects (psychological distal objects) in a natural scene. There are two fuzzy ART networks. One is the fuzzy ART network for perceiving psychological proximal objects selected in the bottom-up SM model, and the other is the fuzzy ART network for perceiving psychological distal objects. The recognized object preference in psychological aspect by the fuzzy ART networks are used for deciding congruency and incongruency between psychological distance and spatial distance of the object in an input scene.

2.2 Selective Attention Reflecting Psychological Distance

Psychological distance refers to the distance of a stimulus (object or event) from the perceiver's direct experience, which is related to the way people think about stimuli since the distance between a stimulus and a perceiver's direct experience affects the perceiver's construal of the stimulus [5]. According to the psychological experiments, when congruency between a psychological distance and a spatial distance is occurred, people respond more rapidly to the corresponding stimulus than when incongruence is occurred [5]. Therefore, psychological distance can be a role for affecting perception and action as well as attention. Although the bottom-up SM model generates plausible salient areas and a scan path, the selected areas may not be appropriate since the bottom-up SM model only considers primitive features such as intensity, orientation, color and symmetry information. Therefore, in this paper, we reflect psychological factor such as psychological distance in order to get more human like selective attention .

In order to propose a new selective attention model reflecting the psychological distance, we conducted experiments measuring human eye scan path and visual response time using an Eye-Track system for verifying whether the psychological distance affects visual response about visual stimuli. Through this experiment, it is revealed that the psychological distance affects visual response time to the visual stimuli. In this experiment, pictures with clear spatial depth cues are considered, which follows the experimental methods conducted in [5]. In order to differently reflect the psychological distance according to congruent visual stimuli and incongruent one, we introduced a factor, $\text{Psycho_distance}(\mathcal{V})$, that is experimentally defined by the ratio between an average response time for congruent visual stimuli and that for incongruent visual stimuli [12]. Then, $\text{Psycho_distance}(\mathcal{V})$ becomes larger than 1 about congruent visual stimuli and becomes smaller than 1 about incongruent visual stimuli. When a final affective SM is generated by integrating an obtained psychological distance factor with the bottom-up SM, $\text{Psycho_distance}(\mathcal{V})$ value is used to magnify the bottom-up saliency at the congruent visual stimuli area in the input scene. Instead, $\text{Psycho_distance}(\mathcal{V})$ value is utilized for diminishing the bottom-up saliency at the incongruent visual stimuli area [12]. Finally, based on this affective SM, the proposed visual selective attention model decides an attention area with priority to focus on. In congruent visual stimuli, the congruent object area becomes more highly salient, which induces faster selection area in a visual scan path. On the other hand, in the case of incongruent condition, the incongruent object area becomes less salient, which induces slower selection area in a visual scan path.

3 Experimental Results

By replacing visual objects instead words in the Bar-Anan's experiments, we have shown that psychological distance can also affect the reaction time, which can cause changes of visual scan path. Thus, we applied such a psychological distance effect to our visual selective attention model. Figure 2 shows the examples of scenes used in the experiments, which are classified as congruent and incongruent scenes. In order to verify the relationship between a psychological distance and a spatial distance related with visual selective attention, we measured real human visual scan path for the pictures based on Bar-Anan's experiments as similar as possible except replacing words with visual objects for given stimuli. In Fig. 2, the white bear is a psychological proximal object and the odd ugly fish is a psychological distal object. Therefore, Figs. 2 (a) and (c) are examples of congruent visual stimuli since the psychological distance and the real distance are harmonious condition. Instead, Figs. 2 (b) and (d) are those of incongruent visual stimuli since the psychological distance and the real distance are different.



Fig. 2. Visual stimuli examples in the experiments; (a) and (c): congruent visual stimuli for psychological proximal and distal objects, (b) and (d): incongruent visual stimuli for psychological proximal and distal objects

3.1 Analysis of Human Response Time to Congruent and Incongruent Visual Stimuli

Our psychological distance experiments followed Bar-Anan's experimental method. We prepared for 40 images with the same background with clear spatial depth cues as shown in Fig. 2. Also, 8 healthy human subjects participated in experiments for measuring human visual scan path about each given visual stimulus, in which we utilized an eye-tracking system. Figure 3 shows measured visual scan path and fixation time plots on the given visual stimuli, which are measured by the eye-tracking system. Rainbow color from red line to purple line shows the sequence of human scan path in Figs. 3 (a) and (c). Fixation times are plotted using red circles on the image as shown in Figs. 3 (b) and (d), in which the size of red circle represents fixation time on that area. Bigger red circle means that human eye fixates at that area during longer time. The reaction time is obtained from the time duration between visual stimuli onset and firstly fixating time at a target object for visual stimulus.



Fig. 3. Examples of real human eye movement measurements using an eye tracking system; (a) and (b): visual incongruent stimuli, respectively, (c) and (d): visual congruent stimuli

In our experiments, congruent visual stimuli (psychologically proximal/spatial proximal, psychologically distal/spatial distal) generate faster response than incongruent conditions (psychologically distal/spatial proximal, psychologically proximal/spatial distal) as shown in Table 1, which are similar with Bar-Anan's experimental results even though using visual object stimuli and visual response instead of word stimuli and keyboard touching response used in Bar-Anan's experiments [5]. Figure 4 shows an error-bar graph using average reaction time and standard deviation for congruent trials and incongruent trials in the experiments using 40 images. According to our experiments, although some participants may show reverse average reaction time, the experimental results show that congruency effect is generally preserved in average.

Table 1. Mean response times (in ms) and standard deviations (SD) for all experiments including congruent and incongruent visual stimuli

Stimuli	spatial proximal		spatial distal	
	Mean	SD	Mean	SD
psychologically proximal	389 ms	20.66	449 ms	47.84
psychologically distal	452 ms	24.93	403 ms	26.10

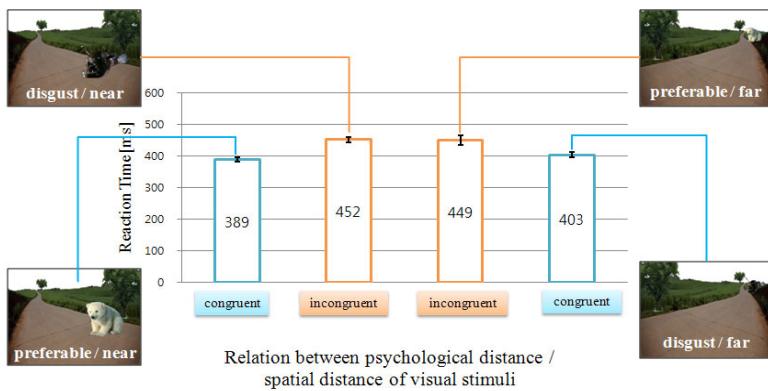


Fig. 4. Average reaction times for congruent and incongruent trials in all experiments

3.2 Visual Selective Attention Reflecting Both Bottom-Up Saliency and Psychological Distance

According to experimental proof that Bar-Anan's psychological distance can also be applicable for visual selective attention process as shown in Section 3.1, we proposed the visual selective attention model reflecting psychological distance as shown in Fig. 1. We have shown that the congruency or incongruity between psychological distance and spatial distance can affect to visual selective attention process. Figure 5 shows that congruent visual stimuli become more salient area after reflecting psychological distance. In the first example on the top row, congruent visual object, bear (psychological proximal object), is 2nd salient area when only considering bottom-up features as shown in the third image from left. However, after magnifying congruent object area by the proposed model owing to congruency, the bear becomes most salient area. In the second row of Fig. 5, congruent object also becomes more salient by reflecting the psychological factor.

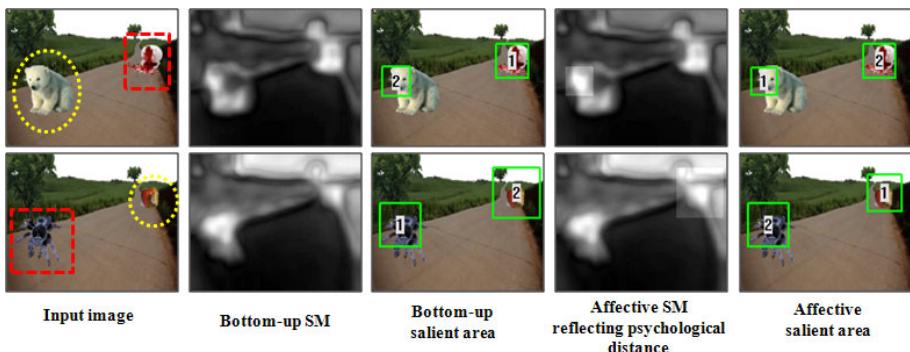


Fig. 5. Attention comparison for congruent/incongruent visual stimuli by the proposed visual selective attention model considering both psychological distance and bottom-up SM (yellow dotted circles: congruent visual stimuli, red dotted rectangles: incongruent visual stimuli)

4 Conclusion

We proposed a new visual selective attention model considering congruency and incongruency between a psychological distance and a spatial distance in conjunction with bottom-up saliency. Psychological factor can be important components to affect in deciding a visual selective attention area. The proposed selective attention model, which is motivated from Bar-Anan's psychological distance experiments, is a novel approach that considers psychological distance.

Through the eye-tracking experiments, we have revealed that human subject can respond faster to congruent visual stimuli than incongruent one. The proposed visual attention model can generate more human-like visual selective attention by a novel approach considering a psychological distance as well as saliency of primitive visual features.

We need more experiments using complex real scenes reflecting various situations in order to verify the performance of the proposed attention model as a further work.

Acknowledgments. This research was supported by the Converging Research Center Program through the Converging Research Headquarter for Human, Cognition and Environment funded by the Ministry of Education, Science and Technology (2010K001130) (50%) and also the Original Technology Research Program for Brain Science through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2010-0018845) (50%).

References

1. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 20(11), 1254–1259 (1998)
2. Park, S.-J., An, K.-H., Lee, M.: Saliency map model with adaptive masking based on independent component analysis. *Neurocomputing* 49(1), 417–422 (2002)
3. Picard, R.W.: *Affective computing*. MIT press, Cambridge (1997)
4. Ward, R., Calder, A.J., Parker, M., Arend, I.: Emotion recognition following human pulvinar damage. *Neuropsychologia* 45(8), 1973–1978 (2007)
5. Bar-Anan, Y., Trope, Y., Liberman, N., Algom, D.: Automatic Processing of Psychological Distance: Evidence From a Stroop Task. *J. Experimental Psychology: General* 136(4), 610–622 (2007)
6. Treisman, A.M., Gelde, G.: A feature-integration theory of attention. *Cognitive Psychology* 12(1), 97–136 (1980)
7. Jeong, S., Ban, S.-W., Lee, M.: Stereo saliency map considering affective factors and selective motion analysis in a dynamic environment. *Neural Networks* 21, 1420–1430 (2008)
8. Fukushima, K.: Use of non-uniform spatial blur for image comparison: symmetry axis extraction. *Neural Networks* 18(1), 23–32 (2005)
9. Barlow, H.B., Tolhurst, D.J.: Why do you have edge detectors? *Optical society of America: Technical Digest* 23, 172 (1992)
10. Bell, A.J., Sejnowski, T.J.: The ‘independent components’ of natural scenes are edge filters. *Vision Res.* 37(23), 3327–3338 (1997)
11. Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen, D.B.: Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks* 3(5), 698–713 (1992)
12. Ban, S.-W., Jang, Y.-M., Lee, M.: Affective saliency map considering psychological distance. *Neurocomputing* (to be published 2010)

Free-Energy Based Reinforcement Learning for Vision-Based Navigation with High-Dimensional Sensory Inputs

Stefan Elfwing, Makoto Otsuka, Eiji Uchibe, and Kenji Doya

Okinawa Institute of Science and Technology,
1919-1, Tancha, Onna-Son, Kunigami, Okinawa 904-0412, Japan
`{elfwing,otsuka,uchibe,doya}@oist.jp`

Abstract. Free-energy based reinforcement learning was proposed for learning in high-dimensional state and action spaces, which cannot be handled by standard function approximation methods in reinforcement learning. In the free-energy reinforcement learning method, the action-value function is approximated as the negative free energy of a restricted Boltzmann machine. In this paper, we test if it is feasible to use free-energy reinforcement learning for real robot control with raw, high-dimensional sensory inputs through the extraction of task-relevant features in the hidden layer. We first demonstrate, in simulation, that a small mobile robot could efficiently learn a vision-based navigation and battery capturing task. We then demonstrate, for a simpler battery capturing task, that free-energy reinforcement learning can be used for online learning in a real robot. The analysis of learned weights showed that action-oriented state coding was achieved in the hidden layer.

1 Introduction

Reinforcement learning [1] has been proven to be effective for a wide variety of delayed reward problems. However, standard reinforcement learning algorithms can not handle high-dimensional state and action spaces. For standard action-value function approximators, such as tile coding and radial basis function networks, the number of features of the function approximator grows exponentially with the dimension of the state and action spaces.

Sallans and Hinton [2] proposed free-energy based reinforcement learning as function approximation method to handle high-dimensional state and action spaces. In their method, the action-value function is approximated as the negative free energy of a restricted Boltzmann machine. The action coding of the nodes in the hidden layer in the restricted Boltzmann machine clusters different sensory input for which the same action is appropriate. Otsuka et al. [3] extended the free-energy reinforcement learning method to handle partially observable Markov decision processes, by incorporating a recurrent neural network that learns a memory representation that is sufficient for predicting future observations and rewards.

In this paper, we test if it is feasible to use free-energy reinforcement learning for real robot control with raw, high-dimensional, sensory inputs through the extraction of task-relevant features in the hidden layer. We first test, in simulation, whether free-energy based reinforcement learning, with the introduction of truncated eligibility traces, can efficiently learn a vision-based navigation and battery capturing task using high-dimensional sensory inputs. We, then, test if free-energy based reinforcement learning can be used for on-line learning in a real robot in a simpler battery capturing task.

A similar approach to vision-based navigation with raw camera images was proposed by Shibata et al. They used a multi-layered neural network as function approximator and demonstrated that a mobile robot could learn to push a box [4] and to approach and “kiss” another robot [5].

2 Method

2.1 Gradient Descent Sarsa(λ)

Sarsa(λ) [6] [7] is an on-policy reinforcement learning algorithm that learns an estimate of the action-value function, Q^π , while the agent follows policy π . In this study, we use gradient descent function approximation of the Q -values. If the approximated action value function, $Q_t \approx Q^\pi$, is parameterized by the parameter matrix Θ_t , then it is updated according to

$$\Theta_{t+1} = \Theta_t + \alpha[r_t + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)]e_t. \quad (1)$$

Here, s_t is the state at time t , a_t is the action selected at time t , r_t is the reward for taking action a_t in state s_t , α is the learning rate, and γ is the discount factor of future rewards, and e_t is the eligibility trace matrix. In this study, we use truncated replacing traces [8] which are defined as

$$e_t(s, a) = \begin{cases} \frac{\partial Q_t(s_t, a_t)}{\partial \Theta_t(s_t, a_t)} & \text{if } s = s_t \text{ and } a = a_t; \\ \gamma \lambda e_{t-1}(s_{t-i}, a_{t-i}) & s = s_{t-i}, a = a_{t-i}, \text{ and } i < N_e; \text{ for all } s, a. \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Here, λ is the trace-decay rate and N_e is the number of time steps the reward is propagated back along the trajectory of state-action pairs, decaying by $\gamma\lambda$ per time step.

2.2 Free-Energy Based Function Approximation

A restricted Boltzmann machine consists of binary state nodes, \mathbf{s} , binary action nodes \mathbf{a} , and hidden nodes, \mathbf{h} . The i th state node, s_i , is connected to hidden node h_k by the weight w_{ik} , and j th action node, a_j , is connected to hidden node h_k by the weight u_{jk} . The Q -values are approximated by the negative free energy, F , of the Boltzmann machine, i.e., $Q(\mathbf{s}, \mathbf{a}) = -F(\mathbf{s}, \mathbf{a})$. The free energy is given as

$$F(\mathbf{s}, \mathbf{a}) = -\sum_{k=1}^K \left(\sum_{i=1}^{N_s} w_{ik} s_i h_k + \sum_{j=1}^{N_a} u_{jk} a_j h_k \right) + \sum_{k=1}^K h_k \log h_k + (1-h_k) \log(1-h_k). \quad (3)$$

Here K is the number of hidden nodes, N_s is the number of state nodes, N_a is the number of action nodes, and h_k is the activation of the k th hidden node. The hidden node activation is given as

$$h_k = \sigma \left(\sum_{i=1}^{N_s} w_{ik} s_i + \sum_{j=1}^{N_a} u_{ik} a_j \right), \text{ where } \sigma(x) = \frac{1}{1 + e^{-x}}. \quad (4)$$

For learning of the action-value function by Equation 1, the eligibility traces in Equation 2 can be updated using the derivatives of the approximated Q -values, i.e., the negative free energy, with respect to the weights (w_{ik} and u_{jk}) as

$$\frac{\partial Q_t(\mathbf{s}, \mathbf{a})}{\partial w_{ik}} = s_i h_k, \quad \frac{\partial Q_t(\mathbf{s}, \mathbf{a})}{\partial u_{ik}} = a_j h_k. \quad (5)$$

We use softmax action selection with a Boltzmann distribution, where the probability to select action a in state \mathbf{s} is defined as

$$P(a|\mathbf{s}) = \frac{\exp(Q(\mathbf{s}, a)/\tau)}{\sum_A \exp(Q(\mathbf{s}, b)/\tau)}. \quad (6)$$

Here, τ is the temperature that controls the trade-off between exploration and exploitation, and A is the set of permissible actions, in which only one of the binary nodes are active.

3 Experiments

The Cyber Rodent robot (see [9] for details) used in the experiments is a small mobile robot. The simulation environment was developed, in Matlab, to mimic the properties of the real robot.

3.1 Simulation Experiment

Figure 1 shows the experimental setup for the simulation experiment. There were two task settings, the battery pack was either initially placed close to the northwest corner (task setting 1) or close to the southwest corner (task setting 2). In the simulation, the battery can be moved by a collision with the robot. In the environment, the robot's camera system could detect eight different colored objects (see Figure 1(c) for an example camera image): 1) the battery pack (red); 2) obstacles (yellow); and 3) four Landmarks, where the lower parts were intended for global navigation (orange, blue, green, and magenta) and the upper parts identified the task setting (white for task setting 1 and black for task setting 2). The input state to the learning algorithm consisted of eight binary camera images, i.e., 5904 binary states, which were constructed by extracting the pixels that detected each of the colors representing the eight different objects in the original camera image. In addition, the input state also consisted of three

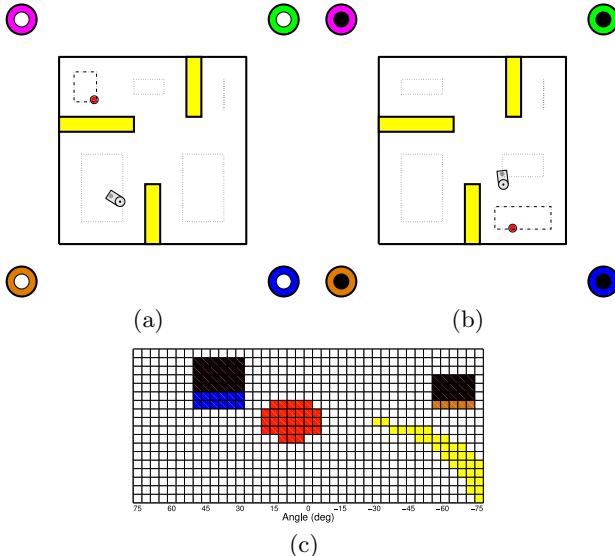


Fig. 1. (a) The simulation environment for task setting 1, (b) task setting 2, (c) the 41×18 simulated camera image corresponding to the robot’s position in (b). The dotted rectangles indicate the four areas in which the robot was randomly located at the start of possible initial positions of the battery pack.

normalized ($[0, 1]$) real-valued distance measures from the robot’s front proximity sensors, located at -30° , 0° , and $+30^\circ$ in relation to the robot’s heading direction.

The robot could execute 5 actions, pairs of velocities (mm/s) of the left and the right wheels: 1) turn right $(200, -200)$; 2) curve right $(400, 200)$; 3) go straight $(300, 300)$; 4) Curve left $(200, 400)$; and 5) turn left $(-200, 200)$. In the simulations, Gaussian noise was added to each wheel velocity with zero mean and a standard deviation equal to 1 % of the amplitude of the velocity. The temperature, τ , decreased hyperbolically according to: $\tau = \tau_{init}/(1 + k_{tau}n_{epi})$. Here, the initial temperature, τ_{init} , was set to 0.5, k_{tau} was set to 0.02, and n_{epi} was the number of episodes. The values of the other parameters were set as follows: time step: 200 ms, α : 0.005, γ : 0.98, λ : 0.6, N_e : 12, and K : 20. An episode ended either when the robot captured the battery pack or when the length of the episode exceeded a threshold of 2000 time steps. The robot received a reward of 1 when it captured the battery pack, otherwise the reward was set to 0.

Figure 2 shows a typical learning result for a simulation run (top figures) as the average number of time steps to capture the battery pack over every 20 episodes, and examples of learned trajectories (bottom figures) for the two task settings. In this simulation run, the learning in task setting 2 was similar for all four starting position areas and the learning converged after approximately 1500 episodes to a performance level similar to the final performance level. The learning in task setting 1 was slower and more unstable. For three of the starting position areas,

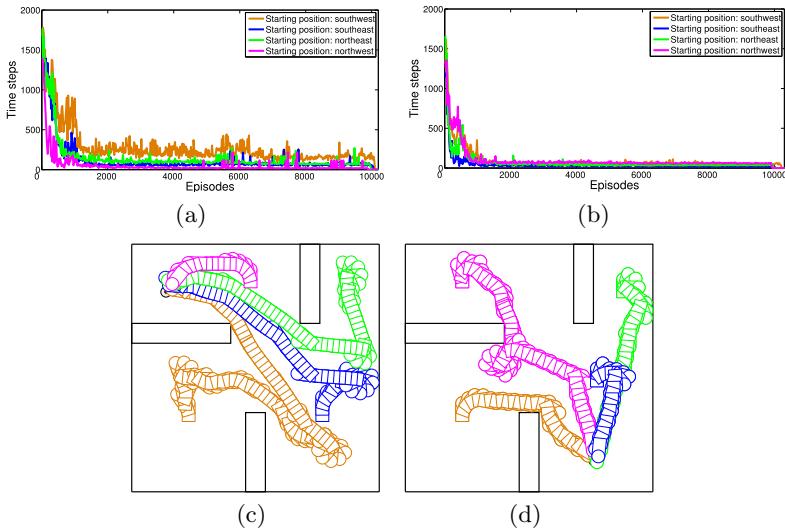


Fig. 2. Learning results for the four starting position areas in task setting 1 (a) and task setting 2 (b), and examples of learned trajectories for the task setting 1 (c) and task setting 2 (d). The top graphs show the average number of time steps to capture the battery pack over every 20 time episodes.

the learning converged after approximately 2500 episodes. For the fourth starting position area, i.e., when the robot was initially positioned close to southwest corner, the learning converged more slowly and reached a more stable level only after approximately 8000 time steps (see the orange curve in the Figure 2(a)). The example trajectory clearly illustrate the reason for the sub-optimal learning (orange trajectory in the bottom left figure). After leaving the southwest quadrant, the robot turned right and moved almost all the way down to the southeast corner before turning around and starting to move towards the battery pack. For the other seven starting positions, the learned trajectories were near-optimal.

Figure 3 shows the action coding in the hidden nodes for the state-action pairs along the green trajectory in task setting 1 in Figure 2(c) and the orange trajectory in task setting 2 in Figure 2(d). The learned action coding was very sparse and non-redundant, i.e., for a selected action only one, or a few, of the hidden nodes were active. This is shown most clearly for the action coding in task setting 2 (Figure 3(b)). For each of the 3 selected actions along the trajectory only one hidden node was active at each time step. For task setting 1 (Figure 3(a)), the action coding was less distinct and in many states several hidden nodes were activated for the same action.

3.2 Hardware Experiment

The experimental setup of the hardware experiment is shown in Figure 4(a). The battery pack was placed in the middle of the 1.8×1.8 m experimental area.

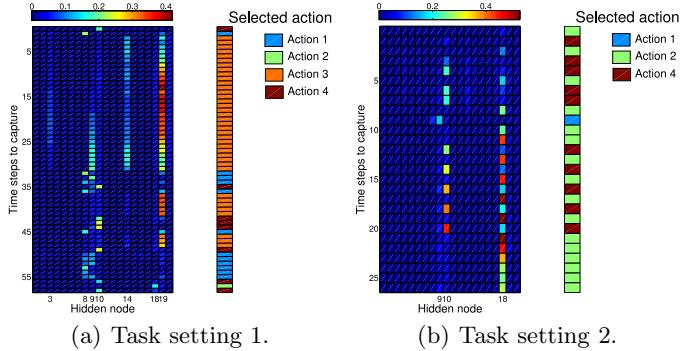


Fig. 3. Action coding in the hidden nodes. The figures show the activation of the hidden nodes, h_k , and the selected greedy actions for the state-action pairs along the trajectory from the starting position to the capturing of the battery pack. The left figure corresponds to the green trajectory in Figure 2(c) and the right figure corresponds to the orange trajectory in Figure 2(d).

At the beginning of each episode the robot faced the battery at a distance of approximately 50 cm. The state consisted of a 24×12 binarized camera image (Figure 4(b)), constructed from the 240×120 original image by mapping each 10×10 patch to one binary state. In addition, the state consisted of three binary states which were active if the robot lost vision of battery: 1) to the left within 3 time steps, 2) to the right within 3 time steps, and 3) for more than 3 time steps.

The parameters were set as follows: time step: 200 ms, α : 0.01, γ : 0.98, τ_{init} : 0.25, k_{tau} : 0.1, and K : 20. The robot received a reward of 1 when it captured the battery and a -0.1 reward if it lost visibility of the battery, otherwise the reward was set to 0. We ran the experiment for 500 episodes and an episode ended either when the robot captured the battery pack or when the length of the episode exceeded 500 time steps. In order to validate the learning performance in hardware, we also conducted a simulation experiment with similar setup. In the simulation, we used camera images similar to the earlier experiment (see Figure 1(c)), but with the same resolution as the binarized hardware image (24×12) and with a field of $\pm 45^\circ$.

The learning curves (Figure 4(c)) were comparable in both hardware and in simulation, and there was no significant difference in mean values over the final 50 episodes: 31.7 ± 45.0 time steps to capture the battery in the hardware experiment and 35.6 ± 59.8 time steps in simulation. The results suggest that it is feasible to use on-line free-energy based reinforcement learning to learn tasks with high-dimensional visual state inputs in real robotic platforms. The main limitations in executing the learning algorithm on the cyber rodent robot is the weak performance of the FPU of the robot's on-board computer (Hitachi SH-4). For example, the calculation of the activation of the hidden nodes and the free energy (see Equations 4 and 3) took approximately 105 ms in hardware,

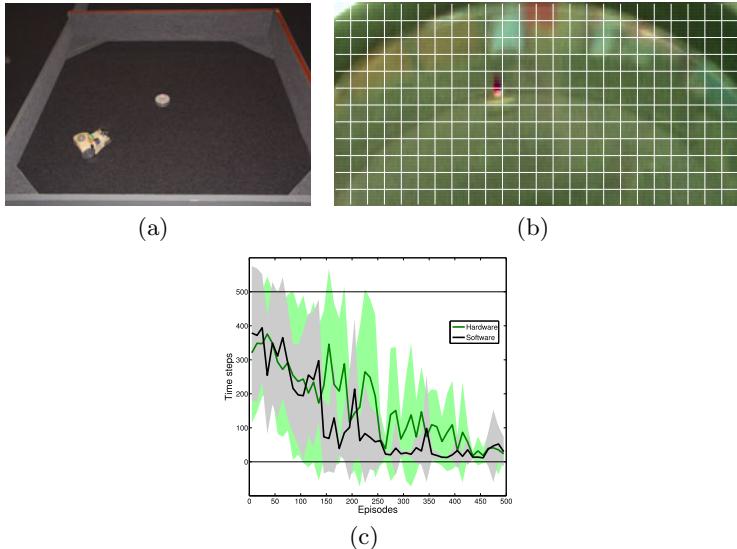


Fig. 4. (a) The hardware experimental setup, (b) the camera image, (c) the average number of time steps, over every 10 episodes, to capture the battery in hardware (green curve), with standard deviation, compared with the learning performance in simulation (black curve)

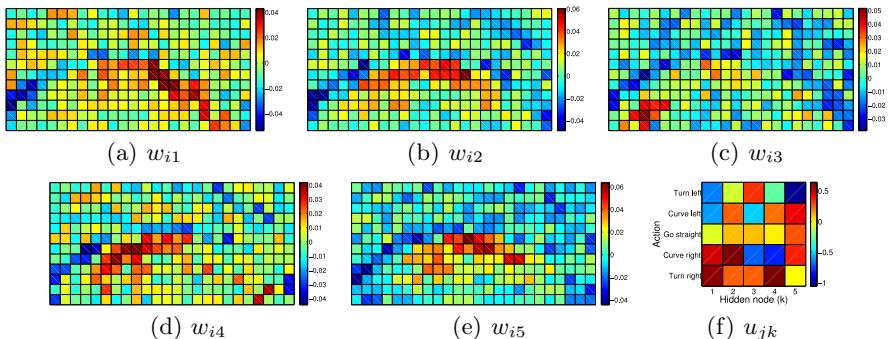


Fig. 5. The learned weights connecting from the state nodes to the hidden nodes, w_{ik} , and from action nodes to hidden nodes, u_{jk}

compared to approximately 0.15 ms in the simulation, running on a Dell precision T7400 with an Intel Xeon(R) CPU with a clock frequency of 2.66 GHz. Figure 5 visualizes the learned hidden layer weights in the hardware experiment. Because of the robot's omnidirectional camera, the positions of the pixels that could detect the battery's LED were along a skewed half circle. The visualized weights show how the restricted Boltzmann machine performed state and action abstraction by clustering state inputs that corresponded to similar actions.

4 Conclusions

In this paper, we demonstrated, in simulation, that free-energy based reinforcement learning can be used for efficient learning in high-dimensional state spaces, using a vision-based navigation and battery capturing task. The learned action-coding in the hidden layer of the restricted Boltzmann machine was very sparse and non-redundant. We also demonstrated, in a simpler battery capturing task, that free-energy reinforcement learning can be used for on-line learning in real robots, and that action-oriented state coding is achieved in the hidden layer.

References

1. Sutton, R.S., Barto, A.: Reinforcement Learning: An Introduction (1998)
2. Sallans, B., Hinton, G.E.: Reinforcement learning with factored states and actions. *Journal of Machine Learning Research* 5, 1063–1088 (2004)
3. Otsuka, M., Yoshimoto, J., Doya, K.: Free-energy-based reinforcement learning in a partially observable environments. In: Proceedings of ESANN, pp. 541–545 (2010)
4. Shibata, K., Iida, M.: Acquisition of box pushing by direct-vision-based reinforcement learning. In: Proceedings of SICE 2003, vol. 3, pp. 2322–2327 (2003)
5. Shibata, K., Kawano, T.: Learning of action generation from raw camera images in a real-world-like environment by simple coupling of reinforcement learning and a neural network. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008. LNCS, vol. 5506, pp. 755–762. Springer, Heidelberg (2009)
6. Rummery, G.A., Niranjan, M.: On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University (1994)
7. Sutton, R.S.: Generalization in reinforcement learning: Successful examples using sparse coarse coding. In: NIPS 1995, pp. 1038–1044 (1996)
8. Singh, S.P., Sutton, R.S.: Reinforcement learning with replacing eligibility traces. *Machine Learning* 22(1-3), 123–158 (1996)
9. Doya, K., Uchibe, E.: The cyber rodent project: Exploration of adaptive mechanisms for self-preservation and self-reproduction. *Adaptive Behavior* 13(2), 149–160 (2005)

Dependence on Memory Pattern in Sensitive Response of Memory Fragments among Three Types of Chaotic Neural Network Models

Toshiyuki Hamada, Jousuke Kuroiwa, Hisakazu Ogura, Tomohiro Odaka,
Haruhiko Shirai, and Izumi Suwa

Graduate School of Engineering, University of Fukui,
3-9-1, Bunkyo, Fukui, Fukui, 910-8507, Japan
`{hamada, jou, ogura, odaka, shirai, suwa}@ci-lab.jp`

Abstract. In this paper, we investigate the dependence on the size and the number of memory pattern in the sensitive response to memory pattern fragments in chaotic wandering states among three types of chaotic neural network (CNN) models. From the computer experiments, the three types of chaotic neural network model show that the success ratio is high and the accessing time is short without depending on the size and the number of the memory patterns. The feature is introduced in chaotic wandering states with weaker instability of orbits and stronger randomness in memory pattern space. Thus, chaos in the three model is practical in the memory pattern search.

Keywords: Instability of orbits, memory pattern fragments, CNN, memory search.

1 Introduction

From the theoretical approach, in complex memory search of neural network model with multi-cycle memory patterns, Nara & Davis presented th interesting results that the chaotic wandering states in chaotic neural network model instantaneously converge into a basin of the attractor of the target memory pattern when its fragment is presented [1][2][3]. In addition, Kuroiwa & Nara also presented the interesting results that the property of the sensitive response to memory pattern fragments in chaotic wandering state is general even though the mechanism of chaos is different [4]. Skarda and Freeman have shown that chaos could play the important roles in a learning process and a recalling process[5]. In a learning process, chaos could provide driving activity essential for memorizing novel inputs. In a recalling process, chaos ensure rapid and unbiased access to previously trained patterns. Thus, the sensitive response to memory pattern fragments with chaotic wandering states could play important roles in realizing the rapid and unbiased access.

In this paper, we investigate the dependence of the size and number of memory patterns in sensitive response to memory pattern fragments among three types

of chaotic neural network models. In this paper, we focus on the following three types of chaotic neural network model:

Aihara model: The existence of relative refractoriness and the continuity of the output function introduce chaotic wandering state in system even if the system is an isolated neuron element [7].

Nara & Davis model: By reducing the connectivity of synaptic connections, chaotic wandering state is induced [1].

Kuroiwa & Nara model: By partly inverting synaptic connections, chaotic wandering state appears [6]

It has been shown that the three types of chaotic neural network models reveal similar dynamical features of chaotic wandering states in the space of the memory patterns. In addition, the three models in the chaotic wandering states have higher potentiality of memory searching than random searching even if the mechanism of deriving chaotic dynamics is different. The three models show the high success ratio and the short accessing time when the system is in chaotic wandering states whose instability of orbits becomes weaker and randomness in memory pattern space is stronger. However, it has been still unknown whether the above results general or not for the size and the number of memory patterns. The purpose of this paper is to investigate the dependence of the size and the number of memory patterns in sensitive response to memory pattern fragments among three types of chaotic neural network models.

2 Chaotic Neural Network Model

2.1 Recurrent Neural Network Model with Associative Memory

Let us explain a recurrent neural network model, briefly. The updating rule of the recurrent neural network model is given as follows:

$$u_i(t+1) = \sum_{j=1}^N w_{ij} z_j(t), \quad (1)$$

where $u_i(t)$ represents an internal state of the i th element at discrete time t , $z_i(t)$ describes its output, w_{ij} gives a synaptic connection between the i th element and the j th element, and N is the total number of elements in the recurrent neural network model. The output is given by the following output function,

$$z_i(t+1) = \tanh(\beta u_i(t+1)), \quad (2)$$

where β corresponds to the steepness of the output function.

In this paper, the synaptic connection is defined as,

$$w_{ij} = \sum_{a=1}^L \sum_{\mu=1}^P v_i^{a \mu+1} (v_j^{a \mu})^\dagger, \quad (3)$$

where $\mathbf{v}^{a \mu}$ denotes the a th cycle and μ th memory pattern vector with $\mathbf{v}^{a P+1} = \mathbf{v}^{a 1}$, L and P represent the number of cycles and the number of patterns per cycle, respectively. The dagger vector of $(\mathbf{v}^{a \mu})^\dagger$ is given by

$$(\mathbf{v}^{a \mu})^\dagger = \sum_{b=1}^L \sum_{\nu=1}^P (\mathbf{o}^{-1})_{a\mu b\nu} \mathbf{v}^{b \nu}, \quad (4)$$

where \mathbf{o}^{-1} is the inverse of the \mathbf{o} defined by

$$(\mathbf{o}_{a\mu b\nu}) = \sum_{k=1}^N v_k^{a \mu} v_k^{b \nu}. \quad (5)$$

Thus, according to Eq.(3), L cycles and P patterns per cycle are embedded in the recurrent neural network model.

2.2 Aihara Model

In Aihara model[7], the updating rule is given by

$$u_i(t+1) = ku_i(t) + \sum_{j=1}^N w_{ij} z_j(t) - \alpha z_i(t) + A_i, \quad (6)$$

where k controls a decay effect, $\alpha z_i(t)$ denotes a relative refractoriness of the i th element, and A_i represents a constant bias input. The output of $z_i(t)$ is given by Eq.(2). The system parameters are k , α , A_i and β .

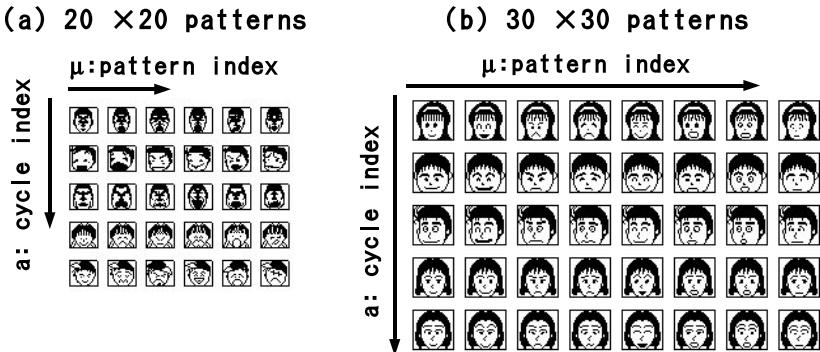


Fig. 1. (a)Memory patterns in [1]. The number of cycles is $L = 5$, the number of patterns per cycle is $P = 6$, and each pattern consists of 20×20 pixels which take ± 1 . (b)Memory patterns in [6]. The number of cycles is $L = 5$, the number of pattern per cycle is $P = 8$, and each pattern consists of 30×30 pixels which take ± 1 .

2.3 Nara & Davis Model

In Nara & Davis model, the updating rule is represented by,

$$u_i(t+1) = \sum_{j=1}^N w_{ij} \epsilon_{ij}(d) z_j(t), \quad (7)$$

where

$$\epsilon_{ij}(d) = \begin{cases} 0 & (j \in F_i(d)) \\ 1 & (\text{otherwise}), \end{cases} \quad (8)$$

where $F_i(d)$ represents a configuration of element's positions at which the incoming synaptic connection is inhibited, and d is the connectivity of remaining synaptic connection, that is, $\sum_j \epsilon_{ij}(d) = d$. In this model, the connectivity d and β are system parameters.

2.4 Kuroiwa & Nara Model

In Kuroiwa & Nara model, the updating rule is written by

$$u_i(t+1) = \begin{cases} \sum_{j=1}^N w_{ij} \epsilon_{ij} z_j(t) & (t = \text{even}) \\ \sum_{j=1}^N w_{ij} z_j(t) & (t = \text{odd}), \end{cases} \quad (9)$$

where

$$\epsilon_{ij}(r) = \begin{cases} -1 & (j \in F_i(r)) \\ 1 & (\text{otherwise}), \end{cases} \quad (10)$$

and $F_i(r)$ defines a configuration of element's positions at which a value of incoming synaptic connections is partly inverted with the number of r . In this model, the inverted number of r and β are system parameters.

3 Dependence of Memory Pattern in Sensitive Response to Memory Pattern Fragments

3.1 The Purpose and the Method of Experiments

We investigate the dependence of the size and the number of memory patterns in sensitive response to memory pattern fragments among three types of chaotic neural network models. In computer experiments, we apply a certain fragment of a memory pattern as shown in Fig. 2 to the chaotic neural network models in chaotic wandering states, and we focus on the following two problems.

1. Success ratio: How many times does it reach the target basin within 30 iteration steps starting from different points of chaotic wandering state? In other words, is the searching procedure by means of chaotic wandering assured?

2. Accessing time: How many steps does it take to reach the target basin related with a memory fragments? In other words, how short is the “access” time for the memory basin corresponding to the external input of a memory fragments?

In applying a memory pattern fragments as shown in Fig. 2, the updating rule is rewritten of as follows:

Aihara model

$$u_i(t+1) = \sum_{j=1}^N w_{ij} z_j(t) + k u_i(t) - \alpha z_i(t) + A_i + c \rho I_i \quad \text{if } i \in F \quad (11)$$

Nara & Davis model

$$u_i(t+1) = \sum_{j=1}^N w_{ij} \epsilon_{ij} z_j(t) + c \rho I_i \quad \text{if } i \in F \quad (12)$$

Kuroiwa & Nara model

$$u_i(t+1) = \begin{cases} \sum_{j=1}^N w_{ij} \epsilon_{ij} z_j(t) + c \rho I_i & \text{if } i \in F \quad (t = \text{even}) \\ \sum_{j=1}^N w_{ij} z_j(t) + c \rho I_i & \text{if } i \in F \quad (t = \text{odd}) \end{cases} \quad (13)$$

The set of $\{I_i\}$ represents a memory pattern fragment, F denotes the configuration of its position, ρ denotes the strength of the input, and c denotes the normalization factor which corresponds to the standard deviation of $\{u_i(t)\}$. In the memory patterns of Fig. 2(a), fragments correspond to Fig. 2(a), which are composed of 40 pixels (40/400). In the memory patterns of Fig. 2(b), on the other hand, fragments correspond to Fig. 2(b), which are composed of 50 pixels (50/900). If one applies different memory patterns, the system parameter dependence of the chaotic wandering state would change. In order to investigate the similarity of the sensitive response, we employ the system parameters as given in

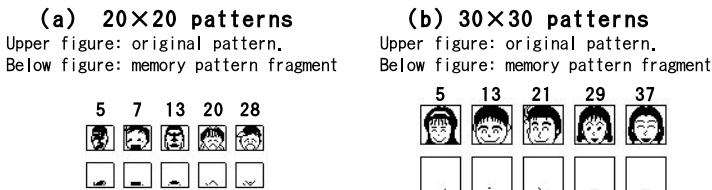


Fig. 2. Memory pattern fragments. In both figures (a) and (b), lower figures show the memory pattern fragment corresponding to each face pattern in the upper figures.

Table 1. System parameter set

model	In Fig.1(a) and Fig.2(a)	In Fig.1(b) and Fig.2(b)
Aihara model	$k = 0.9, \alpha = 1.2, A = 0.1$ $c = 0.87, \rho = 9$	$k = 0.9, \alpha = 1.2, A = 0.1$ $c = 0.74, \rho = 9$
Nara & Davis model	$d = 340, c = 0.17, \rho = 9$	$d = 847, c = 0.09, \rho = 9$
Kuroiwa & Nara model	$r = 214, c = 0.45, \rho = 9$	$r = 437, c = 0.26, \rho = 9$

Table 1, where dynamical features of the chaotic wandering state become similar from the viewpoint of the randomness in the memory pattern space and the instability of orbits. In evaluation of success ratio and accessing time, we identify that the searching procedure is succeeded if the system converge into the target pattern of the fragment with 30 steps while the fragment has been presented. On the other hand, we regard that the procedure is missing if the system does not converge within 30 steps. In addition, in the calculation of accessing time, we regard the accessing time as 30 steps when the procedure is missing. Finally, we evaluate the success ration and accessing time with 1000 different starting points in of the chaotic wandering state.

3.2 Results

Results of success ration and accessing time are given in Table 2, respectively. From the results, three models show much high success ratio and short accessing time without depending on the memory patterns from the random searching. Especially, Nara & Davis model is quite small. At next, we investigate the meaning of the employed system paramters from the viewpoint of the dynamical feature in memory pattern space and the instability of the orbit. In other word, we study why the feature of the sensitive response to the memory pattern fragments is general. We calculate visiting measure, entropy of visiting measure, Lyapunov dimension and Information dimension. Visiting measure evaluate the localization of the chaotic orbit in the memory pattern space [1]. The unbiased distribution of visiting measure means that the degree of chaos is high. The larger entropy of visiting measure show more unbiased distribution of visiting measure. Lyapunov dimension D_L and Information dimension D_I evaluate the instability of the chaotic orbit. The larger Lyapunov dimension D_L and Information dimension D_I reveal that the instability of orbits becomes higher.

Table 2. Success raion and accessing time in accessing to the target memory pattern

model	20 × 20 patterns		30 × 30 patterns	
	Success ratio	Accessing time	Success ratio	Accessing time
Aihara model	99%	5.1	99%	5.8
Nara & Davis model	100%	2.2	100%	2.0
Kuroiwa & Nara model	100%	5.6	99%	3.6
Random search	20%	24.4	8%	27.7

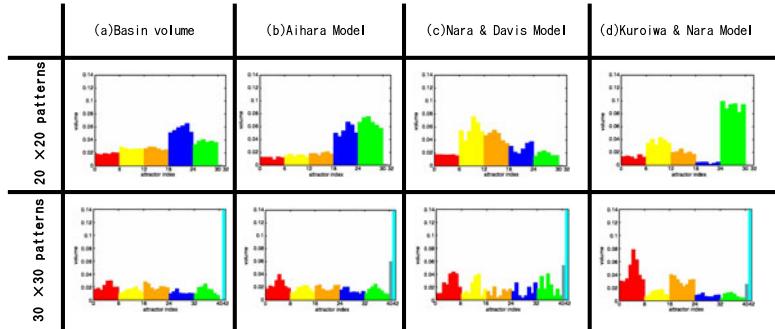


Fig. 3. The relative visiting measure to the basins of the attractors of memory patterns when the system reveals the high potentiality in sensitive response to memory pattern

Table 3. Instability of orbits and unbiased degree when the system reveals the high potentiality in sensitive response to memory pattern. H is entropy evaluated from visiting measure which represents unbiased degree, D_L is Lyapunov dimension and D_I is Information dimension.

model	20 × 20 patterns				30 × 30 patterns			
	Success ratio	H	D_L	D_I	Success ratio	H	D_L	D_I
Aihara model	99%	3.17	203	1.31	99%	3.40	590	1.19
Nara & Davis model	100%	3.28	-	0.89	100%	3.32	-	0.62
Kuroiwa & Nara model	100%	3.03	12.7	1.11	100%	3.36	10.4	0.95

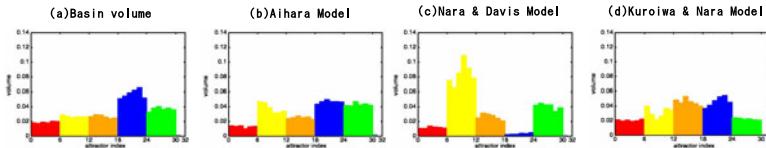


Fig. 4. The relative visiting measure to the basins of attractors of memory patterns with memory pattern in Fig II(a) when the system becomes poor potentiality in sensitive response to memory pattern fragments

Results are given in Fig 3, Table 3, Fig 4 and Table 4. In the parameter of Fig 3 and Table 3, the system reveals high potentiality in sensitivity. In the parameter of Fig 4 and Table 4, on the other hand, the potentiality of the system becomes poor slightly. From the results, for the high performance parameters set, the distribution of visiting measure tend to be unbiased and the instability of the orbit tend to be weaker. In fact, Lyapunov dimension and Information dimension for the high performance parameter set become smaller than the poor performance one. The chaotic wandering state with the unbiased distribution of visiting measure enable to realize unbiased access to memory patterns.

Table 4. Instability of orbits and unbiased degree when the system reveals the poor potentiality in sensitive response to memory pattern. H is entropy evaluated from visiting measure which represents unbiased degree, D_L is Lyapunov dimension and D_I is Information dimension.

model	Success ratio	20 × 20 patterns		
		H	D_L	D_I
Aihara model	90%	3.36	400	2.97
Nara & Davis model	92%	3.03	-	0.92
Kuroiwa & Nara model	99%	3.34	14.5	1.08

The chaotic wandering state with weak instability enable to rapidly response to inputs. Thus, the feature of unbiased distribution and weak instability is important to realize the sensitive response to memory pattern fragments.

4 Conclusion

In this paper, we investigate the dependence on the number and the size of memory patterns in sensitive response to memory pattern fragments among three types of chaotic neural network models. The three types of chaotic neural network model in the chaotic wandering state show the high sensitive response to memory pattern fragments without depending on the memory pattern when the chaotic wandering state is strong and the instability of orbits is weak. Thus, in memory search, chaotic wandering states in its is effective in rapidly converging into the target memory pattern with fragments.

References

1. Nara, S., Davis, P., Kawachi, M., Totsuji, H.: Chaotic Memory Dynamics in a Recurrent Neural Network with Cycle Memories Embedded by Pseudoinverse Method. *J. Bifurcation & Chaos* 5, 1205–1212 (1995)
2. Nara, S.: Can potentially useful dynamics to solve complex problems emerge from constrained chaos and /or chaotic itinerancy? *Chaos* 13, 1110–1121 (2003)
3. Mikami, S., Nara, S.: Dynamical response of chaotic memory dynamics to weak input in a recurrent neural network model. *Neural Comput. Appl.* 11, 129–136 (2003)
4. Kuroiwa, J., Masutani, S., Nara, S., Aihara, K.: Sensitive responses of chaotic wandering to memory pattern fragment inputs in a chaotic neural network model. *Int. J. Bifurcation & Chaos* 14, 1413–1421 (2004)
5. Skarda, S.A., Freeman, J.K.: How brains make chaos in order to make sense of the world? *Behavioral and Brain Sci.* 10, 161–195 (1987)
6. Nakayama, S., Kuroiwa, J., Nara, S.: Partly Inverted Synaptic Connections and Complex Dynamics in a Symmetric Recurrent Neural Network Model. In: Proceedings of international joint conference on neural network (ICONIP 2000), Taejon, Korea, vol. 2, pp. 1274–1279 (2000)
7. Aihara, K., Takabe, T., Toyoda, M.: Chaotic Neural Networks. *Phys. Lett. A* 144, 333–339 (1990)

A Stimulus-Response Neural Network Model Prepared by Top-Down Signals

Osamu Araki

Tokyo University of Science, Department of Applied Physics,
1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan
<http://www.rs.kagu.tus.ac.jp/app-phys/en/index.htm>

Abstract. The stimulus-response circuits in the brain need to have flexible and fast characteristics and these circuits should be activated during the preparatory period for movement. We propose a fundamental neural network model, which can trigger the movement in response to a specific sensory input using top-down signals. When the top-down signal is received in the 1st layer, this circuit waiting for the specific sensory input is in the ready state to move. In response to the specific sensory input, synchrony is emitted and quickly transmitted to the 2nd layer. Because of more convergent connections from 1st to 2nd layers, some synchronous spikes are stably transferred and others are suppressed by the 2nd top-down signal. Thus, appropriate pairing of top-down signals to 1st and 2nd layers enables the circuits to execute an arbitrary stimulus-response behavior.

Keywords: neural network model, spike-timing dependent synaptic plasticity, stimulus-response, top-down signals, supplementary motor area.

1 Introduction

Stimulus-response circuits must be constantly updated in our brain. For example, imagine that you are ready to hit if the next ball is straight in the baseball game, or to hit a controller in response to some happenings in the video game. Especially, an animal must decide what to do next when it encounters a predator. In addition, the behavior will depend on the situation at that time. Thus the stimulus-response relationship should not be fixed. In other words, the stimulus-response circuit in the brain needs to have flexible and fast characteristics. How are the stimulus-response circuits with these characteristics constructed and activated in the neural networks? Neurons in the supplementary motor area are known to be activated more than 1 second earlier before exciting motor neurons evoke a behavior. This activation called 'readiness potential' starts at a few hundred milliseconds before we notice our intention to move [1]. During this readiness period, candidates of stimulus-response pair may be picked up in the brain. This hypothesis implies that stimulus-response circuits should be activated during this period.

On the other hand, from the informational viewpoint, let us consider the relationship between stimulus and responsive behavior. What kind of information

processing in the brain is needed so that we can behave as quickly as possible in response to the anticipated stimulus? First, the behavior plan should be computed previously [2]. Neuronal activities in the dorsal premotor cortex corresponding to the optional behaviors have been reported before the movement [3]. However, the information processing in the neural networks remains unknown. It should satisfy flexible and rapid behaviors in response to the current situation.

The purpose of this paper is to propose a simple and fundamental neural network model, in which stimulus-response circuits can be flexibly self-organized and the transmission from perception to execution of the response is very fast. Here, we summarize the hypothesis and conceptual meaning of our model. The details of the model are explained in the next section. Before the motor cortex executes an action in response to a specific stimulus, we assume that the pre-motor area has prepared for the stimulus-response behavior. This is the reason why our proposed model consists of two layers, whose roles are readiness for a stimulus-response combination and execution of actions, respectively. We think that top-down signals from the prefrontal cortices play an important role for both layers to decide which stimulus is focused and which responsive behavior will be activated. Thus the top-down signal in the model indicates which stimulus is currently focused and which behaviors are suppressed. If sensory and top-down signals are input to the interconnected neural network model with spike-timing dependent synaptic plasticity (STDP) [4], the network learns the transformation from the input spatiotemporal pattern into a synchronous spike [5]. The synchronous signal activates the neurons in the primary motor cortex, which leads to the movement. Exceptionally, unnecessary movements are prohibited by the other top-down to this area. Thus, required movements can be exclusively evoked in this model. Only the top-down or sensory signal provided to the interconnected network cannot evoke the response. Even when the top-down signal is provided, no response occurs unless the specific sensory pattern is input. An arbitrary stimulus-response pair can be selected if the top-down signals are appropriately selected. Additionally, since simultaneous spikes have strong and instant effects on the received neurons, the latency to response is very short, which implies that the transmission speed is faster than the case of mean firing rates. This paper proposes a basic numerical model and shows the validity through computer simulations. We expect that even if the details of model neurons are different, the results should be the same.

2 A Stimulus-Response Model Prepared by Top-Down Signals

This section explains the fundamental system of the proposed model. And one example to implement the system using mathematical equations is provided in the next section. Roughly, the model consists of 1st neuron group (1st layer) which receives sensory input signals and 2nd neuron group (2nd layer) which receives outputs of the 1st layer (Fig. 1). Both layers are assumed to be respectively controlled by top-down signals from higher cortical areas such as prefrontal cortices. The details of each layer are explained below.

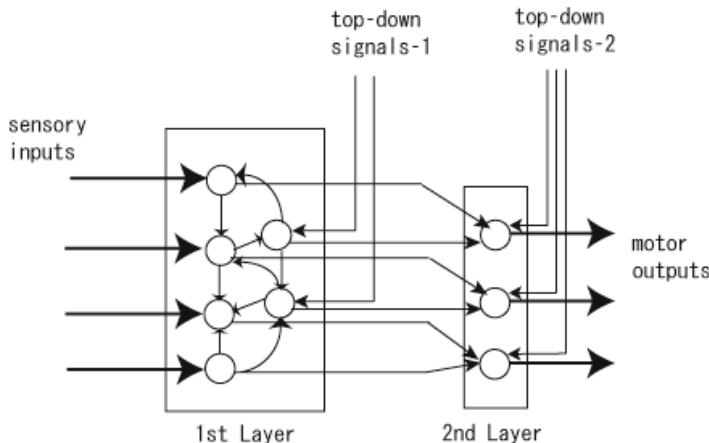


Fig. 1. Schematic structure of the stimulus-response model prepared by top-down signals

The 1st layer consists of interconnected model neurons, whose partial connections are randomly selected for simplicity. The neurons are leaky integrate-and-fire models. 80% are excitatory and the rest are inhibitory neurons in the 1st layer to avoid bursting by recurrent connections. 2nd layer neurons are excitatory only. The rate of synaptic connections is assumed to be 20% in the computer simulations because the biological brain has such sparse connections. The inputs to each neuron in this layer are sensory and top-down signals as well as the other neuron's outputs. Synaptic plasticity called STDP is assumed to be applied only to the connections between excitatory neurons in the 1st layer. The sensory and top-down inputs are called external inputs in this paper.

We assumed that some important information is represented by the spatiotemporal spike pattern. A spatiotemporal pattern during a constant period is repeatedly applied to the 1st layer as an external input. The spatiotemporal pattern is generated by independent Poisson trains with a common mean interspike interval. We assume that most part of the spatiotemporal pattern is from sensory inputs and the rest is from top-down signals. With enough learning times, it is known that a synchronous spike occurs only if the learned spatiotemporal pattern is input [5]. The synchronous signal can activate the primary motor cortex, which leads to the movement.

Assuming that the 1st layer learns the transformation from a spatiotemporal to a synchronous pattern, both of sensory and top-down signals are necessary to synchronous spikes. Concretely, when the sensory inputs are input without the top-down signal or the top-down signal is on without the sensory signal, synchronous spikes seldom occur. Therefore, when the top-down signal is received in the 1st layer, we can consider that this circuit waiting for the specific sensory input is in the ready state to move. This model premises that higher areas in the brain know the control code corresponding to the specific sensory information.

The synchronous spikes occur after the pattern is input to the circuit. Even one synchrony enables received neurons to fire immediately if convergent connections exist between sending and receiving neurons. Thus, the transmission speed is faster than the firing rate coding. In the neural multiple layers with divergent and convergent connections, the synchrony fast and stably propagates (called ‘synfire chain’) [6].

Since excitatory neurons in the 1st layer are connected to the 2nd layer neurons, the synchrony tends to activate the 2nd layer neurons. The connections from 1st layer to 2nd layer are divergent and convergent. And the connection probability is assumed to be higher as the difference of index numbers of neurons is smaller. We assume that top-down signals can previously suppress unnecessary movements. This top-down is different from that in the 1st layer. Thus, necessary movements are only evoked in response to the specific sensory inputs. Appropriate pairing of top-down signals to 1st and 2nd layers enables the circuits to execute arbitrary stimulus-response.

3 Computer Simulations

3.1 Model Neurons

The rate of synaptic connections is assumed to be 20% in the 1st layer neurons. The number of neurons is 200. Assuming that each neuron is a leaky integrate-and-fire model, the internal potential $V_i(t)$ of i neuron follows the differential equation as follows:

$$\tau \frac{dV_i(t)}{dt} = V_{rest} - V_i(t) + \sum_j w_{ij} f(V_j(t))(E_0 - V_i(t)) , \quad (1)$$

where $\tau = 20$ is a time constant, V_{rest} is a resting potential, w_{ij} is the synaptic efficacy from j to i neuron, $f(x)$ is an output function which corresponds to 1 if x is greater than the threshold θ and 0 otherwise, and E_0 is either of excitatory (E_{ex}) or inhibitory (E_{inh}) potential which depends on the j neuron is an excitatory (80%) or inhibitory (20%) neuron. When $V_i(t)$ reaches the threshold, $V_i(t)$ is reset to V_{rest} . $E_{ex} = 0$, $E_{inh} = -100$, $V_{rest} = -65.0$, $\theta = -60.0$.

3.2 Learning of Synchronous Spikes

Different independent Poisson trains with period $T = 100$ are periodically and frequently input to the 1st layer as the sensory and top-down signals. In Poisson trains, the interspike interval (ISI) follows an exponential distribution. Fig. 2 shows an example of the input pattern (mean ISI= 50), and all figures shown below are the results when this pattern is input to the 1st layer. We assume that the external inputs to neuron #0 to #19 are regarded as top-down signals and the inputs from #20 to #199 are sensory signals. STDP is applied to the first neuronal layer in our model. The relation between pre- and postsynaptic spike

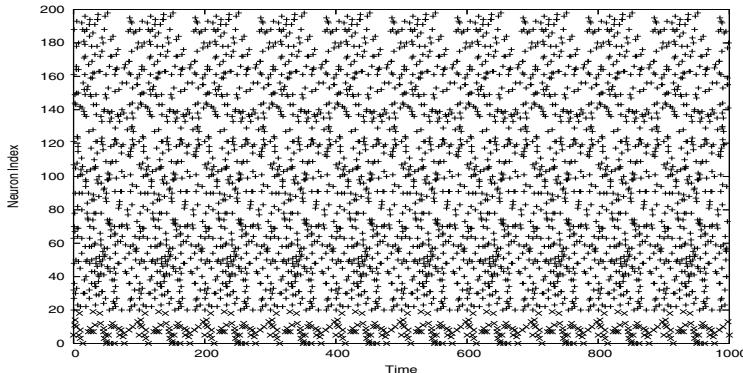


Fig. 2. An example of external input pattern to the 1st layer neurons

timings and change of synaptic weights follows an exponent function. Concretely, the algorithmic method of [7] is adopted here for the sake of simplicity.

Fig. 3 shows the output spikes of the 1st layer when the network is learning to make synchronous spikes in response to the external input pattern. The synchrony occurs periodically with period $T = 100$. This phenomenon has been examined and analyzed using computer simulations [5]. According to the literature, the abstract of generation mechanism of synchrony is as follows. The connections with cause and effect spike relations are strengthened and the others are weakened by STDP. This forces the flows of spikes to trigger an avalanche. Once a synchrony occurs, excited inhibitory neurons suppress other neurons for a while in addition to refractory effect. These effects spend some time before the next wave is elicited, then the synchrony easily occurs periodically.

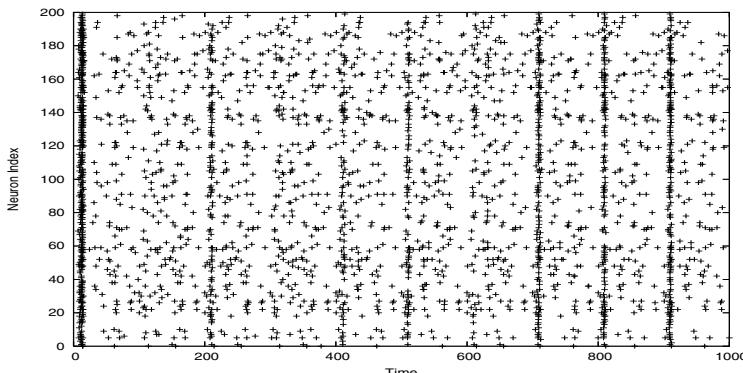


Fig. 3. Output of the 1st layer neurons. Synchronous spikes emerge around $t = 710, 810, 910$. The network learns to evoke synchronous spikes in response to the input pattern.

3.3 Prepared by the Top-Down Patterns

In the previous section, we show that the transformation from a spatiotemporal pattern to a synchrony can be learned by STDP. The proposed model assumes that a part of the spatiotemporal pattern comes from sensory inputs and the other come from top-down signals from higher functional areas. Briefly, the relation between sensory and top-down signals is assumed to be previously learned. Fig. 4 shows that learned synchronous spikes vanish even if the learned sensory pattern (#20-199) is input without the top-down signal (#0-19). Similarly, Fig. 5 shows the reverse case that different patterns (#20-199) are input as sensory signals with exactly the same top-down signal (#0-19). Thus, when the top-down signal is received in the 1st layer, we can consider that this circuit waiting for the specific sensory input is in the ready state to move. Then the synchrony in response to the sensory input is emitted toward the 2nd layer.

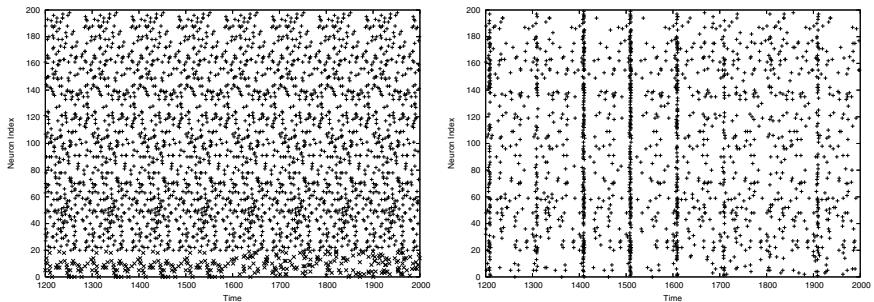


Fig. 4. Input patterns to the 1st layer (*left*) and output patterns (*right*) when sensory inputs are the same but top-down signals are different. From $t = 0$ to $t = 1600$, the learning (STDP) is applied. After $t = 1600$, only the top-down inputs are changed without STDP. The learned pattern is the same as Fig. 2

The connections from 1st layer to 2nd layer are divergent and convergent synaptic connections with higher connection probability for neighboring neurons. The number of neurons in 2nd layer is $1/3$ of the excitatory neurons in the 1st layer. Since the number of synapses for each 2nd layer neuron is constant, convergent connections for each neuron are three times more than those of divergent connections from a neuron. This structure contributes to stabilizing the transmission of synchrony. The model neuron of 2nd layer is almost the same as 1st layer except the interconnections within the layer. Fig. 6 shows the output patterns in 2nd layer when the pattern in Fig. 2 was input to the 1st layer. The spikes other than synchronous patterns are eliminated, compared with the spikes in Fig. 3. Thus synchronous spikes are emphasized through the convergent connections like a filter.

However, synchronous spikes tend to spread over the motor cortex at a price of speed. To prevent this, 2nd top-down signals are needed to previously suppress unnecessary movements. For example, if the arm is intended to move, suppressive

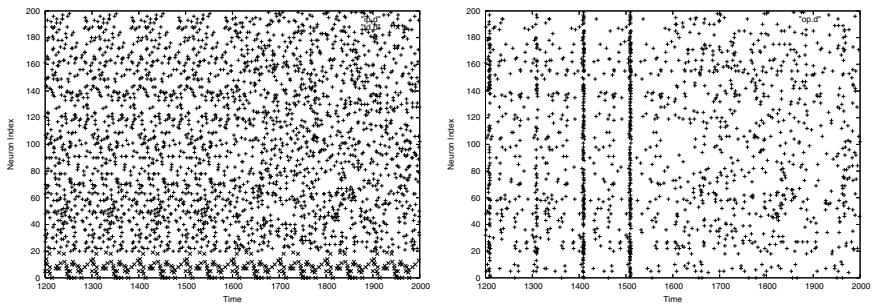


Fig. 5. Input patterns to the 1st layer (*left*) and output patterns (*right*) when top-down signals are the same but sensory inputs are different. From $t = 0$ to $t = 1600$, the learning (STDP) is applied. After $t = 1600$, only the sensory inputs are changed without STDP. The learned pattern is the same as Fig. 2.

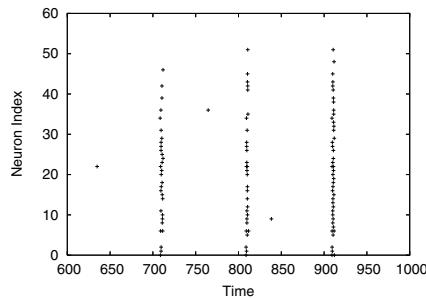


Fig. 6. Output of the 2nd layer neurons

signals are sent to the areas which move the other body parts such as legs and hands during movement planning. In this way, synchronous signals generated in 1st layer are sent to 2nd layer and unsuppressed motor areas are activated. At the end, synchronous firing will be transformed to high frequent firing rates in the primary motor cortex to put into action. As shown above, the sensory input is indicated by the 1st top-down signal and unnecessary motor areas are suppressed by the 2nd top-down signal. Then arbitrary stimulus-responsive behavior can be executed.

4 Conclusion

We propose a fundamental neural network model, which can trigger the movement in response to a specific sensory input using top-down signals supposed from highly functional brain areas. The feasibility is confirmed through computer simulations. Experimental data suggest that supplementary motor area (SMA) or pre-SMA much concerns the association between conditions and responses [8]. Since the damaged SMA cannot suppress the movements, SMA is thought to

contribute to the suppression of movements [9]. This feature corresponding to the 2nd top-down signal in our framework suggests that the function of our model is involved in SMA, pre-SMA, prefrontal area, and motor cortex. This model assumes that the stimulus-response pair is only one. Further refinements for more complicated conditions such as more than two pairs should be our future works. Additionally, the generation mechanism of top-down signals is also another important issue.

References

1. Libet, B., Gleason, C.A., Wright, E.W., Pearl, D.K.: Time of Conscious Intention to Act in Relation to Onset of Cerebral Activity (Readiness-Potential). The Unconscious Initiation of a Freely Voluntary Act. *Brain* 106, 623–642 (1983)
2. Tanji, J., Shima, K.: Role for Supplementary Motor Area Cells in Planning Several Movements Ahead. *Nature* 371, 413–416 (1994)
3. Cisek, P., Kalaska, J.F.: Neural Correlates of Reaching Decisions in Dorsal Premotor Cortex: Specification of Multiple Direction Choices and Final Selection of Action. *Neuron* 45, 801–814 (2005)
4. Markram, H., Lübke, J., Frotscher, M., Sakmann, B.: Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. *Science* 275, 213–215 (1996)
5. Hosaka, R., Araki, O., Ikeguchi, T.: STDP Provides the Substrate for Igniting Synfire Chains by Spatiotemporal Input Patterns. *Neural Computation* 20, 415–435 (2008)
6. Abeles, M.: *Corticonics – Neural Circuits of the Cerebral Cortex*. Cambridge University Press, Cambridge (1991)
7. Song, S., Miller, K.D., Abbott, L.F.: Competitive Hebbian Learning through Spike-timing-dependent Synaptic Plasticity. *Nature Neuroscience* 3, 919–926 (2000)
8. Nachev, P., Kennard, C., Husain, M.: Functional Role of the Supplementary and Pre-supplementary Motor Areas. *Nature Reviews Neuroscience* 9, 856–869 (2008)
9. Haggard, P.: Human Volition: towards a Neuroscience of Will. *Nature Reviews Neuroscience* 9, 934–946 (2008)

A Novel Shape-Based Image Classification Method by Featuring Radius Histogram of Dilating Discs Filled into Regular and Irregular Shapes

Xiaoyu Zhao, Chi Xu, Zheru Chi, and Dagan Feng

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

Abstract. In this paper, a novel feature for shape-based image classification is proposed, in which a set of randomly dilating discs are distributed on an unknown regular or irregular shape, and the radius histogram of discs is used to represent the target shape. By such doing, a shape can be modeled by the radius histogram. The proposed feature is rather effective for shape retrieval with rotation and scaling invariance. The proposed feature is particularly effective in the retrieval of string-linked objects in which conventional approaches may fail badly. Experimental results on seven shapes and string-linked objects show that our proposed new feature is very effective in shape classification and shape-based image retrieval.

Keywords: shape feature, radius histogram, shape classification, shape-based image retrieval, dilating discs.

1 Introduction

Object recognition have found many important applications such as surveillance, image and video retrieval, and robotics. For decades automated quantitative characterization of object shapes has been an very important pursue in developing better object recognition systems [1] [2].

In the area of regular or irregular shape recognition, the most popular approaches include

- Fourier shape descriptor which applies the Fourier transformation of object contours to describe object shapes and
- moment invariants which is derived from the theory of algebraic invariant and used to extract image features with rotation, scale, and translation (RST) invariance.

The other features explored include intra-angle histogram, finite element methods, turning function, and wavelet descriptor, etc [9] [3] [4]. Many new features are continuously reported. Although these object shape features are quite effective to recognize a limited number of regular or irregular shapes but they may

perform quite badly in cases of shape deformation or distortion. In addition, conventional features cannot recognize the objects which are lightly touched together or they are string-linked (see Fig. 5(b)).

Inspired by ideas of fractal that irregular objects can be approximated by a set of rough or fragmented geometric shapes, and the original root can be traced back by continuous functions [6] [7] [8]. In this paper, a novel feature is proposed to recognize the shapes with deformation and distortion, and string-linked. In our method, the histogram of radiiuses of dilating discs deposited onto a target shape is generated and used to characterize the shape. With its simplicity in formulation, the feature shows its effectiveness in characterize various shapes.

In experiments, seven artificially generated shapes as well as several string-linked objects are tested with satisfactory results. The feature has also been tested on the recognition of several typical plant leaf types. Our results are also compared with method of moment invariant, which is commonly used in shape recognition [5] [10] [11]. This paper is organized as follows. In the next section, the definition of the radius histogram is given and its properties are discussed. Experimental results are reported in Section III. Finally, concluding remarks are drawn in Section IV.

2 Methodology

2.1 Shape Feature Extraction

In this shape-based classification approach, a model of the radius histogram is established, which composes a radius histogram distribution that represents a shape.

Radius Histogram Distribution Construction. Initially, two thousand points ($i = 2000$) are randomly deposited into a shape as centers of dilating discs, and the discs expand until they reach the border of the shape. Assuming the radius r_i of the discs are normalized into range of $[0, 1]$ for further computation.

The normalization of radius (r_i , where $i = 1, \dots, 2000$) can form a radius space based on the following formula:

$$r'_i = \frac{r_i - r_{min}}{r_{max} - r_{min}} \quad (1)$$

where r_{min} and r_{max} are shortest and longest radius of the discs among those have been reached the borders of the shape. The process of normalization is important to guarantee the radius histogram is not sensitive to re-scaling of the shapes.

Since an interested shape is covered by two thousand dilating discs. The filling dilating discs could be possibly overlapped with each other. Given a shape of interest S , for a dilating disc originating from a random point in the shape S to its boundary, the dilating disc can be described as a vector $\mathbf{d}_i = \{(x, y), r_i\}$. The radius space can then be clearly quantified into k -bin histogram, and the bin k_i is incremented as the occurrence probability of a particular radius increases.

Thus, any shape can be represented by an index set that is in form of binary number. The radius set could be treated as a feature array of a shape, which is defined as shape-array. With different radius, the radius histogram distribution formed should not be sensitive if the target shape in spite of rotations by any angles. When the interval of radii is small enough, an approximate continuous curve can be drawn for the peak values of the histogram. By applying this approach, any shapes can be transferred into different integrals of probability density function (PDF) of radius, and can be described by different curves in the same ordinate system.

Implementation of the Algorithm. To the cases of two obviously distinguishable shapes, round and rectangle as shown in Figure 1(a), the curve of PDF is generated as shown in Figure 1(b).

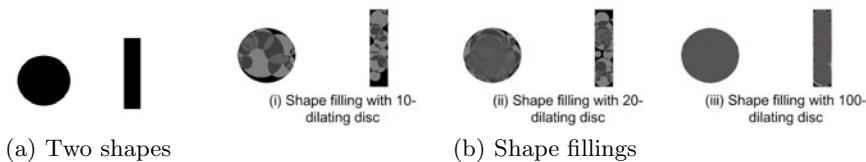


Fig. 1. Dilating disc filling

The procedure of shape-filling is illustrated in Figure 1(b). Generally speaking, the more dilating discs deposited into the shape, the more closely the density function can represent the shape (shown in Figure 1(b) from (i) to (iii)).

2.2 Classification of Shapes

To test the robustness of the proposed method when it recognizes the shapes with rotation and scaling, the curve of PDF is used to represent a shape. Every shape can generate a curve of PDF which shows the frequency of the appearance of each bin from different length of radius, and the same kind of shape with rotation and re-scaling should generate similar curves (j curves in the i -th kind of shape) of PDF, since the probability of a radius falling within a given bin is given by the integral of its density over the bin. By calculating the mean of the curves from the same kind, a general model of the particular shape can be determined, and plotted clearly in the 2-dimension ordinate system:

$$u_i = \frac{1}{k} \sum_{j=1}^k p_{ij} \quad (2)$$

from which, u_i is the model curve representing a particular shape, k is the number of bins, and p_{ij} represents PDF curves from several similar training samples. A new radius distribution of any shapes should be evaluated with the models. The linear least square method is used for such evaluation. Let $f_t(x, y)$ be the curve

of a PDF obtained from dilating disc radius of an unknown target shape, the distance, d_{ti} , between $f_t(x, y)$ and one of the existing model curve $f_i(x, y)$ is calculated by:

$$d_{ti} = \sqrt{\sum_{j=1}^k (f_{tj}(x, y) - f_{ij}(x, y))^2} \quad (3)$$

Then, such distances are found out between the target curve and all candidate models, and the classification is determined by choosing the minimal distance:

$$\text{Classification} = \min_{t \in \text{sample nbr.}} (d_t) \quad (4)$$

KNN (K-nearest neighbor) and BP (Back-Propagation neural network) models are used to evaluate the performance of the proposed method. SVM (support vector machine) test is applied to estimate the regression.

3 Experiment Design and Result Analysis

The evaluation simulation is conducted in three levels. The first level is to recognize seven different regular and irregular artificial shapes, the second level is to classify similar dumbbell- and lace-like stringed shapes with different class components and the third level is to identify five kinds of dichromatic images of real leaves. Each experiment is carried out through learning and evaluation procedures. In the learning procedure, sample shape should be deposited by 2000 dilating discs 100 times at different rotated positions, and a model for a particular shape is abstracted. During the evaluation procedure, only the PDF of the testing samples are calculated and compared with all existing models. The minimal distance of PDF between the testing sample from that of a particular model indicates that the testing sample falls into the class that the model represents.

3.1 Recognition of Regular and Irregular Shapes

Figure 2 shows the seven regular and irregular shapes to be recognized, including triangle, round, flash, 4-angle star, new moon, 5-angle star, and rectangle.

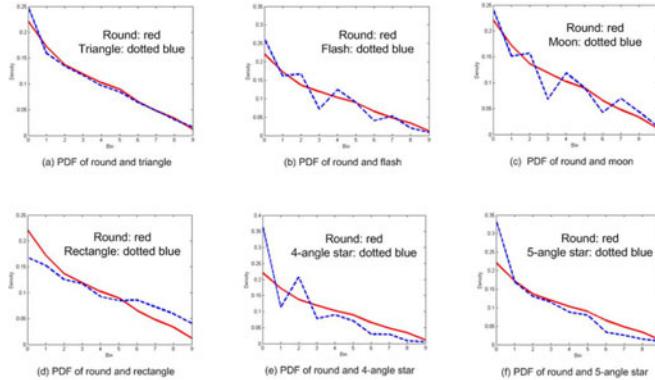
By applying the mentioned procedures in II, the PDFs or the models are calculated. The obtained PDFs of seven shapes are shown in Figure 3, and the classification results are calculated and recorded in Table 1. The test results

No.	Name	Shape	No.	Name	Shape
0	Triangle		4	Moon	
1	Round		5	5-angle	
2	Flash		6	Rectangle	
3	4-angle				

Fig. 2. Seven different regular and irregular shapes

Table 1. Classification of seven different shapes by proposed dilating disc method

Samples	Method	Accuracy
105	KNN	100%
350	KNN	99.43%
700	KNN	99.85%
3112	KNN	99.94%

**Fig. 3.** PDF comparison of seven shapes

suggest that the method is quite effective and the classification accuracy is very high.

Comparing with the results calculated by moment invariants in Table 2, the performance of the proposed dilating disc method is at least equivalent to those of the conventional ones.

3.2 Recognition of Dumbbell- and Lace-Like String-Linked Shapes

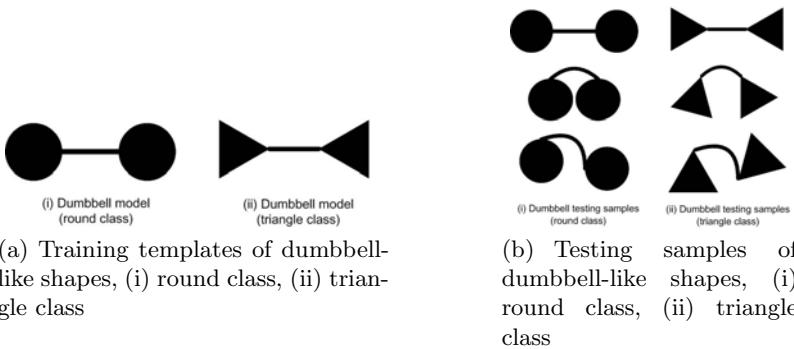
The obvious difference, and maybe an outstanding advantage, of the proposed approach over the conventional ones is that the shape information is gradually obtained from endogenous elements instead of exogenous features of a shape itself, although the information of shape borders is indispensable. For testing the ability that the proposed approach is powerful dealing with some shapes those conventional methods are difficult to achieve, the dumbbell- and lace-like string liked shapes are chose to be conducted the experiment.

Firstly, as shown in Figure 4(a), two dumbbell shapes are trained to be models, in which (i) is for shapes belong to round class, and (ii) is for shapes for triangle class.

The testing samples are shown in Figure 4(b), while more samples with different rotations are not listed, and the testing samples are 100% correctly falling into the class they belong.

Table 2. Classification of seven different shapes by conventional approaches

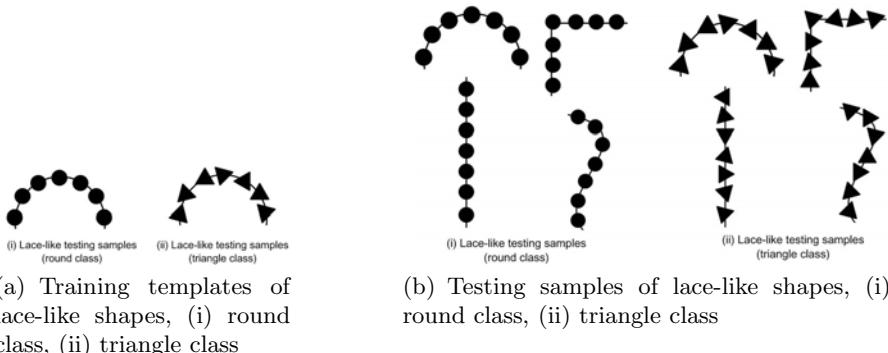
Samples	Method	Accuracy
105	Normalized KNN	85.7%
	Moment invariants	90%
	BP supervised moment invariants	92%
350	Normalized KNN	85.7%
	Moment invariants	90%
	BP supervised moment invariants	92%
700	Normalized KNN	85.7%
	Moment invariants	90%
	BP supervised moment invariants	92%
3112	Normalized KNN	85.7%
	Moment invariants	90%
	BP supervised moment invariants	92%

**Fig. 4.** Dumbbell-like shape classification

Secondly, as shown in Figure 5(a), two lace-like stringed shapes are trained to be models, in which (i) is for shapes belong to round class, and (ii) is for shapes for triangle class.

The testing samples are shown in Figure 5(b), while more samples with different rotations are not listed, and, excitingly, the testing samples are again 100% correctly falling into the class they belong.

For evaluating the robustness and efficiency of the proposed method, the classification results are compared with those by the moment invariant. The moment invariant approach is one of the most powerful image classification techniques, and is trained to recognize the same image samples. The supervised BP neural networks are applied to optimize the moment invariants by adjusting different weights of seven moments. The experiment target is to classify the round and triangle shapes which are connected into dumbbell- and lace-like positions. The results of classification by moment invariant are about 50%, which are listed in Table 3. This fact means when the shapes are stringed together and deposed differently, the moment invariant approach failed to work properly by choosing

**Fig. 5.** Lace-like shape classification**Table 3.** Performance of classification of round and triangle shapes

Shape	Dilating disc		Moments invariants w/BP	
	Class round	Class triangle	Class round	Class triangle
Dumbbell-like	100%	100%	51%	49%
Lace-like	100%	100%	50%	50%

the class for the shapes arbitrarily. The reason could probably be that the important information for the moment invariant to recognize a shape is the border of the shape, and when small shapes are connected together, the information from borders of the assembled shape just misleads the moment invariant with neglecting the detail information of the real shapes.

The results clearly indicates that in recognition of dumbbell- and lace-like string linked shapes, the proposed method works perfectly for classification of round and triangle although the shapes are stringed together into different positions, while the conventional moment invariants approach just randomly picks up the objects from the candidate samples if the border of the entire string is differently deposed.

4 Conclusion

A novel shape-based image classification method is proposed. The proposed method constructs classification features based on the radius histogram of dilating discs filling in an object. The feature in the proposed method is basically rotation and scaling invariant.

Moreover, the method can deal very well with regular and irregular artificial shapes, even if they are string-linked to each other. Experimental results suggest that the proposed method can at least achieve similar results as the best conventional methods, such as moment invariant, and provide a much better performance when classifying string-linked artificial objects. The most outstanding

advantage of this proposed method is that it attempts to utilize the endogenous property of any shapes for characterizing the shapes. The future work will include extending the proposed method for gesture recognition.

References

1. Flusser, J., Suk, T., Zitová, B.: Moments and moment invariants in pattern recognition. J. Wiley, Chichester (2009)
2. Wang, Z., Chi, Z., Feng, D.: Shape based leaf image retrieval, IEE Proceedings - Vision. Image and Signal Processing 150(1), 34–43 (2003)
3. Fu, H., Chi, Z., Feng, D.: Attention-Driven Image Interpretation with Application to Image Retrieval. Pattern Recognition 39(9), 1604–1621 (2006)
4. Zou, W., Chi, Z., Lo, K.C.: Improvement of Image Classification Using Wavelet Coefficients with Structured-Based Neural Network. International Journal of Neural Systems 18(3), 195–205 (2008)
5. Mukundan, R., Ramakrishnan, K.R.: Moment functions in image analysis: theory and applications. World Scientific, Singapore (1998)
6. Pesin, Y., Climenhaga, V.: Lectures on fractal geometry and dynamical systems, Mathematics Advanced Study Semesters. American Mathematical Society, Providence University (2009)
7. Edgar, G.A.: Measure, topology, and fractal geometry. Springer Science+Business Media, LLC, New York (2008)
8. Lapidus, M.L.: Fractal geometry, complex dimensions and zeta functions: geometry and spectra of fractal strings. Springer, Dordrecht (2006)
9. Micheli-Tzanakou, E.: Supervised and unsupervised pattern recognition: feature extraction and computational intelligence. CRC Press, Boca Raton (2000)
10. Rodrigues, M.A. (ed.): Invariants for pattern recognition and classification. World Scientific, Singapore (2000)
11. Wang, Y., Makedon, F.: R-Histogram: Quantitative Representation of Spatial Relations for Similarity-based Image Retrieval. In: Proceedings of the 11th Annual ACM International Conference on Multimedia, Berkeley, California, USA, pp. 323–326 (2003)

Learning Visual Object Categories and Their Composition Based on a Probabilistic Latent Variable Model

Masayasu Atsumi

Dept. of Information Systems Science, Faculty of Engineering, Soka University
1-236 Tangi, Hachioji, Tokyo 192-8577, Japan
matsumi@t.soka.ac.jp

Abstract. This paper addresses the problem of statistically learning typical features which characterize object categories and particular features which characterize individual objects in the categories. For this purpose, we propose a probabilistic learning method of object categories and their composition based on a bag of feature representation of co-occurring segments of objects and their context. In this method, multi-class classifiers are learned based on intra-categorical probabilistic latent component analysis with variable number of classes and inter-categorical typicality analysis. Through experiments by using images of plural categories in an image database, it is shown that the method learns probabilistic structures which characterize not only object categories but also object composition of categories, especially typical and non-typical objects and context in each category.

1 Introduction

A visual object has two-fold features which classify it to a category and identify it from others. The former is a common and typical feature among objects of a category and the latter is a differential feature peculiar to an object. By the way, though people can classify and identify objects by drawing on some features, these features have not been learned in a supervised manner but learned in a unsupervised manner. Since it is known that context facilitates recognition of ambiguous objects in a crowded scene , it is assumed that categories and objects are learned in association with a typical context with other related objects. In this paper, we address the problem of statistically learning typical features which characterize object categories and particular features which characterize individual objects, that is, object composition of categories. Concretely, given a collection of categorized image sets each of which contains co-occurring segments of the categorical objects and their context, it is required to learn probabilistic structures of features which characterize categories and object composition of categories.

For this purpose, this paper proposes a probabilistic learning method of object categories and their composition based on a local feature representation of

segments. In order to represent local feature of segments, the SIFT features [2] of keypoints in co-occurring segments are clustered by the K-tree method [3] to generate a set of key features (a code book) and a bag of features (BoF) [4] of each segment is calculated by using this code book. In probabilistic learning of object categories and their composition, multi-class classifiers are learned based on intra-categorical probabilistic latent component analysis (PLCA) [5,6] with variable number of classes and inter-categorical typicality analysis. The intra-categorical PLCA of each category learns a probabilistic structure of categorical objects and their context from BoF histograms of segments in the category. The inter-categorical typicality analysis judges how typical each class is in each category and synthesizes a typical probabilistic structure for each category. We call this learning method a probabilistic latent component forest (PLCF).

As for related works, probabilistic latent semantic analysis has been applied to learn object categories [7]. It has also learned approximate spatial layout of object categories as context. There have been several learning methods [8,9,10] which incorporate context into object categorization. The main difference of our method from those existing ones is that it learns not only probabilistic structures of object categories but also object composition of categories, especially typical and non-typical objects and context in each category.

This paper is organized as follows. Section 2 presents the proposed probabilistic learning method. Experimental results are shown in section 3 and we conclude our work in section 4.

2 The Proposed Method

2.1 Object Representation by Using a BoF

An object segment is represented by a BoF histogram [4] of its local feature. In order to calculate the BoF histogram, first of all, any keypoints in a segment are extracted and a 128-dimensional SIFT feature [2] is calculated for each keypoint. Then, all the SIFT features of all segments are clustered by the K-tree method [3] to obtain a set of key features as a code book. Finally, a BoF histogram of each segment is calculated by using this code book. As a result, feature of co-occurring segments is represented by BoF histograms.

In this paper, let s_{c,i_j} be a segment j extracted from an image i of a category c , S_c be a set of segments extracted from any images of a category c , and N_{c_s} be the number of segments in S_c . Let f_n be a n -th element of key features F , N_f be the number of key features and $\{H(s_{c,i_j}, f_n)\}_{f_n \in F}$ be a BoF histogram of a segment s_{c,i_j} . Let $q_{c,r}$ be a latent class of a category c , Q_c be a set of latent classes of a category c , N_{c_q} be the number of classes of a category c , C be a set of categories and N_c be the number of categories.

2.2 Learning Intra-categorical Multi-class Classifier of Objects

The problem of learning a multi-class classifier for segments in a category c is estimating probabilities $p(s_{c,i_j}, f_n) = \sum_r p(q_{c,r})p(s_{c,i_j}|q_{c,r})p(f_n|q_{c,r})$ namely

$\{p(q_{c,r})|q_{c,r} \in Q_c\}$ C $\{p(s_{c,i_j}|q_{c,r})|s_{c,i_j} \in S_c, q_{c,r} \in Q_c\}$, $\{p(f_n|q_{c,r})|f_n \in F, q_{c,r} \in Q_c\}$ and the number of latent classes N_{c_q} that maximize the following log-likelihood

$$L_c = \sum_{i_j} \sum_n H(s_{c,i_j}, f_n) \log p(s_{c,i_j}, f_n). \quad (1)$$

When the number of latent classes is given, these probabilities are estimated by the EM algorithm in which the following E-step and M-Step are iterated until convergence

[E-step]

$$p(q_{c,r}|s_{c,i_j}, f_n) = \frac{[p(q_{c,r})p(s_{c,i_j}|q_{c,r})p(f_n|q_{c,r})]^\beta}{\sum_{q_{c,r'}} [p(q_{c,r'})p(s_{c,i_j}|q_{c,r'})p(f_n|q_{c,r'})]^\beta} \quad (2)$$

[M-step]

$$p(f_n|q_{c,r}) = \frac{\sum_{s_{c,i_j}} H(s_{c,i_j}, f_n) p(q_{c,r}|s_{c,i_j}, f_n)}{\sum_{f_{n'}} \sum_{s_{c,i_j}} H(s_{c,i_j}, f_{n'}) p(q_{c,r}|s_{c,i_j}, f_{n'})} \quad (3)$$

$$p(s_{c,i_j}|q_{c,r}) = \frac{\sum_{f_n} H(s_{c,i_j}, f_n) p(q_{c,r}|s_{c,i_j}, f_n)}{\sum_{s_{c,i_j'}} \sum_{f_n} H(s_{c,i_j'}, f_n) p(q_{c,r}|s_{c,i_j'}, f_n)} \quad (4)$$

$$p(q_{c,r}) = \frac{\sum_{s_{c,i_j}} \sum_{f_n} H(s_{c,i_j}, f_n) p(q_{c,r}|s_{c,i_j}, f_n)}{\sum_{s_{c,i_j}} \sum_{f_n} H(s_{c,i_j}, f_n)} \quad (5)$$

where β is a temperature coefficient.

The number of latent classes is determined through an EM iterative process with subsequent class division. The process starts with one or a few classes, pauses at every certain number of EM iterations less than an upper limit and calculates the following index, which is called the degree of scatter,

$$\delta_{q_{c,r}} = \frac{\sum_{s_{c,i_j}} (\sum_{f_n} |p(f_n|q_{c,r}) - D(s_{c,i_j}, f_n)|) \times p(s_{c,i_j}|q_{c,r})}{N_f \times N_{c_s}} \quad (6)$$

where

$$D(s_{c,i_j}, f_n) = \frac{H(s_{c,i_j}, f_n)}{\sum_{f_{n'}} H(s_{c,i_j}, f_{n'})} \quad (7)$$

for $\forall q_{c,r} \in Q_c$. Then a class whose degree of scatter takes a maximum value among all classes is divided into two classes. This iterative process is continued until $\delta_{q_{c,r}}$ -values for all classes become less than a certain threshold. The latent class is divided into two classes as follows. Let q_{c,r_0} be a source class to be divided and let q_{c,r_1} and q_{c,r_2} be target classes after division. Then, for a segment $s_{c,i_j^*} = \arg \max_{i_j} \{p(s_{c,i_j}|q_{c,r_0})\}$ which has the maximum conditional probability and its BoF histogram $H(s_{c,i_j^*}, f_n) = [h_{c,i_j^*}(1), \dots, h_{c,i_j^*}(N_f)]$, one class q_{c,r_1} is set

by specifying its conditional probability of key features, conditional probabilities of segments and a class probability as

$$p(f_n|q_{c,r_1}) = \frac{h_{c,i_j^*}(n) + \alpha}{\sum_{n'}(h_{c,i_j^*}(n') + \alpha)} \quad \forall f_n \in F \quad (8)$$

$$p(s_{c,i_j}|q_{c,r_1}) = p(s_{c,i_j}|q_{c,r_0}) \quad \forall i_j \in S_c \quad (9)$$

$$p(q_{c,r_1}) = \frac{p(q_{c,r_0})}{2} \quad (10)$$

respectively where α is a positive correction coefficient. Another class q_{c,r_2} is set by specifying its conditional probability of key features $\{p(f_n|q_{c,r_2})|f_n \in F\}$ at random, conditional probabilities of segments $\{p(s_{c,i_j}|q_{c,r_2})|i_j \in S_c\}$ as 0 for s_{c,i_j^*} and equal for other segments, and a class probability as $p(q_{c,r_2}) = \frac{p(q_{c,r_0})}{2}$. As a result of subsequent class division, latent classes form a binary tree, which we call a probabilistic latent component tree (PLCT).

The temperature coefficient β is set 1.0 until the number of classes is fixed and after that it is gradually decreased according to a given schedule of the tempered EM until convergence.

2.3 Inter-categorical Typicality Analysis

Inter-categorical typicality analysis evaluates each latent class of each category whether or not it is typical in the category and calculates a conditional probability of key features for the category by synthesizing those probabilities of its typical classes. The typicality of a class is judged based on whether it appears in its category with high frequency but does not appear in other categories only with low frequency. Typical classes consist of classes of object segments and co-occurring contextual segments which distinguish a category and a synthesized conditional probability of key features encodes characteristics of the category. Here, in general, co-occurring contextual segments are objects of other categories or background. By the way, exceptional object segments, as well as typical object segments, are also encoded by conditional probabilities of key features of some non-typical classes.

For the inter-categorical typicality analysis, let the distance between classes $q_{c_1,r_1} \in Q_{c_1}$ and $q_{c_2,r_2} \in Q_{c_2}$ of any different categories c_1 and c_2 be

$$L_1(q_{c_1,r_1}, q_{c_2,r_2}) = \frac{\sum_{f_n} |p(f_n|q_{c_1,r_1}) - p(f_n|q_{c_2,r_2})|}{2} \quad (11)$$

by using their conditional probabilities of key features. Then the distance of a class $q_{c,r} \in Q_c$ of a category c from classes of other categories is defined as

$$d(c, q_{c,r}) = \frac{\sum_{c' \in C - \{c\}} \sum_{q_{c',r'} \in Q_{c'}} (L_1(q_{c,r}, q_{c',r'}) \times p(q_{c',r'}))}{N_c - 1}. \quad (12)$$

Now, for the mean distance $\bar{d}(c) = \frac{\sum_{q_{c,r} \in Q_c} d(c, q_{c,r})}{N_{c_q}}$ of all classes of a category c , the deviation of a distance $d(c, q_{c,r})$ is given by $\Delta(c, q_{c,r}) = d(c, q_{c,r}) - \bar{d}(c)$. Then the typicality index of a class $q_{c,r}$ of a category c is defined as

$$\gamma(q_{c,r}) = \frac{1}{1 + \exp\left(-\mu\left(p(q_{c,r}) - \left(\frac{1}{N_{c_q}} - \Delta(c, q_{c,r})\right)\right)\right)} \quad (13)$$

where μ is a gain coefficient. This index is called the degree of categorical class.

The conditional probability of key features for a category c is defined for a set of typical classes $Q_c^* = \{q_{c,r} | \gamma(q_{c,r}) \geq \theta, q_{c,r} \in Q_c\}$ as

$$p(f_n | Q_c^*) = \sum_{q_{c,r} \in Q_c^*} (\lambda(q_{c,r}) \times p(f_n | q_{c,r})) \quad (14)$$

$$\lambda(q_{c,r}) = \frac{p(q_{c,r})}{\sum_{q_{c,r'} \in Q_c^*} p(q_{c,r'})} \quad (15)$$

where θ is a threshold which determines whether or not a class is typical.

3 Experimental Results

A collection of 14 categorized image sets each of which contained about 15 images and the total number of images was 209 was prepared from the MSRC labeled image database v2¹. Fig. 1 shows some categorical images and co-occurring segments with labels for the images. As shown in Fig. 1, co-occurring segments were extracted according to given 23 labels and BoFs of extracted segments were calculated by using a code book which was generated from all SIFT features of those segments.

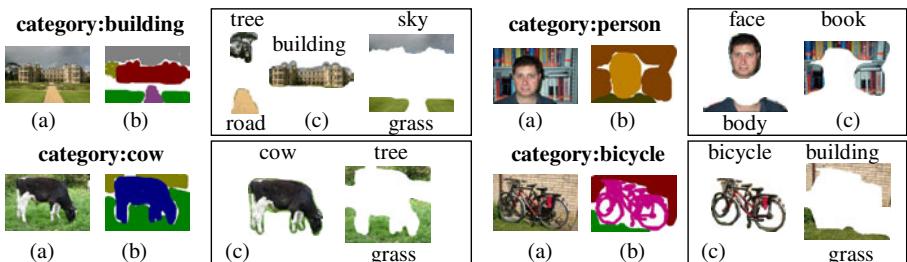


Fig. 1. Examples of (a)categorized images, (b)labels and (c)co-occurring segments. Images of 14 categories (“tree”, “building”, “airplane”, “cow”, “person”, “car”, “bicycle”, “sheep”, “sign”, “bird”, “chair”, “cat”, “dog”, “boat”) were used in experiments.

¹ <http://research.microsoft.com/vision/cambridge/recognition/>

Main learning parameters are set as follows. As for learning of multi-class classifiers, a threshold of class division is 0.046 and a correction coefficient α in the expression (8) is 2.0. In the tempered EM, a temperature coefficient β was decreased by multiplying it by 0.95 at every 20 iterations until it became 0.8. As for inter-categorical typicality analysis, a gain coefficient μ in the expression (13) is 2.5 and a threshold θ for determining typical classes is 0.5.

The number of SIFT features which were extracted from all segments was 58633. The code book size of key features which were obtained by the K-tree method was 412. The BoF histograms were calculated for 366 segments whose number of SIFT features were more than 30.

3.1 Results on Characterizing Object Composition

Fig. 2 shows PLCTs of multi-class classifiers for some categories. In Fig. 2, a typical segment of a class r of each category c is a segment s_{c,i_j} that maximizes $p(q_{c,r}|s_{c,i_j})$. Also, a typical co-occurring segment of each category c is a co-occurring segment $s_c = \{s_{c,i_k} | k \in K\}$ that maximizes the following typicality index $R(s_c) \equiv \sum_{k \in K} \max_{q_{c,r} \in Q_c^*} p(q_{c,r}|s_{c,i_k})$, where Q_c^* is a set of typical classes of the category c . The mean number of classes and typical classes per PLCT for 14 categories are 7.57 and 2.93 respectively. As shown in Fig. 2, typical classes mainly distinguish segments of categorical objects but classes of frequent co-occurring contextual segments may also become typical ones. On the other hand, non-typical classes represent contextual segments and also rare categorical objects.

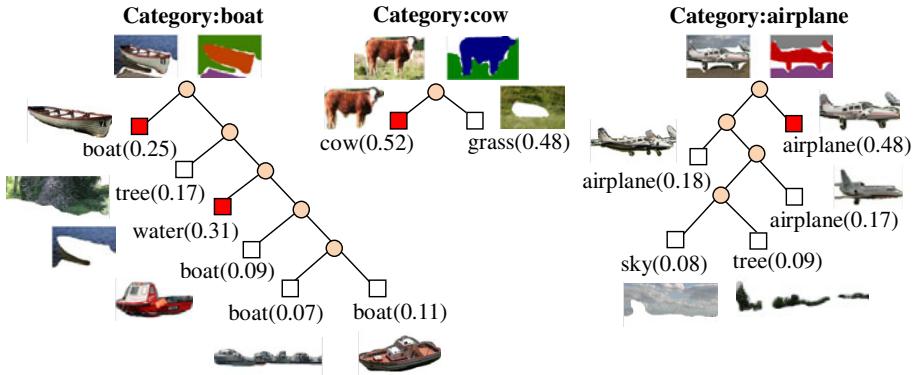


Fig. 2. PLCTs of multi-class classifiers for some categories. A colored square represents a typical class, a white square represents a non-typical class and a value in a parenthesis represents a class probability. A typical co-occurring segment of each category is depicted above a PLCT and a typical segment of each class is depicted beside the class.

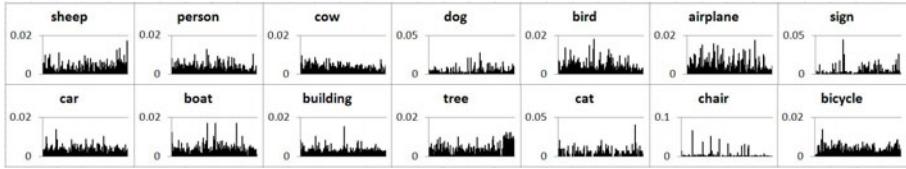


Fig. 3. Conditional probabilities of key features for all categories

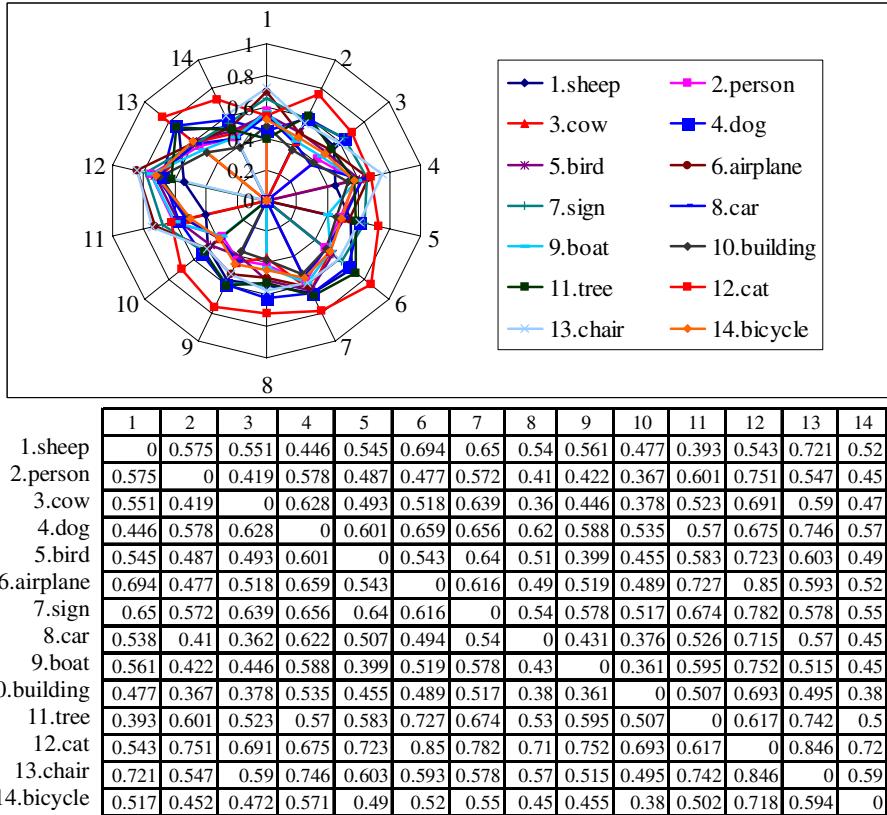


Fig. 4. Distance between conditional probabilities of categorical key features

3.2 Results on Characterizing Object Categories

Fig. 3 shows conditional probabilities of key features for all categories and Fig. 4 shows distance between each pair of them which is defined by the expression (II). The mean distance of all pairs of categories is 0.56. Since a conditional probability of key features for a category is a weighted sum of conditional probabilities of key features for its typical classes with their class probabilities, it mainly reflects features of typical categorical objects. It is observed that each category

has a different distribution of conditional probability of key features and is distinguishable from others by those conditional probabilities of key features since the distance between them is large. This distance is dependent on a threshold θ which determines typical classes in inter-categorical typicality analysis. When the threshold is made smaller, more classes become typical classes, which in general decreases the distance, that is, distinguishability among categories.

4 Conclusions

We have proposed a probabilistic learning method of object categories and their composition based on intra-categorical probabilistic latent component analysis with variable number of classes and inter-categorical typicality analysis. Through experiments by using images of plural categories in the MSRC labeled image database, it was shown that the method learned probabilistic structures which characterized not only object categories but also object composition of categories, especially typical and non-typical objects and context in each category.

The main ongoing works are to incorporate a preprocessor of image segmentation into this learning method and to build an integrated category and object recognizer which makes full use of this probabilistic structure.

References

1. Bar, M.: Visual objects in context. *Nature Reviews Neuroscience* 5, 617–629 (2004)
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
3. Shlomo, G.: K-tree; a height balanced tree structured vector quantizer. In: Proc. of the 2000 IEEE Signal Processing Society Workshop. vol. 1, pp. 271–280 (2000)
4. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Proc. of ECCV Workshop on Statistical Learning in Computer Vision, pp. 1–22 (2004)
5. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning* 42, 177–196 (2001)
6. Shashanka, M., Raj, B., Smaragdis, P.: Probabilistic latent variable models as non-negative factorizations. *Computational Intelligence and Neuroscience* 2008 (2008)
7. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: Proc. of IEEE ICCV, pp. 370–377 (2005)
8. Torralba, A.: Contextual priming for object detection. *International Journal of Computer Vision* 53, 169–191 (2003)
9. Rabinovich, A., Vedaldi, C., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: Proc. of IEEE ICCV (2007)
10. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: Proc. of IEEE CS Conf. on CVPR (2008)

Evidence for False Memory before Deletion in Visual Short-Term Memory

Eiichi Hoshino^{1,2} and Ken Mogi²

¹ Tokyo Institute of Technology, Department of Computational Intelligence and Systems Science, 4259 Nagatsutacho Midori-ku Yokohama-shi Japan 226-8502

² Sony Computer Science Laboratories, Fundamental Research Laboratory, Takanawa Muse Bldg. 3-14-13, Higashigotanda, Shinagawa-ku, Tokyo, Japan 141-0022
eiichi@brn.dis.titech.ac.jp

Abstract. Forgetfulness results in interference and/or deletion. Visual short-term memory (VSTM) gradually decays as the retention time elapses, causing forgetfulness. Little is known about forgetfulness in VSTM, while substantial studies on VSTM have focused on the process of memory encoding, often with control of attention. Evidences suggest that the prefrontal cortex may contribute to maintain short-term memory during extended retention periods while the posterior parietal cortex may support the capacity-limited store of visual items. Here we conduct a visual memory experiment to measure the levels and source of memory decay. In particular, multiple retention intervals were used between the presentation of a study array and a cue. The results show that the correct response to cued objects decreased as retention interval increased while that to uncued and novel objects remain unchanged. These data indicate that forgetfulness in VSTM is primarily due to interference rather than memory deletion.

Keywords: Visual short-term memory, forgetfulness, retro-cue, interference.

1 Introduction

In delayed matching tasks, such as change blindness, there is more failure in recognition as the retention time increases. Memory of an item to which subjects is exposed to tends to be maintained by employing voluntary attention to the memory itself (recall) during the time of retention and results in a better performance in a later recognition task [2][3]. Although much studies on visual memory have focused on how a better performance in recognition task is achieved, little is known about the nature of this failure. The failure of retrieval accompanying forgetfulness is possibly related to memory deletion or false memory.

Visual short-term memory (VSTM) is known to be memory storage of visual information that has a limited capacity of at least four items, depending on stimulus complexity and the organization of objects in the memory array [4], and lasts approximately from one second to several seconds from the onset of the memory array. Memory before one second from the onset is best understood with iconic or sensory memory which stores a representation with unlimited capacity but decaying more quickly and is more distracted by eye movements and intervening stimuli compared with VSTM. It has been

shown that several brain regions are associated with VSTM. The prefrontal cortex is associated with the maintenance of working memory and shows increased activity with memory load [6][7][8]. The inferior intraparietal sulcus (IPS) participates in encoding and maintenance of about four objects at different spatial locations independent of object complexity whereas superior IPS and lateral occipital complex participates in that of variable capacity depending on object complexity [5].

Employing attention to an external visual stimulus [1][9][10] as well as that to an internal representation of a stimulus that is stored in VSTM [2][3] gives rise to a better memory performance. Visual attention to an item in the external world enhances processing of information for that item, whereas suppressing that for unattended items through gating the processing resources on perception [1][3], activating posterior parietal cortex and extrastriate cortex [12]. On the other hand, attention to internal representation is different from the visual attention in respect of targets because the internal representation in VSTM is encoded in the brain in advance, giving a better performance in a later recognition task [2][3]. In addition, attention to internal representation selectively activates several frontal lobe regions, although there is also an overlapping mechanism associated with orienting attention to locations in external and internal representations in parietal, frontal, and occipital areas [11]. A question is whether memory of unattended items that were once encoded in the brain in advance attenuates with or without attention to memory of an item in VSTM. In other words, memory of unattended items, or not recalled items, may be deleted or mistaken with other items.

Griffin and Nobre first described attention directed to internal representations of enhanced object memories in a similar manner to attention directed to external stimuli [3]. In their experiment, an informative cue was presented either before (pre-cue) or after (retro-cue) an array of four coloured crosses followed by a memory probe. Each cue had 80 % validity to targets, consisting of two distinct conditions valid and invalid (in the so-called Posner paradigm [12]) together with a neutral condition that was cued with a non-informative signal. The results showed that, as well as the enhancement of accuracy for valid objects, accuracy for invalid objects was significantly impaired when directing attention to both the external and internal. They did not, however, address contents of memory degradation, namely memory deletion and interference in the invalid condition. In case of retro-cues that prompt attention to the internal representation, differing from pre-cues, the array might be once encoded and stored in VSTM.

Here we investigate these issues by instructing the subjects to direct attention to visual memory of an object after multiple retention intervals. A delayed match paradigm was designed to investigate the effect of retention intervals when attention was oriented to visual memory (internal representation) of an object within VSTM during its retention (Figure 1). Attention was controlled by a predictive symbolic cue which indicates a possible target identity, using the Posner paradigm [12]. In 67 % of total trials, targets appeared with the cued identities. In half of the remaining trials, targets appeared with an identity other than the cued identity. In the other half, targets appeared with a novel identity (But see table 1). Nonsensical figures were used to control subjective familiarity with the stimuli and to minimize the effect of verbal encoding. They were also easily distinguishable and prone to forgetting so as to facilitate investigation of the nature of mere VSTM.

2 Materials and Methods

Subjects: 8 subjects (4 females and 4 males, age 25-32, with an average of 28.1) participated in this experiment. All subjects were right-handed by self-report and had normal or corrected-to-normal vision. The subjects gave written informed consent after being explained about the purpose and nature of the experiments. The stimuli were presented on a computer screen. The subjects responded by key pressing.

Stimuli and apparatus: Nonsensical figures were generated by the following algorithm. 20 invisible vertices were randomly chosen by the computer on a 120 x 120 pixel bitmap with black background, of which 2 vertices were randomly chosen to draw a white line segment between them. The nonsensical figures were composed of 5 line segments each. The 5 segments composition was designed to avoid a resemblance to the letters while minimizing the complexity of the figure. A single vertex may be shared by more than one line segments.

Procedure: The experiment consisted of 18 blocks with 24 trials each. A trial started with a 300 ms fixation followed by the study phase. In the study phase, an array of 4 nonsensical figures was presented in the centre of the screen for 800 ms followed by retention intervals of multiple durations (0, 1500, 3000 ms). After the retention intervals, a cue was presented for 1400 ms. After the presentation of the cue and a 100 ms interval, the probe was displayed in the centre of the screen until the subjects responded with a key press. The cue was an arrow (i.e. top left, top right, left bottom, and right bottom) displayed in the centre of screen predicting one of four identities of a figure that was previously shown in the study phase, whereas the probe was one of four figures displayed at the study phase of the same trial or a novel figure. There were 67 % “valid-trials” (where the correct answer is “valid”), 16.5 % “invalid-trials” (where the correct answer is “invalid”) and 16.5 % “novel-trials” (where the correct answer is “novel”). Further details of number of trials are shown in Table 1. One picture was displayed in only a single trial so that all pictures apparently had comparable familiarity level. The inter-trial interval was 2000 ms.

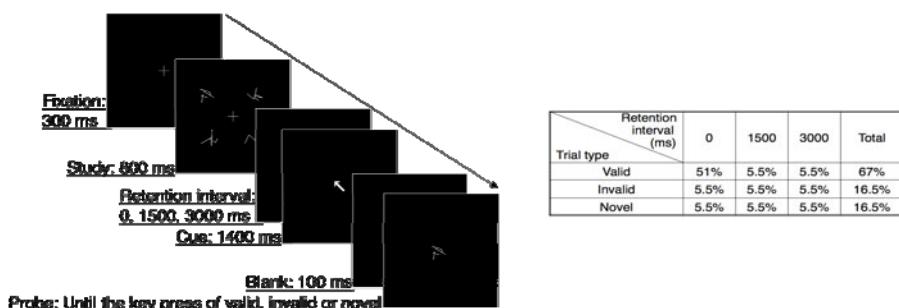


Fig. 1. Time course of a trial. A predictive cue indicating a possible probe identity was presented after multiple retention intervals (0, 1500, 3000 ms). Subjects were instructed to answer VALID, INVALID or NOVEL in a three-alternative forced-choice at the probe phase. Four nonsensical figures consisting of 5 lines were randomly generated by the computer.

Table 1. Percentage of numbers of trials in the designed conditions in the experiment.

Before the experiment, the subjects were informed that a single block consisted of 24 trials and there would be a total of 18 blocks. They sat comfortably in front of the screen at the distance of 60 cm. They were instructed to look at the crisscross throughout its presentation, to remember the object identity in the study phase and to answer a task question (“valid” “invalid” or “novel” with a key press using right index, middle or third finger, respectively) as accurately and quickly as possible. After the instruction, the subjects practiced the tasks.

3 Results

Hit rate of attended objects (VALID response in the valid trial) decreased significantly when retention intervals were increased (one way ANOVA, $p<0.001$; between 0 and 1500 ms retention interval, Tukey post hoc test, $p<0.05$; between 0 and 3000 ms retention interval, Tukey post hoc test, $p<0.001$) while the hit rate of invalid objects (INVALID response in the invalid trial) and of novel objects (NOVEL response in the novel trial) remained unchanged ($p>0.1$) (Figure 2 (a)). Comparison of trial types on 0 ms retention interval by Tukey post hoc test revealed significant differences between the valid and novel trials ($p<0.001$) and between the invalid and novel trials ($p<0.05$) (Figure 2 (a)). d' in each trial was calculated in a case of three alternative forced choice with chance level 33% (Figure 2 (c)). The statistical significances were the same as the hit rate above.

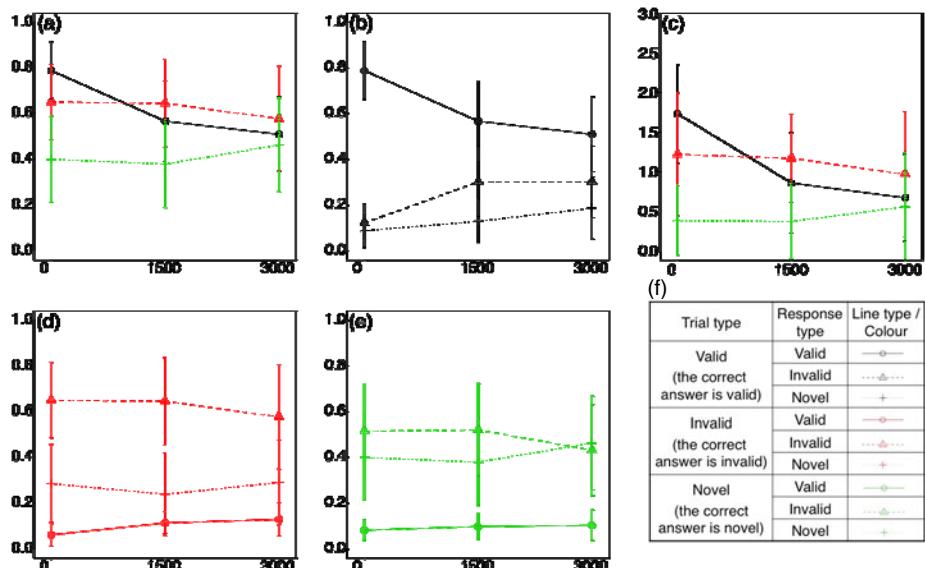


Fig. 2. (a) Hit rate in each trial type (the rate of VALID response in valid trial, INVALID response in invalid trial or NOVEL response in novel trial). (c) Values of d' in each trial type. (b),(d),(e) Rate of response types in the valid, invalid, and the novel trials, respectively (see Table 1). (f) Line type and colour. Colours of the line indicate the trial types: black for the valid trials, red for the invalid trials and green for the novel trials. Types of the line indicate the response type: solid lines for valid, long dashed lines for invalid and short dashed lines for novel. The horizontal axis is the retention intervals (ms). The bars give the standard errors.

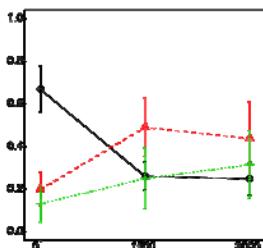


Fig. 3. Rate of responses irrespective to the trial types. Colours of the line indicate the response types: black for the VALID response, red for INVALID response and green for NOVEL response. Note that numbers of trials conducted in the experiment are shown in Table 1.

With an increase in failure in recognition for valid objects after longer retention intervals, wrong answers of invalid and novel objects were increased (one way ANOVA for invalid objects, $p<0.001$; between 0 and 1500 ms retention interval, Tukey post hoc test, $p<0.05$; between 0 and 3000 ms retention interval, Tukey post hoc test, $p<0.05$ and one way ANOVA for novel objects, $p<0.05$; but none of combinations among 0, 1500 and 3000 ms retention interval by Tukey post hoc test revealed significant differences) (Figure 2 (b)). Paired t-tests were performed on two types of the wrong answers for each retention interval. Only with the interval of 1500 ms, the rate of INVALID response was significantly higher than that of NOVEL response ($p<0.05$), indicating that the failure in recognition of a valid object may result in having false memory (i.e. of an unattended object) rather than memory deletion at that retention interval. As hit rate of the invalid trial (INVALID response in the invalid trial) and that of the novel trial (NOVEL response in the novel trial) were not changed with increasing retention interval, neither wrong answers in the invalid trial (i.e. VALID and NOVEL response) or that in the novel trial (i.e. VALID and INVALID response) were changed with increasing retention interval.

There is also the possibility that the set of data in the valid trial includes false alarms from other trial types. To exclude the mutual effects of trial types, the mere rates of response types (= number of each response / total number (= 432) of trials) were analyzed (Figure 3). Although the numbers of trials were designed asymmetrically to control for the validity of cues (Table 1), the result shows significantly different tendencies (Figure 3). The rate of VALID response decreased (one way ANOVA, $p<0.001$; between 0 and 1500 ms retention interval and 0 and 3000 ms retention interval, Both Tukey post hoc test, $p<0.001$) whereas the rate of both INVALID and NOVEL response increased (both one way ANOVA, $p<0.001$; between 0 and 1500 ms retention interval with NOVEL response. Tukey post hoc test, $p<0.0001$; between 0 and 3000 ms retention interval with NOVEL response. Tukey post hoc test, $p<0.01$; between 0 and 3000 ms retention interval with NOVEL response. Tukey post hoc test, $p<0.05$). When the retention intervals were 1500 ms, the rate of INVALID response was significantly higher than that of both VALID and NOVEL responses (Tukey post hoc test, both $p<0.05$). In post-experimental interview, all subjects reported that they realized that the valid trials were more frequent but did not realize the difference among retention intervals within the valid trial (see Table 1).

4 Discussion

The nature of memory degradation when orienting attention to the internal representation (one of remembered objects) in VSTM was investigated. Unlike previous studies [2][3], our experiments provide the time course effect on retrieval VSTM. As the retention interval between the offset of stimuli and orienting attention increased, only memory of attended objects degenerated while hit rate of unattended objects (i.e. INVALID response in the invalid trial and NOVEL response in the novel trial) did not (Figure 2 (a)). Thus, attention did not simply bring about a better memory performance but rather it seemed to make a gradient on VSTM. Orienting attention to more abstract VSTM with a long retention may lead to a quick decay of VSTM. Further study is necessary to investigate whether orienting attention to internal representation actually causes a quick decay after a long retention interval.

To investigate the nature of forgetfulness in VSTM, we focused on wrong answers in the valid trial. Both INVALID response and NOVEL response increased with increasing retention interval, while the rate of INVALID response and NOVEL response did not make a difference at both 0 and 3000 ms retention interval (Figure 2 (b)). The responses of both INVALID and NOVEL have a similar tendency and these responses in daily life may often appear coincidentally. Here we show a separation of INVALID and NOVEL response in wrong answers at the time of 1500 ms retention interval from the offset of memory array. In the valid trial, response of INVALID is interpreted as interference within VSTM. Having a false memory and response of NOVEL is considered to be the deletion of memory. Thus, as forgetfulness increases with a longer retention, there is a tendency to have a false memory more than memory deletion at an early time of retention. And later, the difference in the level of having false memory and memory deletion becomes insignificant. Once the VSTM is established, it would be maintained by recurrent neural activation during short-term period [13] rather than by physical changes on the structure of connectivity such as the long-term potentiation. Such recurrent activation would maintain the VSTM, degrading over time. When the VSTM is retained at a certain level at 1500 ms retention interval, attention that activates neurons in several brain regions may distract the recurrent activation of VSTM, as much as bringing out a mistake for the other remembered objects, but not as much as causing a deletion of VSTM. This may explain why VSTM may be distorted by orienting attention to memory of an object that leads to a new wave of activations on a circuit. Consequently, it is possible that forgetfulness in VSTM is primarily due to interference rather than memory deletion.

The effect of Posner paradigm was analyzed in the mere rates of response types in all of trial types (Figure 3). The experiment was designed in an asymmetrical manner regarding the number of trials (Table 1). Only the number of the valid trial at 0 ms retention interval was large and the subjects actually chose VALID the most often at 0 ms retention interval. If subjects had been successfully biased with this asymmetric number of trials (Posner paradigm), they would have tended to choose VALID the most frequently. However, at 1500 ms and later the rates of VALID response were low and stayed around the chance level of 33 %. Interestingly, the rate of INVALID response at 1500 ms retention interval was significantly higher than that of VALID and NOVEL. There are two possibilities: The subjects were not biased. Or, they were biased but tended to choose INVALID response more. According to the subject's reports, they did not strategically respond to retention intervals, such that they responded equally to three

alternatives for the longer intervals. Suppose they did so without being aware, the rate of response cannot be reversed at 1500 ms retention interval. Thus the latter hypothesis may be more appropriate. As discussed above, if there is distractive effect with attention to internal representation at the intermediate retention period, interference of memory may be more often the case than deletion of memory.

5 Conclusion

When the retention intervals increased, success in the recognition of attended (valid) objects decreased while that of unattended or novel objects did not. This failure in recognition of attended objects as early as 1500 ms retention interval from offset of a memory array is possibly due to having a false memory. Further study is necessary to investigate the effect of attention to internal representation stored in VSTM.

Acknowledgments. We thank all laboratory members for assistance and effective discussion.

References

1. Mack, A., Rock, I.: Inattentional Blindness. MIT Press, Cambridge (1998)
2. Makovski, T., Sussman, R., Jiang, Y.: Orienting Attention in Visual Working Memory Reduces Interference From Memory Probes. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 34(2), 369–380 (2008)
3. Griffin, I.C., Nobre, A.: Orienting Attention to Locations in Internal Representations. *J.Cogn. Neurosci.* 15(8), 1176–1194 (2003)
4. Sligach, I.G., Scholte, H.S., Lamme, V.: Are there multiple visual short-term memory stores? *PLOS one* 3(2), e1699 (2008)
5. Xu, Y., Chun, M.: Dissociable neural mechanisms supporting visual short-term memory for objects. *Nature* 440(7080), 91–95 (2006)
6. Pessoa, L., Gutierrez, E., Bandettini, P., Ungerleider, L.: Neural correlates of visual working memory: fMRI amplitude predicts task performance. *Neuron* 35(5), 975–987 (2002)
7. Cohen, J.D., Perlstein, W.M., Braver, T.S., Nystrom, L.E., Noll, D.C., Jonides, J., Smith, E.: Temporal dynamics of brain activation during a working memory task. *Nature* 386(6625), 604–608 (1997)
8. Funahashi, S.: Prefrontal cortex and working memory processes. *Neuroscience* 139(1), 251–261 (2006)
9. Vuilleumier, P., Schwartz, S., Duhoux, S., Dolan, R.J., Driver, J.: Selective attention modulates neural substrates of repetition priming and “implicit” visual memory: suppressions and enhancements revealed by fMRI. *J. Cogn. Neurosci.* 17(8), 1245–1260 (2005)
10. Prinzmetal, W., McCool, C., Park, S.: Attention Reaction Time and Accuracy Reveal Different Mechanisms. *J Exp Psychol Gen* 134(1), 73–92 (2005)
11. Nobre, A.C., Coull, J.T., Maquet, P., Frith, C.D., Vandenberghe, R., Mesulam, M.: Orienting Attention to Locations in Perceptual Versus Mental Representations. *J. Cogn. Neurosci.* 16(3), 363–373 (2004)
12. Posner, M.: Orienting of attention. *Q J Exp Psychol.* 32(1), 3–25 (1980)
13. Constantinidis, C., Wang, X.: A neural circuit basis for spatial working memory. *Neuroscientist* 10(6), 553–565 (2004)
14. Hoshino, E., Mogi, K.: Trade-off in the Effect of Attention for Visual Short-term Memory. *Association for the Scientific Study of Consciousness* 14 (2010)

Novel Alternating Least Squares Algorithm for Nonnegative Matrix and Tensor Factorizations

Anh Huy Phan¹, Andrzej Cichocki^{1,*}, Rafal Zdunek^{1,2}, and Thanh Vu Dinh³

¹ Lab for Advanced Brain Signal Processing, Brain Science Institute - RIKEN
Wako-shi, Saitama 351-0198, Japan

² Institute of Telecommunications, Teleinformatics and Acoustics, Wroclaw, Poland
³ HoChiMinh City University of Technology, Vietnam

Abstract. Alternative least squares (ALS) algorithm is considered as a “work-horse” algorithm for general tensor factorizations. For nonnegative tensor factorizations (NTF), we usually use a nonlinear projection (rectifier) to remove negative entries during the iteration process. However, this kind of ALS algorithm often fails and cannot converge to the desired solution. In this paper, we proposed a novel algorithm for NTF by recursively solving nonnegative quadratic programming problems. The validity and high performance of the proposed algorithm has been confirmed for difficult benchmarks, and also in an application of object classification.

Keywords: PARAFAC, nonnegative tensor factorization, NMF, nonnegative quadratic programming, parallel computing, ALS, object classification.

1 Introduction

PARAFAC and nonnegative PARAFAC (or nonnegative tensor factorization - NTF) have recently been found many applications [1][2][3]. NMF algorithms can be straightforwardly extended to NTF using unfolding approach [2][4][5]. For instance, the multiplicative LS or KL NTF update rules minimize the Frobenius cost function or the Kullback-Leiber divergence [2]. The “work-horse” Alternating Least Squares (ALS) algorithm has been experimentally proved to work very well on general data. However, this extended algorithm for NTF using a rectifier for estimated factors $[x]_+ = \max(x, 0)$ often fails, and hardy converges to the desired solution. The HALS algorithm [4][5] was proposed to employ the ALS algorithm but to estimate only one component for each factor. This algorithm not only improves the ALS algorithm but also is proved to outperform the multiplicative algorithms [6]. However, for tensor composed by highly collinear factors, both multiplicative and HALS algorithms often fail to factorize such data.

* Also from Dept. EE Warsaw University of Technology and Systems Research Institute, Polish Academy of Science, POLAND.

In this paper, we propose a novel ALS algorithm for NTF by recursively solving nonnegative quadratic problems. The proposed algorithm is verified to overwhelmingly outperform “state-of-the-art” algorithms for difficult benchmarks with bottlenecks, swamps. A real-world application in object classification is also considered in the experimental section.

2 Tensor Notation and Basic Models

We shall denote a tensor by underlined bold capital letters, e.g., $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, matrices by bold capital letters, e.g. $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_J]$, and vectors by bold italic lowercase letters, e.g. \mathbf{a}_j . The Khatri-Rao, Kronecker, and Hardamard products are denoted respectively by $\odot, \otimes, \circledast$. The product of a tensor and a matrix along the mode- n is defined as $\underline{\mathbf{Y}} = \underline{\mathbf{G}} \times_n \mathbf{A}$, or $\mathbf{Y}_{(n)} = \mathbf{A} \mathbf{G}_{(n)}$. Multiplication of a tensor with all but one mode is defined as

$$\underline{\mathbf{G}} \times_{-n} \{\mathbf{A}\} = \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \dots \times_2 \mathbf{A}^{(2)} \dots \times_{n-1} \mathbf{A}^{(n-1)} \times_{n+1} \mathbf{A}^{(n+1)} \dots \times_N \mathbf{A}^{(N)}.$$

PARAFAC [7] can be formulated as follows, “Factorize a given N -th order data tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \dots \times I_N}$ into a set of N component matrices (factors): $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \dots, \mathbf{a}_R^{(n)}] \in \mathbb{R}^{I_n \times R}$, ($n = 1, 2, \dots, N$) representing the common (loading) factors”, that is, $\underline{\mathbf{Y}} \approx \underline{\mathbf{I}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \times_N \mathbf{A}^{(N)} = \widehat{\underline{\mathbf{Y}}}$, where $\|\mathbf{a}_r^{(n)}\|_p = 1$ for $n = 1, 2, \dots, N-1$, $r = 1, 2, \dots, R$, and $p = 1, 2$. Tensor $\widehat{\underline{\mathbf{Y}}}$ is an approximation of the data tensor $\underline{\mathbf{Y}}$, and tensor $\underline{\mathbf{I}}$ is an identity tensor.

The ALS algorithm for the (nonnegative) matrix and tensor factorizations uses the update rules for factors $\mathbf{A}^{(n)}$ given by

$$\mathbf{A}^{(n)} = \left[\mathbf{Y}_{(n)} \{\mathbf{A}\}^{\odot-n} \left(\{\mathbf{A}^T \mathbf{A}\}^{\circledast-n} \right)^{-1} \right]_+, \quad (n = 1, 2, \dots, N). \quad (1)$$

where $\mathbf{Y}_{(n)}$ is the mode- n matricized version of tensor $\underline{\mathbf{Y}}$, and the Hadamard product is given by $\{\mathbf{A}\}^{\circledast-n} = \mathbf{A}^{(N)} \circledast \dots \circledast \mathbf{A}^{(n+1)} \circledast \mathbf{A}^{(n-1)} \dots \circledast \mathbf{A}^{(1)}$. The ALS algorithm with a rectifier (projection) may not converge to the solution without additional regularization parameters. In the sequence, we present the novel algorithm for NTF based on alternating least squares approach.

3 Nonnegative Quadratic Programming

In order to derive a novel algorithm for NTF, we introduce the following non-negative quadratic programming problem.

Problem 1 (Nonnegative Quadratic Programming). Consider the quadratic programming problem with nonnegative constraints

$$\text{minimize} \quad f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} - \mathbf{b}^T \mathbf{x} \quad \text{subject to} \quad \mathbf{x} \geq 0, \quad (2)$$

where $\mathbf{Q} \in \mathbb{R}^{R \times R}$ is a nonnegative symmetric semi-positive definite matrix, $\mathbf{b} = [b_1 b_2 \dots b_R]^T$, $\mathbf{x} = [x_1 x_2 \dots x_R]^T$. This problem has an unique global optimal solution $\mathbf{x}^* = [x_1^* x_2^* \dots x_R^*]^T$, $x_r^* \geq 0, \forall r$.

Problem 1 can be solved by consideration of two Lemmas 1 and 2.

Lemma 1. *A stationary point of function (2) is denoted by*

$$\tilde{\mathbf{x}} = [\tilde{x}_1 \ \tilde{x}_2 \ \cdots \ \tilde{x}_R]^T = \mathbf{Q}^{-1} \mathbf{b}. \quad (3)$$

If $\exists \tilde{x}_r < 0$, then $x_r^* = 0$.

Proof. Without loss of generality, we assume that $\tilde{x}_R < 0$. The gradient of (2) with respect to x_R at point x_1^*, \dots, x_{R-1}^* given by

$$g(x_R) = \nabla_{x_R} f(x_1^*, \dots, x_{R-1}^*, x_R) = q_{R,R} x_R + \mathbf{q}_{R,1:R-1} \mathbf{x}_{1:R-1}^* - b_R \quad (4)$$

has a solution given by $\tilde{x}_R = \frac{b_R - \mathbf{q}_{R,1:R-1} \mathbf{x}_{1:R-1}^*}{q_{R,R}}$. Assume that $x_R^* > 0$, by taking into the assumption of $f(x_1^*, \dots, x_{R-1}^*, x_R^*) \leq f(x_1^*, \dots, x_{R-1}^*, x_R), \forall x_R \in [0, +\infty)$, we straightforwardly obtain $x_R^* \equiv \tilde{x}_R$. That means \mathbf{x}^* is the solution of equation (3): $\mathbf{Q} \mathbf{x}^* = \mathbf{b}$. Consequently, $\mathbf{x}^* \equiv \tilde{\mathbf{x}}$ or $x_R^* = \tilde{x}_R > 0$. That conflicts to the above assumption $\tilde{x}_R < 0$. Therefore, we have $x_R^* = 0$.

Lemma 2 (Solution of Problem 1). *Solution \mathbf{x}^* of Problem 1 can be found under a recursive formulation.*

Proof. The gradient of the function $f(\mathbf{x})$ with respect to \mathbf{x} given by $\nabla f(\mathbf{x}) = \mathbf{Q} \mathbf{x} - \mathbf{b}$ has an unique solution $\tilde{\mathbf{x}} = [\tilde{x}_1 \ \tilde{x}_2 \ \cdots \ \tilde{x}_R]^T = \mathbf{Q}^{-1} \mathbf{b}$. It is straightforward that if $\tilde{\mathbf{x}} \geq 0$, then $\mathbf{x}^* \equiv \tilde{\mathbf{x}}$ is solution of (2). For another case, we assume that there are $K \leq R$ nonnegative entries $\bar{\mathcal{I}} = \{r : \tilde{x}_r < 0, 1 \leq r \leq R\}$, $\text{card}\{\bar{\mathcal{I}}\} = K$. Based on Lemma 1, all the corresponding variables are zeros $x_{r_k}^* = 0$. The rest $(R - K)$ variables are solutions of a similar Problem 1 but of an lower order $(R - K)$

$$\text{minimize } f(\mathbf{x}_{\mathcal{I}}) = \frac{1}{2} \mathbf{x}_{\mathcal{I}}^T \mathbf{Q}_{\mathcal{I}} \mathbf{x}_{\mathcal{I}} - \mathbf{b}_{\mathcal{I}}^T \mathbf{x}_{\mathcal{I}} \quad \text{subject to } \mathbf{x}_{\mathcal{I}} \geq 0, \quad (5)$$

where $\mathbf{Q}_{\mathcal{I}}$ is a part of the matrix \mathbf{Q} deleted K rows and K columns with indices $\bar{\mathcal{I}}$, and $\mathbf{b}_{\mathcal{I}}$ is a part of the vector \mathbf{b} removed K entries $\bar{\mathcal{I}}$. This establishes a recursive formulation to find nonnegative solution of (2). This procedure described in Algorithm 1 iterates until there are not any zero entry \tilde{x} , that means $K = 0$.

Algorithm 1. Recursive Algorithm for NQP

Input: \mathbf{Q} : nonnegative symmetric matrix ($R \times R$), \mathbf{b} : vectors of R entries
Output: \mathbf{x} : nonnegative vector minimizes (2)

```

1 begin
2   |    $\mathcal{I} = \{1, 2, \dots, R\}$ 
3   |   repeat
4   |   |    $\tilde{\mathbf{x}}_{\mathcal{I}} = \mathbf{Q}_{\mathcal{I}}^{-1} \mathbf{b}_{\mathcal{I}}$ ;  $\mathcal{I}_- = \{r \in \mathcal{I} : \tilde{x}_r < 0\}$ ;  $\mathcal{I} \leftarrow \mathcal{I} \setminus \mathcal{I}_-$ 
5   |   until  $\mathcal{I}_- = \emptyset$ 
6   |    $\mathbf{x} = \max\{0, \tilde{\mathbf{x}}\}$ 
7 end

```

4 Novel Alternative Least Square Algorithm

The proposed algorithm will be derived based on the ALS minimization of the squared Euclidean distance (Frobenius norm) to estimate the factor $\mathbf{A}^{(n)}$ while the others fixed.

$$\begin{aligned} D &= \frac{1}{2} \|\underline{\mathbf{Y}} - \hat{\underline{\mathbf{Y}}}\|_F^2 = \frac{1}{2} \|\underline{\mathbf{Y}}\|_F^2 + \frac{1}{2} \|\hat{\underline{\mathbf{Y}}}\|_F^2 - \langle \underline{\mathbf{Y}}, \hat{\underline{\mathbf{Y}}} \rangle \\ &= \frac{1}{2} \|\underline{\mathbf{Y}}\|_F^2 + \frac{1}{2} \text{vec}(\mathbf{A}^{(n)T})^T (\mathbf{I}_{I_n} \otimes \boldsymbol{\Gamma}^{(n)}) \text{vec}(\mathbf{A}^{(n)T}) \\ &\quad - \text{vec}(\boldsymbol{\Phi}^{(n)T})^T \text{vec}(\mathbf{A}^{(n)T}), \end{aligned} \quad (6)$$

where $\boldsymbol{\Gamma}^{(n)} = \{\mathbf{A}^T \mathbf{A}\}^{\circledast-n} \in \mathbb{R}^{R \times R}$, and $\boldsymbol{\Phi}^{(n)} = \mathbf{Y}_{(n)} \mathbf{A}^{\odot-n} \in \mathbb{R}^{I_n \times R}$. The first term in (6) $\|\underline{\mathbf{Y}}\|_F^2$ is constant, hence it can be ignored. The last two terms in (6) can be formulated as a minimization problem in the form of Problem 11

$$\min \quad f(\mathbf{v}^{(n)}) = \frac{1}{2} \mathbf{v}^{(n)T} \boldsymbol{\Gamma} \mathbf{v}^{(n)} - \text{vec}(\boldsymbol{\Phi}^{(n)T})^T \mathbf{v}^{(n)}, \quad \text{s.t. } \mathbf{v}^{(n)} \geq 0, \quad (7)$$

where $\boldsymbol{\Gamma} = \mathbf{I}_{I_n} \otimes \boldsymbol{\Gamma}^{(n)} \in \mathbb{R}^{I_n R \times I_n R}$ is a sparse symmetric semi-positive definite matrix, and $\mathbf{v}^{(n)}$ is a vectorized version of the factor $\mathbf{A}^{(n)}$: $\mathbf{v}^{(n)} = \text{vec}(\mathbf{A}^{(n)T})$. The factor $\mathbf{A}^{(n)}$ will be updated using Algorithm 11

It is worth noting that stationary point of the function (7) without non-negative constraints given by $\tilde{\mathbf{v}} = \boldsymbol{\Gamma}^{-1} \text{vec}(\boldsymbol{\Phi}^{(n)T}) = \text{vec}((\boldsymbol{\Phi}^{(n)} \boldsymbol{\Gamma}^{(n)-1})^T)$ is indeed a vectorized version of the ALS algorithm (11). If there is not any negative entry, the factor $\mathbf{A}^{(n)}$ is exactly the ALS update. Otherwise, we solve the reduced system as in (5). Pseudo-code of the proposed algorithm is listed in Algorithm 2. The function nqp refers to as Algorithm 11

Algorithm 2. NQP for Nonnegative PARAFAC - QALS

Input: $\underline{\mathbf{Y}}$: tensor $I_1 \times I_2 \times \dots \times I_N$,
 R : number of approximation components
Output: $\mathbf{A}^{(n)}$: N nonnegative factors of size $(I_n \times R)$ minimize the problem (6).

```

1 begin
2   Random or leading singular vectors to initialize for  $\mathbf{A}^{(n)}$ 
3   repeat
4     for  $n = 1$  to  $N$  do
5        $\text{vec}(\mathbf{A}^{(n)T}) = \text{nqp}(\mathbf{I}_{I_n} \otimes (\{\mathbf{A}^T \mathbf{A}\}^{\circledast-n}), \text{vec}((\mathbf{Y}_{(n)} \mathbf{A}^{\odot-n})^T))$ 
6     end
7   until a stopping criterion is met
8 end

```

5 Novel Algorithm for Low Memory Machine and Parallel Computing

The NQP problem (7) demands to solve a system of $I_n R$ variables. For large-scale data, it could demand high computational cost, and also large space cost. To this end, an alternative algorithm will be presented in this section to run on low memory machine, or in a parallel system with multiple nodes.

The cost function (6) forms I_n simultaneous nonnegative quadratic programming problems for all I_n rows $\mathbf{a}_{i_n:}^{(n)}$, $i_n = 1, 2, \dots, I_n$ of the factor $\mathbf{A}^{(n)}$

$$\min \quad f_{i_n}(\mathbf{a}_{i_n:}^{(n)}) = \frac{1}{2} \mathbf{a}_{i_n:}^{(n)} \mathbf{\Gamma}^{(n)} \mathbf{a}_{i_n:}^{(n)T} - \boldsymbol{\phi}_{i_n:} \mathbf{a}_{i_n:}^{(n)T}, \quad \text{s.t.} \quad \mathbf{a}_{i_n:}^{(n)} \geq 0$$

where $\boldsymbol{\phi}_{i_n:}^{(n)}$ is the i_n -th row of the matrix $\boldsymbol{\Phi}^{(n)}$. Each problem (8) requires to solve only R variables. Hence this demands much lower computational cost than solving Problem (7). Moreover, these simultaneous problems can be independently performed in a parallel system. The pseudo-code of this algorithm is given in Algorithm 3 in which **parfor** denotes the parallel loop.

Algorithm 3. Parallel Algorithm for Nonnegative PARAFAC - pQALS

Input: \mathbf{Y} : tensor $I_1 \times I_2 \times \dots \times I_N$, R : number of approximation components
Output: $\mathbf{A}^{(n)}$ N nonnegative factors of size $(I_n \times R)$ minimize the problem (6)

```

1 begin
2   Random or leading singular vectors to initialize for  $\mathbf{A}^{(n)}$ 
3   repeat
4     for  $n = 1$  to  $N$  do
5        $\mathbf{\Gamma}^{(n)} = \{\mathbf{A}^T \mathbf{A}\}^{\circ -n}$ ,  $\boldsymbol{\Phi}^{(n)} = \mathbf{Y}_{(n)} \mathbf{A}^{\odot -n}$ 
6       parfor  $i_n = 1$  to  $I_n$  do /* parallel loop for update  $\mathbf{A}^{(n)}$  */
7          $\mathbf{a}_{i_n:}^{(n)T} = \text{nqp}(\mathbf{\Gamma}^{(n)}, \boldsymbol{\phi}_{i_n:}^{(n)T})$ 
8       end
9     end
10    until a stopping criterion is met
11 end

```

6 Novel Algorithm for NMF

A simplified version of Algorithms 2 and 3 for 2 dimensional data (matrix) formulates the novel algorithms for NMF described by the following model $\mathbf{Y} \approx \mathbf{A} \mathbf{X}^T$, where $\mathbf{Y} \in \mathbb{R}_+^{I \times J}$ is the observed data, $\mathbf{A} \in \mathbb{R}_+^{I \times R}$, and $\mathbf{X} \in \mathbb{R}_+^{J \times R}$ are two nonnegative factors. The algorithm sequentially updates factors \mathbf{A} and \mathbf{X} by solving two nonnegative quadratic programming problems

$$\begin{aligned} \min f_{\mathbf{A}}(\mathbf{A}) &= \frac{1}{2} \operatorname{vec}(\mathbf{A}^T)^T (\mathbf{I}_I \otimes (\mathbf{X}^T \mathbf{X})) \operatorname{vec}(\mathbf{A}^T) - \operatorname{vec}(\mathbf{X}^T \mathbf{Y}^T)^T \operatorname{vec}(\mathbf{A}^T), \\ \text{s.t. } \mathbf{A} &\geq 0, \\ \min f_{\mathbf{X}}(\mathbf{X}) &= \frac{1}{2} \operatorname{vec}(\mathbf{X}^T)^T (\mathbf{I}_J \otimes (\mathbf{A}^T \mathbf{A})) \operatorname{vec}(\mathbf{X}^T) - \operatorname{vec}(\mathbf{A}^T \mathbf{Y})^T \operatorname{vec}(\mathbf{X}^T), \\ \text{s.t. } \mathbf{X} &\geq 0, \end{aligned}$$

7 Experiments

7.1 Factorization of Collinear Synthetic Tensors

In the first example, we constructed 3-D and 5-D synthetic tensors \mathbf{Y} with $I_n = 100$, $(n = 1, \dots, N)$. Factors comprising $R = 20$ components were drawn from uniform distribution, but were forced to be collinear with others by a simple modification $\mathbf{a}_r^{(n)} = \mathbf{a}_1^{(n)} + 0.5 \mathbf{a}_r^{(n)}$, $(r = 2, 3, \dots, R)$. All the algorithms were analyzed under the same experimental conditions, those were leading singular values for initialization, iteration until difference of consecutive relative errors $\varepsilon = \frac{\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2}{\|\mathbf{Y}\|_F^2}$ is lower than 1e-10, or the maximum number of iterations (1000) is exceeded: Comparison of performances of various algorithms is given in Table 1. The results were averaged over 100 runs, and standard deviations are inside brackets. For all the MC runs, the QALS algorithm achieved perfect factorizations with the lowest relative errors $\varepsilon \approx 1e-7$ and $1e-6$ after 1000 iterations for 3-D and 5-D tensors respectively. Moreover, in order to evaluate the estimation accuracy, the SIR index is calculated for the true and estimated components after permutation matching and normalization. Factors estimated by the QALS algorithm have high SIR values: 28.02 dB and 32.77 dB for 3-D and 5-D tensor factorizations, respectively. Those results overwhelmingly outperformed all the other algorithms. The HALS algorithm achieved better performances than those of the multiplicative KL and LS algorithms. A more intuitive visualization for convergences of algorithms is shown in Fig. 1(a). Multiplicative algorithms were stacked in local minima, and did not improve the performance after 400 iterations. The distributions of SIR indices obtained by NTF algorithms are shown in Fig. 1(b).

Table 1. Performance comparison for various algorithms. The relative errors are expressed in logarithmic scale. A successful algorithm is with low relative error and high SIR index or high classification accuracy (inside brackets are standard deviations).

Algorithm	Error	SIR (dB)	Error	SIR (dB)	Error	Accuracy (%)
	$N = 3, I_n = 100, R = 20$	$N = 5, I_n = 100, R = 20$	$COIL-20, R = 15$			
ALS					-0.5577(-2.91)	98.64 (0.96)
KL	-3.66(-4.43)	4.95 (4.79)				
LS	-3.93(-5.16)	12.83 (3.11)	-3.41(-4.58)	13.41 (2.22)	-0.5139(-2.91)	99.57 (0.42)
HALS	-5.48(-6.90)	13.39 (2.51)	-5.21(-7.09)	19.79 (1.38)	-0.5149(-2.96)	99.64 (0.42)
QALS	-7.11(-7.21)	28.02 (2.64)	-6.23(-6.54)	32.77 (3.57)	-0.5142(-2.96)	99.71 (0.42)

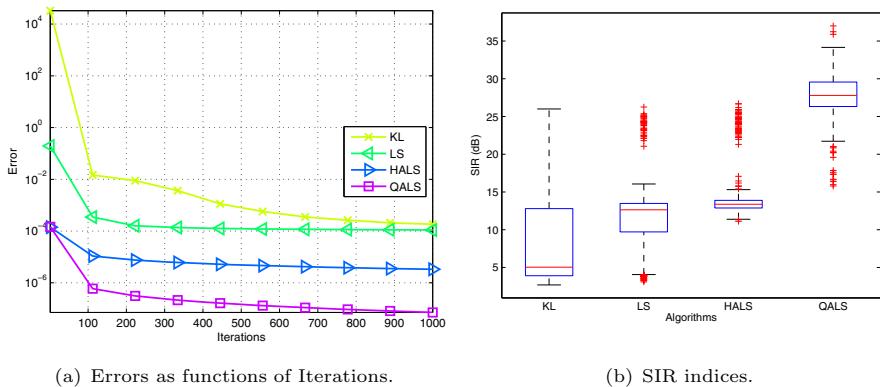


Fig. 1. Performance comparison of NTF algorithms for 3-D dimensional tensors for Example 7.1

7.2 Object Classification with the COIL-20 Dataset

In the second simulation, we considered the Columbia University Image Library (COIL-20) dataset [8] consists of 1,420 grayscale images of 20 objects (72 images per object) with a wide variety of complex geometric and reflectance characteristics. Each image was downsampled to 32×32 grayscale (0-255). The dataset was randomly divided into two separate sets with 720 training and 700 test images. The results were averaged over 100 trials. The training data was constructed as a 3-D tensor of size $32 \times 32 \times 720$ images. We factorized the training tensor to obtain basis factors with and without nonnegativity constraints. The ALS algorithm was also considered in this experiment. For all the methods, basis factors $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ were fixed to 15 components. To classify the data, we trained an SVM classifier [9]. With the same hold/out ratio of 50%, classification accuracies of the ALS algorithm for PARAFAC, the multiplicative LS, HALS and QALS algorithms for NTF are shown in Table 1. Although the ALS algorithm explained the data well (73.56% fitness), classification with its factors achieved

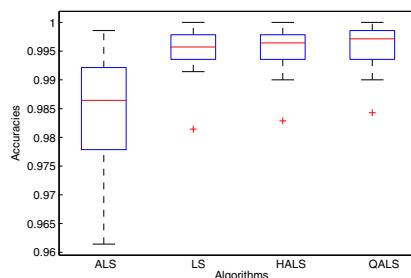


Fig. 2. Classification accuracies of NTF algorithms for Example 7.2

an accuracy of 98.64%. The QALS gave the highest accuracy of 99.71% while its fitness was only 69.39%. Distributions of the classification accuracies for four algorithms are shown in Fig. 2.

8 Conclusions

Novel algorithm for nonnegative PARAFAC has been proposed in this paper based on recursively solving the nonnegative quadratic programming problems. A variation of the proposed algorithm which is suitable for low memory or parallel computing is also presented. The performance of the proposed algorithm has been confirmed for difficult synthetic benchmarks and also for a classification problem.

References

1. 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)
2. 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)
3. Paatero, P.: A weighted non-negative least squares algorithm for three-way PARAFAC factor analysis. Chemometrics Intelligent Laboratory Systems 38(2), 223–242 (1997)
4. Cichocki, A., Phan, A.H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE (March 2009) (invited paper)
5. Phan, A.H., Cichocki, A.: Multi-way nonnegative tensor factorization using fast hierarchical alternating least squares algorithm (HALS). In: Proc. of The 2008 International Symposium on Nonlinear Theory and its Applications, Budapest, Hungary (2008)
6. Gillis, N., Glineur, F.: Nonnegative factorization and the maximum edge biclique problem. CORE Discussion Papers 2008064, Université catholique de Louvain, Center for Operations Research and Econometrics, CORE (2008)
7. 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)
8. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (coil-20). Technical Report CUCS-005-96, Columbia University (February 1996)
9. Canu, S., Grandvalet, Y., Guigue, V., Rakotomamonjy, A.: SVM and kernel methods –Matlab toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France (2005)

Computational Modeling and Analysis of the Role of Physical Activity in Mood Regulation and Depression

Fiemke Both, Mark Hoogendoorn, Michel C.A. Klein, and Jan Treur

VU University Amsterdam, Department of Artificial Intelligence

De Boelelaan 1081, 1081HV Amsterdam, The Netherlands

{fboth, mhoogen, mcaklein, treur}@cs.vu.nl

<http://www.few.vu.nl/~{fboth, mhoogen, mcaklein, treur}>

Abstract. Physical activity is often considered an important factor in handling mood regulation and depression. This paper presents a computational model of this role of physical activity in mood regulation. It is shown on the one hand how a developing depression can go hand in hand with a low level of physical activity, and on the other hand, how Exercise Therapy is able to reverse this pattern and make the depression disappear. Simulation results are presented, and properties are formally verified against these simulation runs.

Keywords: Mood and emotions, physical activity, depression, personal assistant agent.

1 Introduction

Avoiding negative moods may be a nontrivial challenge faced by a human organism. In the richer countries the number of persons struggling with longer periods of negative moods, like in a depression, is relatively high, and is expected to increase further (cf. [23]). To avoid or to recover from a negative mood requires mechanisms for mood regulation. Within neurological literature, mechanisms are investigated by which mood regulation takes place; for example, [9, 21]. In particular, also dysfunctioning of these mechanisms in persons in a depression is analyzed, for example in [1, 2, 19, 22, 24]. A specific area that receives more attention recently is the way in which brain functioning interacts with the amount of physical activity of a person; for example, it has been shown that a substantial level of physical activity has a positive effect on mood and brain plasticity; e.g., [8, 11, 12]. Indeed, it is known that physical exercise may be an effective way for humans to improve mood (cf. [4, 8]). In particular, this has been studied for the case of mood regulation mechanisms in depressions, among others by using an animal model for human learned helplessness; e.g., [15, 16, 17, 18, 25]. It turns out that the extent of learned helplessness can substantially decrease when more physical activity is undertaken. It has been shown that persons who perform a substantial amount of physical exercise suffer from depression less frequently (cf. [14]); therapies based upon physical exercise are reported, for example in (cf. [8, 13]) and are found to be probably effective (for meta-analyses e.g. [20, 10]).

Computational models of mechanisms for mood regulation are rather rare; one example is [6]. Such a model can be used to analyse how, and to which extent different

therapies can achieve a desired effect, as was shown in [7]. However, physical activity and its effect on mood and depression is not addressed this model. In this paper, inspired by the literature on the relevant biological mechanisms, a computational model is presented which models the role of physical exercise in mood regulation; the presented model subsumes the model described in [6] as well. Moreover, a therapy which is based upon performing physical exercise has also been integrated in the model to show how the principles and the workings of these therapies can be simulated and analyzed using the model. Example simulations that have been performed are presented to illustrate the functioning of the model, and the results have been analyzed by verifying particular desired properties against the simulation traces that have been generated.

This paper is organized as follows. In Section 2 the model for mood regulation and depression as taken from [6] is explained in more detail. The connection with physical exercise is introduced into the model in Section 3. Section 4 presents simulation results, whereas Section 5 verifies that these results indeed comply with existing theories within clinical psychology. Finally, Section 6 is a discussion.

2 A Model for the Role of Physical Activity in Mood Regulation

In this section it is discussed how the role of physical activity in mood regulation can be modeled computationally. In order to model the role of the physical activity, four states are distinguished, and the causal relationships between them and with other states in the mood regulation process are described. Among the represented relevant aspects are, for example, the impulse to undertake physical activity, the effect of physical activity on physical state, and the effect of physical state on mood, as indicated in the literature. These states and causal relationships are combined with other states and relationships in a mood regulation process. The latter types of states and relationships have been adopted from an existing mood regulation model based on psychological and neurological literature, in which physical activity and states were not covered (cf. [6]). Furthermore, in Section 3 the main effects of interventions by Exercise Therapy upon mood and depression are modeled in order to simulate and analyze effects of such a therapy.

Figure 1 shows an overview of the relevant states within the model and the relations between the states. In the figure, the states and lines that are depicted in grey represent states from the original model from [6]. States and dashed lines in black have been added to model the points of impact of the interventions as described in [7]. The thick black states and solid black lines represent the additional physical states.

States directly related to physical action. In order to model the role of physical activity of a person, four states are introduced. First, the *physical state* of a human is used, which expresses a combination of physical properties that are influenced by exercise: endorphin, monoamine and cortisol levels [12]. Exercise itself is represented in *objective physical value of situation*, the value of the current physical situation the human is in. The *long term physical norm* (or *LT physical norm*) expresses what physical state the human is striving for in the long term, whereas the *short term physical norm* (or *ST physical norm*) represents the goal for physical state on the shorter term. Each state is represented by a number in the interval $[0,1]$.

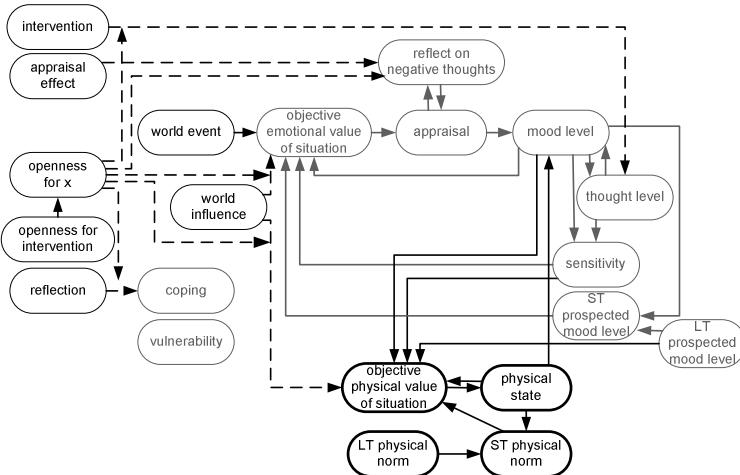


Fig 1. Model for mood and depression (grey states and lines indicate the original model [6], dashed lines and black states indicate intervention influences from [18], the thick black states represent the additional physical characteristics)

States not directly related to physical action. In addition to these states representing aspects of the processes directly related to physical activity, a number of other states relevant for mood regulation are used, adopted from the existing model in [6]. The *mood level* represents the current mood of the human, whereas *thoughts level* the current level of thoughts (i.e. the positivism of the thoughts). The state *objective emotional value of situation* represents the value of the situation a human is in (without any influence of the current state of mind of the human). The state *appraisal* represents the current judgment of the situation given the current state of mind (e.g. when you are feeling down, a pleasant situation might no longer be considered pleasant). The *long term prospected mood level* expresses what mood level the human is striving for in the long term, whereas the *short term prospected mood level* represents the goal for mood on the shorter term (in case you are feeling very bad, your short term goal will not be to feel excellent immediately, but to feel somewhat better). The *sensitivity* indicates the ability to select situations in order to bring the *mood level* to the *short term prospected mood level*. *Coping* expresses the ability of a human to deal with negative moods and situations, whereas *vulnerability* expresses how vulnerable the human is for getting depressed. Finally, *world event* indicates an external situation which is imposed on the human (e.g., losing your job). In addition to the states mentioned above, in [7] a number of states have been added to the model in [6] that relate to therapeutical interventions. First, a state representing the intervention (i.e., *intervention*) expressing that an intervention is taking place. The state *reflection on negative thoughts* expresses the therapeutic effect that the human is made aware of negative thinking about situations whereas the *appraisal effect* models the immediate effect on the appraisal of the situation. The *world influences* state is used to represent the impact of a therapy aiming to improve the *objective emotional value of situation* and possibly the *objective physical value of situation*. The *openness for intervention* is

a state indicating how open the human is for therapy in general, which is made more specific for each specific influence of the therapy in the state *openness for X*. Finally, *reflection* represents the ability to reflect on the relationships between various states, and as a result learn something for the future.

Dynamics directly related to physical activity. The states as explained above are causally related, as indicated by the arrows in Figure 1. These influences have been mathematically modeled. The *physical state* is regulated towards the *long-term physical norm* by choosing physical situations (*opvs*) that help increase or decrease the physical state towards the *short-term physical norm*. The long-term physical norm *lt_physnorm* is considered a personal characteristic. The *ST physical norm* is determined as follows: *physical state* can have an increasing or a decreasing influence, the *LT physical norm* has a regulating function.

$$\begin{aligned} st_physnorm(t+\Delta t) = & st_physnorm(t) + \\ & (vulnerability \cdot (phys_state(t) - lt_physnorm)) + coping \cdot (lt_physnorm - st_physnorm(t)) \cdot \Delta t \end{aligned}$$

The *objective physical value of situation* (*opvs*) is calculated in two steps. First, the change in *opvs* is calculated based on a physical regulation part using *physical state* and *ST physical norm* (*action_p1*) and a *mood* influence part (*action_p2*). When *mood level* is lower than *LT prospected mood level*, the action will have a lower value and therefore the human is less capable of choosing appropriate situations to increase the *physical state*. This models the idea that persons with a depression tend to be less physically active [14].

$$\begin{aligned} action_p1 = & opvs(t) + sens(t) \cdot (Neg(opvs(t) \cdot (st_phys_norm(t) - phys_state(t))) + \\ & Pos((1-opvs(t)) \cdot (st_phys_norm(t) - phys_state(t)))) \\ action_p2 = & opvs(t) + (1-sens(t)) \cdot (Neg(opvs(t) \cdot (mood(t) - prospmood)) + \\ & Pos((1-opvs(t)) \cdot (mood(t) - prospmood))) \\ action_p(t) = & w_{phys_reg} \cdot action_p1 + w_{mood} \cdot action_p2 \end{aligned}$$

Second, the new *opvs* is determined using the action from above and influence from the world to do physical activities (*world influence*).

$$\begin{aligned} opvs(t+\Delta t) = & opvs(t) + \\ & ((action_p(t) + openness(t) \cdot world_influence_p(t) \cdot (1 - action_p(t))) - opvs(t)) \cdot \Delta t \end{aligned}$$

Physical state is affected by *opvs*. The adaptation factor *adapt_phys* determines the speed with which the *physical state* changes.

$$phys_state(t+\Delta t) = phys_state(t) + adapt_phys \cdot (opvs(t) - phys_state(t)) \cdot \Delta t$$

Dynamics not directly related to physical activity. Next, a number of relations adopted from the model described in [6] are briefly discussed. The *objective emotional value of situation* (*oevs*) represents the situation selection mechanism of the human. First, the change in situation as would be selected by the human is determined (referred to as *action* in this case) as an intermediate step:

$$\begin{aligned} action(t) = & oevs(t) + sensitivity(t) \cdot (Neg(oevs(t) \cdot (st_prosp_mood(t) - mood(t))) + \\ & Pos((1-oevs(t)) \cdot (st_prosp_mood(t) - mood(t)))) \end{aligned}$$

In the equation, the *Neg(X)* evaluates to 0 in case *X* is positive, and *X* in case *X* is negative, and *Pos(X)* evaluates to *X* in case *X* is positive, and 0 in case *X* is negative. The formula expresses that the selected situation is more negative compared to the previous *oevs* in case the *short term prospected mood* is lower than the current mood and more positive in

the opposite case. Note that the whole result is multiplied with the *sensitivity*. The *action* in combination with the external influences now determines the new value for *oevs*:

$$\begin{aligned} oevs(t+\Delta t) &= oevs(t) + \\ &(world_event(t) \cdot (action(t) + openness(t) \cdot world_influence(t) \cdot (1 - action(t))) - oevs(t)) \cdot \Delta t \end{aligned}$$

The above equations basically take the value of actions as derived before in combination with the external influences (i.e. *world influence* and *world event*). The second step is that the human starts to judge the situation (i.e. *appraisal*) based upon his/her own state of mind:

$$\begin{aligned} appraisal(t+\Delta t) &= appraisal(t) + \\ &\alpha (\gamma + openness_intervention(t) \cdot reflect_neg_th(t) - appraisal(t)) \Delta t \end{aligned}$$

where

$$\gamma = (vulnerability \cdot oevs(t) \cdot thoughts(t) + coping \cdot (1 - (1 - oevs(t)) \cdot (1 - thoughts(t))))$$

The value of *appraisal* is determined by the *thoughts* of the human in combination with the *coping* skills and *vulnerability*. In addition, the intervention related state *reflection on negative thoughts* plays a role (i.e. being aware that you are judging the situation as more negative than a person without a depression) in combination with the openness to this type of intervention. The state *reflection on negative thoughts* is calculated as follows:

$$reflect_neg_th(t) = (basic_reflection(t) + appraisal_effect(t) \cdot openness_X(t)) \cdot (1 - appraisal(t))$$

Hence, the value increases based upon the *appraisal effect* of the intervention in combination with the *openness* to this specific part of the intervention. Furthermore, a *basic reflection* is expressed, which is the reflection already present in the beginning. Therapy can also dynamically change this *basic reflection* which can be seen as one of the permanent effects of therapy:

$$\begin{aligned} basic_reflection(t+\Delta t) &= basic_reflection(t) + \\ &\alpha \cdot intervention(t) \cdot learning_factor \cdot (1 - basic_reflection(t)) \Delta t \end{aligned}$$

The value for *mood* depends on a combination of the current *appraisal*, the *thoughts* and the *objective physical value of situation (opvs)*, whereby a positive influence (i.e. *thoughts*, *appraisal* and *opvs* are higher than *mood*) is determined by the *coping* and the negative influence by the *vulnerability*.

Thoughts is a bit more complex, and is expressed as follows:

$$\begin{aligned} thoughts(t+\Delta t) &= thoughts(t) + \alpha (\zeta + (1 - (thoughts(t) + \zeta)) \cdot intervention(t) \cdot w_{intervention}(t)) \Delta t \\ \text{where:} \end{aligned}$$

$$\begin{aligned} \zeta &= Pos(coping \cdot (appraisal(t) \cdot w_{appraisal_thoughts} + mood(t) \cdot w_{mood_thoughts} - thoughts(t))) - \\ &Neg(vulnerability \cdot (appraisal(t) \cdot w_{appraisal_thoughts} + mood(t) \cdot w_{mood_thoughts} - thoughts(t))) \end{aligned}$$

$$w_{intervention}(t+\Delta t) = w_{intervention}(t) + \alpha (openness_X(t) - w_{intervention}(t)) \Delta t$$

This indicates that *thoughts* are positively influenced by the fact that you participate in an intervention (you start thinking a bit more positive about the situation, you are in therapy). The weight of this contribution depends on the *openness* for the intervention at that time point. In addition, the *thoughts* can either be positively influenced due to the higher combination of the levels of *mood* and *appraisal* (again multiplied with the *coping*), or negatively influenced by the same state (whereby the *vulnerability* plays a role). The *sensitivity* is calculated in a similar manner (without the influence of therapy):

$$\text{sensitivity}(t+\Delta t) = \text{sensitivity}(t) + \alpha (\text{Pos}(\text{coping} \cdot (\eta - \text{sensitivity}(t))) - \text{Neg}(\text{vulnerability} \cdot (\eta - \text{sensitivity}(t)))) \Delta t$$

where

$$\eta = \text{mood}(t) \cdot w_{\text{mood_sens}} + \text{thoughts}(t) \cdot w_{\text{thoughts_sens}}$$

The *short term prospected mood* is calculated as follows:

$$\text{st_prospmood}(t+\Delta t) = \text{st_prospmood}(t) + \alpha (\text{vulnerability} \cdot (\text{mood}(t) - \text{lt_prospmood}) + \text{coping} \cdot (\text{lt_prospmood} - \text{st_prospmood}(t))) \Delta t$$

Dynamics of mood combining physical and nonphysical effects. Finally, the *mood level* combines effects from different (physical and non-physical) sides, represented by the concepts defined above. The effect of the physical state models the positive influence of physical exercise on mood [4] and the effectiveness of Exercise Therapy [4, 8, 13, 14, 23, 24].

$$\text{mood}(t+\Delta t) = \text{mood}(t) + \alpha (\text{Pos}(\text{coping} \cdot (\varepsilon - \text{mood}(t))) - \text{Neg}(\text{vulnerability} \cdot (\varepsilon - \text{mood}(t)))) \Delta t$$

where

$$\varepsilon = \text{appraisal}(t) \cdot w_{\text{appraisal_mood}} + \text{thoughts}(t) \cdot w_{\text{thoughts_mood}} + \text{phys_state}(t) \cdot w_{\text{opvs_mood}}$$

3 Modeling Exercise Therapy for Mood Regulation and Depression

Exercise Therapy consists of a structured form of exercise, often prescribed together with another intervention and/or anti-depressant medications. Physical exercise has a positive influence on mood [4] and can therefore aid in the recovery from a depression [8, 13, 14, 20, 10].

Figure 2 shows the mood regulation and depression model with the influences of Exercise Therapy. *Appraisal effect* is shown in light grey, because Exercise Therapy does not influence this concept. Furthermore, arrows from *mood level* and *physical state* to *coping* are added, to represent the learning of coping skills by doing more physical activities.

The intervention influences the human via four states: *thought level*, *objective physical value of situation (opvs)*, *coping* and *openness for ET*. The *thought level* is increased by the fact that the human is participating in an intervention, the formula is explained in Section 2. Second, the choice for a situation with a certain value for physical activeness (*opvs*) is influenced by *world influence*. Again, this effect is described in Section 2. The third effect of the intervention is on *coping skills*: learning the relationship between *mood* and *objective physical value of situation* results in better *coping* (as the human can now better cope with a lower mood since he/she knows that an option is to select more situations with a higher value for physical activeness). This is expressed as follows:

$$\begin{aligned} \text{coping}(t+\Delta t) &= \text{coping}(t) + \\ &\quad \alpha \text{ reflection}(t) \cdot w_{\text{reflection}}(t) \cdot (1 - |\text{opvs}(t) - \text{mood}(t)|) \cdot (1 - \text{coping}(t)) \Delta t \end{aligned}$$

where

$$w_{\text{reflection}}(t+\Delta t) = w_{\text{reflection}}(t) + \alpha (\text{openness_et}(t) - w_{\text{reflection}}(t)) \Delta t$$

When there is a moment of reflection in the therapy, the value for *coping* will increase as the difference between the *mood* and *ops* is perceived small (which makes it easy to see the relationship and improve *coping*). The last effect is that the openness for the specific therapy increases as the coping skills go up (since the human notices that the therapy works). Theta (θ) is an adaptation factor determining the speed with which openness changes.

$$\text{openness_et}(t+\Delta t) = \text{openness_et}(t) + \theta \cdot ((\text{coping}(t) - \text{coping}(t-\Delta t)) / \Delta t) \cdot \Delta t$$

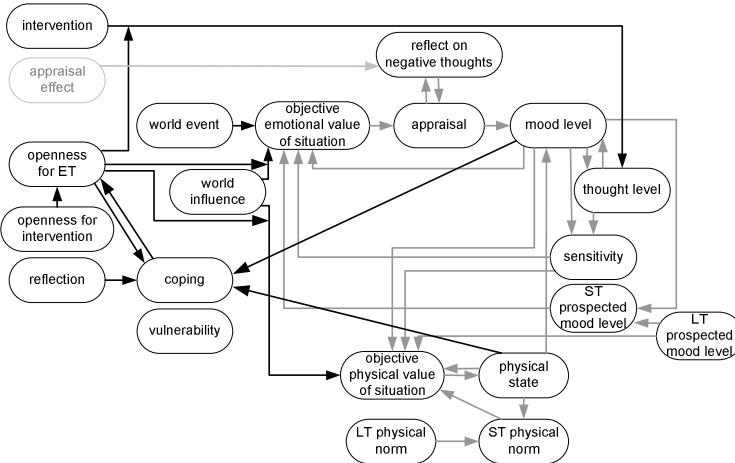


Fig. 2. Computational model for Exercise Therapy

4 Simulation Results

The computational model as described in Section 2 combined with Exercise Therapy (as presented in Section 3) is implemented in Matlab in order to run a number of simulations. For the simulations, three fictional persons are studied which are all susceptible for depression (low coping skills and high vulnerability) and a high openness for intervention. These persons are referred to as person 1, 2, and 3 respectively. The *LT physical norm* is set to low, medium, and high for person 1, 2, and 3 respectively. The values are chosen to show the effect of different physical norms on the mood level and on the influences of the therapy. In all traces, a world event with a very low value occurs at the beginning of the simulation (value 0.1 during 80 hours). For half of the simulations the intervention starts after two weeks of low mood and lasts for ten weeks. In the other half, no intervention is simulated.

Figure 3 shows the simulation trace of person 1 (with a low *LT physical norm*) whereby no intervention is present. Panel a shows the values for *objective emotional value for situation*, *mood* and *ST prospected mood level*; the bottom graph (b) shows the values for *objective physical value for situation*, *physical state* and *ST physical norm*. It can be seen that the person recovers very slowly from the negative event.

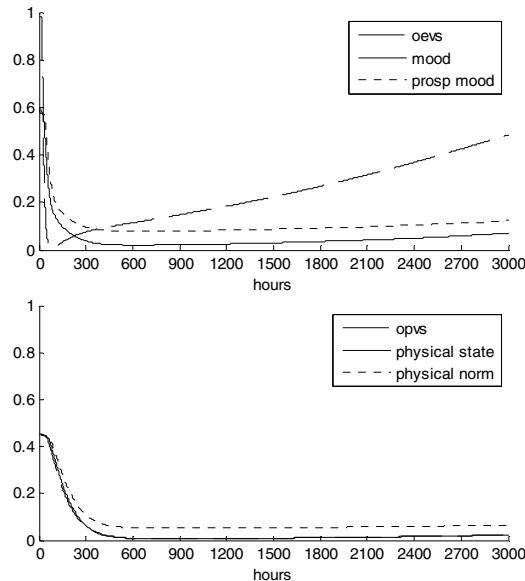


Fig. 3a and b. Simulation trace of person 1 (low value for *LT physical norm*) with no intervention. The top panel (a) shows the mood related values; the bottom panel (b) shows the physical state values.

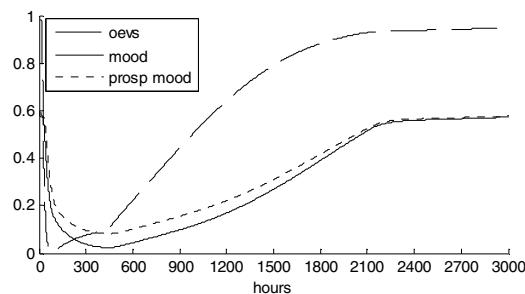


Fig. 4a. Simulation trace of person 1 (low value for *LT physical norm*)

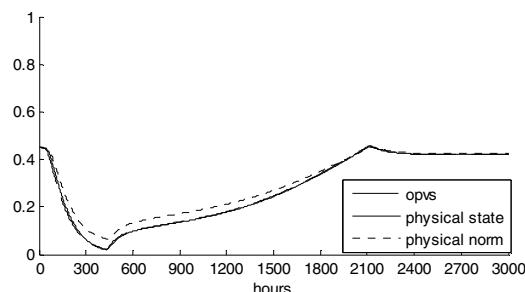


Fig. 4b. Simulation trace of person 1 (low value for *LT physical norm*)

Figure 4 shows the simulation run where the same person does follow the exercise therapy intervention, this results in a more rapid recovery. In Figure 5 person 3 (with a high *LT physical norm*) is shown, following the same intervention is. This results in a faster recovery compared to the person with a low physical norm.

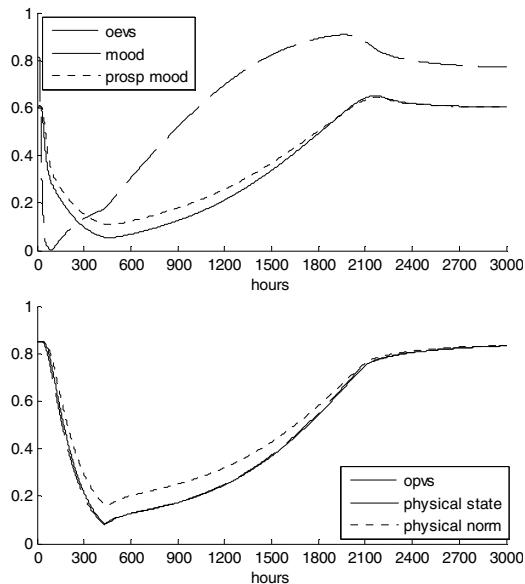


Fig. 5a and b. Simulation trace of person 3 (high value for LT physical norm)

5 Dynamic Properties and Their Verification

A number of temporal properties that reflect a number of general patterns and characteristics of the process of depression and the treatment have been formulated. The properties were specified in the language TTL [5]. This predicate logical temporal language supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. TTL is built on atoms referring to *states* of the world, *time points* and *traces*, i.e. trajectories of states over time. In addition, *dynamic properties* are temporal statements that can be formulated with respect to traces based on the state ontology *Ont* in the following manner. Given a trace γ over state ontology *Ont*, the state in γ at time point t is denoted by $\text{state}(\gamma, t)$. These states can be related to state properties via the infix predicate \models , where $\text{state}(\gamma, t) \models p$ denotes that state property p holds in trace γ at time t . Based on these statements, dynamic properties can be formulated in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as $\neg, \wedge, \vee, \Rightarrow, \forall, \exists$. For more details, see [5]. Automated tool support is also available that allows for dedicated editing and for verifying whether the properties hold in a set of simulation traces. A number of simulations (thereby considering the different types of persons mentioned in Section 4 in combination with Exercise Therapy or no intervention) have been used as basis for the verification. Some of the properties considered are the following.

P1: Effectiveness of Exercise Therapy

Persons that follow exercise therapy are depressed for a shorter period than persons who do not.

$$\forall \gamma_1, \gamma_2: \text{TRACE}, \forall t: \text{TIME}$$

$$[[\text{state}(\gamma_1, t) \models \text{intervention_ET} \& \text{state}(\gamma_2, t) \models \text{not intervention_ET}]$$

$$\Rightarrow \exists t_2: \text{TIME} > t, R_1, R_2 : \text{REAL} [R_1 < \text{MIN_LEVEL} \& R_2 > \text{MIN_LEVEL} \&$$

$$\text{state}(\gamma_2, t_2) \models \text{has_value}(\text{mood}, R_1) \& \text{state}(\gamma_1, t_2) \models \text{has_value}(\text{mood}, R_2)]$$

P2: A higher physical norm results in a shorter depression

Persons that have a higher long term physical norm remain depressed for a shorter period than those with a lower norm.

$$\forall \gamma_1, \forall \gamma_2: \text{TRACE}, \forall R_1, R_2: \text{REAL}, t: \text{TIME}$$

$$[[\text{state}(\gamma_1, t) \models \text{has_value}(\text{phys_norm}, R_1) \& \text{state}(\gamma_2, t) \models \text{has_value}(\text{phys_norm}, R_2) \& R_2 < R_1]$$

$$\Rightarrow \exists t_2: \text{TIME}, R_3, R_4: \text{REAL} [R_3 < \text{MIN_LEVEL} \& R_4 > \text{MIN_LEVEL} \&$$

$$\text{state}(\gamma_2, t_2) \models \text{has_value}(\text{mood}, R_3) \& \text{state}(\gamma_1, t_2) \models \text{has_value}(\text{mood}, R_4)]$$

P3: Effect of exercise therapy on coping skills

After a person has followed exercise therapy for some time, the coping skills have improved.

$$\forall \gamma: \text{TRACE}, t: \text{TIME}, R_1: \text{REAL}$$

$$[[\text{state}(\gamma, t) \models \text{intervention_ET} \& \text{state}(\gamma, t) \models \text{has_value}(\text{coping}, R_1)]$$

$$\Rightarrow \exists t_2: \text{TIME} > t + \text{MIN_DURATION}, R_2: \text{REAL} [R_2 > R_1 + \text{MIN_INCREASE} \& \text{state}(\gamma, t_2) \models \text{has_value}(\text{coping}, R_2)]$$

P4: Effect of exercise therapy on physical state

After a person has followed exercise therapy for some time, his physical state is higher than the physical state of person that did not follow exercise therapy

$$\forall \gamma_1, \gamma_2: \text{TRACE}, \forall R_1, R_2: \text{REAL}, t_1, t_2: \text{TIME}$$

$$[[\text{state}(\gamma_1, t_1) \models \text{intervention_ET} \& \text{state}(\gamma_2, t_1) \models \text{not intervention_ET} \&$$

$$\text{state}(\gamma_1, t_2) \models \text{has_value}(\text{physical_state}, PS_1) \& \text{state}(\gamma_2, t_2) \models \text{has_value}(\text{physical_state}, PS_2) \&$$

$$T_2 > T_1 + \text{MIN_DUR}] \Rightarrow PS_1 > PS_2]$$

The first three properties were shown to be satisfied for the simulation traces as described in Section 4. The last one, however, did not satisfy. This can be explained by the fact that the physical norm has a large influence on the physical state in the long run. In other words, a person with a high physical norm can have a higher physical state without following exercise therapy, than a person that did follow exercise therapy but has a lower long term physical norm. For persons with the same level of physical norm, the property does hold.

6 Discussion

The computational model for the role of physical activity in mood regulation and depression introduced here shows a number of encouraging results. It shows how for certain types of persons a depression can develop when physical activity is low or absent. Moreover, it shows that by exploiting Exercise Therapy in order to increase physical activity the depression can disappear. Both types of effects show a gradual, pattern over time. This relates to an interesting discussion in the literature on how positive effects of physical activity can be explained, for example in [25]:

'Broadly, there are two possible types of explanation. One is that emotional benefits arise from the accumulation of acute mood improvement caused by the individual sessions of exercise. Accumulation of acute effects has been suggested by mainly anecdotal, single-case, or uncontrolled reports that have suggested that mood deteriorates rapidly when exercise regimes are interrupted (...). However, a theory based entirely on

acute emotional effects is implausible because, as was argued above, exercise is likely to be aversive to many people, particularly at the start of training.' ([25], p. 42-43).

'The alternative to attributing the stress-reducing effects of exercise to the accumulation of acute effects is to suppose that a long-term process is recruited. One way to distinguish short-term from long-term effects is to study the effects of interruption of regular exercise. Whereas an acute effect should dissipate rapidly, a long-term effect would be expected to persist.' ([25], p.47)

In the model the choice has been made to define a direct impact of the physical state on mood, and not of the physical activity. This physical state accumulates prior physical activity. The result is indeed a longer term effect with some persistence. In this sense the model fulfills the requirements suggested in [25] as alternative explanation. It would also be not difficult to include as well direct effects in the model from physical activity or physical state to sensitivity and/or to coping and vulnerability. However, the fact that the dynamics of the current model already display the longer term effects as indicated in [25] suggests that this is not required.

Acknowledgments. This research has been conducted as part of the FP7 ICT program of the European Commission under grant agreement No 248778 (ICT4Depression). Furthermore, the authors would like to thank Pim Cuijpers within the Department of Clinical Psychology at the VU University Amsterdam for the fruitful discussions.

References

1. Anand, A., Li, Y., Wang, Y., Wu, J., Gao, S., Bukhari, L., Mathews, V.P., Kalnin, A., Lowe, M.J.: Activity and connectivity of brain mood regulating circuit in depression: A functional magnetic resonance study. *Biological Psychiatry* 57, 1079–1088 (2005)
2. Beauregard, M., Paquette, V., Levesque, J.: Dysfunction in the neural circuitry of emotional self regulation in major depressive disorder. *Learning and Memory* 17, 843–846 (2006)
3. Beck, A.T.: Depression: Causes and Treatment. University of Pennsylvania Press, Philadelphia (1972)
4. Biddle, S.J.H.: Emotion, mood and physical activity. Physical activity and psychological well-being. Routledge, London (2000)
5. Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., Treur, J.: Specification and Verification of Dynamics in Agent Models. *International Journal of Cooperative Information Systems* 18, 167–193 (2009)
6. Both, F., Hoogendoorn, M., Klein, M.A., Treur, J.: Formalizing Dynamics of Mood and Depression. In: Ghallab, M., Spyropoulos, C.D., Fakotakis, N., Avouris, N. (eds.) Proc. of the 18th European Conf. on Art. Int., ECAI 2008,, pp. 266–270. IOS Press, Amsterdam (2008)
7. Both, F., Hoogendoorn, M., Klein, M.C.A., Treur, J.: Computational Modeling and Analysis of Therapeutic Interventions for Depression. In: Yao, Y., Sun, R., Poggio, T., Liu, J., Zhong, N., Huang, J. (eds.) BI 2010 LNCS (LNBI), vol. 6334, pp. 274–287. Springer, Heidelberg (2010)
8. Cotman, C.W., Berchtold, N.C.: Exercise: a behavioral intervention to enhance brain health and plasticity. *Trends Neurosci.* 25, 295–301 (2002)

9. Davidson, R.J., Lewis, D.A., Alloy, L.B., Amaral, D.G., et al.: Neural and behavioral substrates of mood and mood regulation. *Bio. Psychiatry* 52, 478–502 (2002)
10. Daley, A.: Exercise and depression: A review of reviews. *Journal of Clinical Psychology in Medical Settings* 15, 140–147 (2008)
11. Dishman, R.K., Berthoud, H.R., Booth, F.W., et al.: Neurobiology of Exercise. *Obesity* 14, 345–356 (2006)
12. Duclos, M., Gouarne, C., Bonnemaison, D.: Acute and chronic effects of exercise on tissue sensitivity to glucocorticoids. *Journal of Applied Physiology* 94, 869–875 (2003)
13. Dunn, A.L., Trivedi, M.H., Kampert, J.B., Clark, C.G., Chambliss, H.O.: Exercise treatment for depression: efficacy and dose response. *Am. J. Prev. Med.* 28, 1–8 (2005)
14. Goodwin, R.C.: Association between physical activity and mental disorders among adults in the United States. *Preventative Medicine* 36, 698–703 (2003)
15. Greenwood, B.N., Foley, T.E., Day, H.E.W., et al.: Freewheel running prevents learned helplessness/behavioral depression: role of dorsal raphe serotonergic neurons. *J. Neurosci.* 23, 2889–2898 (2003)
16. Greenwood, B.N., Foley, T.E., Burhans, D.J., Maier, S.F., Fleshner, M.: The consequences of uncontrollable stress are sensitive to the duration of prior wheel running. *Brain Res.* 1033, 164–178 (2005)
17. Greenwood, B.N., Strong, P.V., Foley, T.E., Fleshner, M.: A behavioral analysis of the impact of voluntary physical activity on hippocampus-dependent contextual conditioning. *Hippocampus* 19, 988–1001 (2009)
18. Greenwood, B.N., Fleshner, M.: Exercise, learned helplessness, and the stress-resistant brain. *Neuromol. Med.* 10, 81–98 (2008)
19. Harrison, P.J.: The neuropathology of primary mood disorder. *Brain* 125, 1428–1449 (2002)
20. Lawlor, D.A., Hopker, S.W.: The effectiveness of exercise as an intervention in the management of depression: systematic review and meta-regression analysis of randomised controlled trials. *BMJ* 322, 763–767 (2001)
21. Lévesque, J., Eugene, F., Joanette, Y., et al.: Neural circuitry underlying voluntary suppression of sadness. *Biological Psychiatry* 53, 502–510 (2003)
22. Lewinsohn, P.M., Youngren, M.A., Grosscup, S.J.: Reinforcement and depression. In: Dupue, R.A. (ed.) *The psychobiology of depressive disorders: Implications for the effects of stress*, pp. 291–316. Academic Press, New York (1979)
23. Mathers, C.D., Loncar, D.: Projections of global mortality and burden of disease from 2002 to 2030. *PLoS Med.* 3, e442 (2006)
24. Mayberg, H.S.: Modulating dysfunctional limbic-cortical circuits in depression: towards development of brain-based algorithms for diagnosis and optimized treatment. *British Medical Bulletin* 65, 193–207 (2003)
25. Salmon, P.: Effects of physical exercise on anxiety, depression, and sensitivity to stress: A unifying theory. *Clin. Psychol. Rev.* 21, 33–61 (2001)

Representation of Hypertext Documents Based on Terms, Links and Text Compressibility

Julian Szymański¹ and Włodzisław Duch^{2,3}

¹ Department of Computer Systems Architecture, Gdańsk University of Technology, Poland
julian.szymanski@eti.pg.gda.pl

² Department of Informatics, Nicolaus Copernicus University, Toruń, Poland

³ School of Computer Engineering, Nanyang Technological University, Singapore
Wduch@is.umk.pl

Abstract. Three methods for representation of hypertext based on links, terms and text compressibility have been compared to check their usefulness in document classification. Documents for classification have been selected from the Wikipedia articles taken from five distinct categories. For each representation dimensionality reduction by Principal Component Analysis has been performed, providing rough visual presentation of the data. Compression-based feature space representation needed about 5 times less PCA vectors than the term or link-based representations to reach 90% cumulative variance, giving comparable results of classification by Support Vector Machines.

1 Introduction

WWW can be seen as a very large repository of documents that changes in time and constantly grows. The challenge is to organize Internet documents automatically. Categorization (supervised or unsupervised) strongly depends on the methods used to represent text and for many hypertext documents not only the words, but also the links between the documents have been found useful to determine a text category. The best example of such organization is given by Wikipedia, which is ideal for testing link and term-based methods of text representation. Successful validation of information retrieval algorithms on the Wikipedia articles should lead to improvements of information retrieval in the Internet, for example by assigning information to categories found in the Wikipedia. Although the current manually-made system of Wikipedia categories is not perfect it can be used for evaluation of methods based on various text representations. An important advantage of Wikipedia comes from the fact that the data is available for download as semi-structured SQL files and XML dumps^{1,2}. The experiments presented in this article have been performed on the Wikipedia in simple English version³, reducing the data to the most popular articles only.

In recent years significant progress in machine learning methods brought a wide spectrum of techniques for data analysis, especially clustering and classification. These

¹ http://en.wikipedia.org/wiki/Wikipedia_database

² <http://download.wikimedia.org/>

³ http://simple.wikipedia.org/wiki/Main_Page

algorithms represent objects (such as documents) using feature vectors, or creating kernel features that are based on similarity between the objects. Even the best machine learning algorithms without appropriate representation of objects will fail. The aim of the experiments presented here is to find hypertext representation suitable for automatic categorization. Three methods of text representation have been studied: bag-of-words based on terms, the use of links between documents, and estimation of similarity between documents based on their compressibility. Cumulative percentage of variance captured by the most important PCA components [1] already tells us a lot about the quality of representation, and also allow to make rough view of the data in 2D. SVM classification has been performed in the three feature spaces before and after PCA reduction.

2 Text Representation

Humans understand text using a lot of background knowledge; spreading activation processes in the brain invoke additional concepts through automatic (usually shallow) inferences. This process may be partially captured in simple algorithms provided with the help of large ontologies or semantic networks [2]. Here three approaches to text representation that do not use *a priori* knowledge are presented.

In Information Retrieval text is typically represented by the so-called **Bag of Words** (BoW), using frequencies of words as features. The lack of word order and simple grammatical constructions is a severe limitation of such representation. There are several methods that try to deal with that problem. First, features may include collocations and frequent phrases. Second, features may be constructed from statistical analysis of co-occurrence of successive words using *n*-grams [3]. Disadvantage of *n*-grams approach is that it produces very high dimensional feature spaces and requires large training sets. Dimensionality reduction based on PCA may automatically discover some phrases important for document categorization. The Latent Semantic Analysis (LSA) [4] and newer spectral methods work in such reduced spaces, automatically discovering useful combinations of words that contributes to document categorization.

The words that appear in a text have different inflections and require pre-processing to avoid redundant features. Stemming maps words that have the same root (stem) but different inflections on their basic forms (ex: living, lives → live). Words that appear frequently in all texts are removed using stop-words list⁴.

2.1 Terms

The preprocessed words are called terms and in the BoW text representation they are used as features. The value, or descriptiveness of a term for a given document may be estimated by the strength w of association between the term and the text. Typically for n -th term and k -th document w value is calculated as a product of two factors: term frequency tf and inverse document frequency idf , given by $w_{k,n} = tf_{k,n} \cdot idf_n$. The term frequency is computed as the number of its occurrences in the document and is divided by the total number of terms in the document. The frequency of a term in

⁴ <http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words/>

a text determines its importance for document content description. If a term appears in the document frequently, it is considered as more important. The inverse document frequency increase the weight of terms that occur in a small number of documents. The idf_n factor describes the importance of the term for distinguishing documents from each other and is defined as $idf_n = \log(k/k_{term(n)})$, where k is the total number of documents, and $k_{term(n)}$ denotes the number of documents that contain term n .

Features (terms) and weights w that associate them with the collection of documents allows to represent each document by a single point in the Vector Space Model (VSM) [5]. Document similarity is then easily computed using different distance measures such as cosine or euclidean measures [6].

2.2 Links

Representations of texts based on terms lead to high-dimensional feature spaces (compare the size of feature spaces in Table 2). Without preprocessing the number of features would be equal to the total number of the distinct words that appear in all documents. Another more compact way to create numerical representation of texts for evaluation of document similarity is based on references that appear between documents. For articles and books the list of references and bibliographical notes about their authors contain useful information. If hypertext documents are considered their hyperlinks can be used as additional features. This is particularly useful in Wikipedia and in scientific articles, where the number of references is relatively large.

Feature space based on links and shared references may be constructed in several ways. Each link provides a new dimension and the simplest document representation creates a binary vector, where 1 denotes the presence of the link (reference) to another document, and 0 means that there is no link. Documents on similar topics tend to link to similar set of other documents and cite the same references. Possible extensions of this representation involve frequency of references, various forms of weighting, the use of directed links (± 1 for links from or to the document) and personal names that serve as links. These modifications haven't been considered here, only binary representations of articles have been used below.

2.3 Compression

The third approach to the representation of text documents is based on algorithmic information [7]. If two documents are similar their concatenation will not lead to a significant increase of algorithmic complexity. The measure of algorithmic information contained in the text may be estimated using standard file compression techniques. If two text files are quite different compressed concatenated file will have the size approximately equal to the sum of sizes of the two files compressed separately. If the two files are similar compressed concatenated file will be only slightly larger than the size of a single compressed file. To express the complexity-based similarity measure as a fraction by which the sum of the separately compressed files exceeds the size of the jointly compressed file the following formula is used:

$$sim_{A,B} = 2 \left(1 - \frac{size(A+B)_p}{size(A)_p + size(B)_p} \right) \quad (1)$$

where A and B denote text files, and the suffix p denotes the compression operation. This is a good measure of similarity that implicitly takes into account strings of letters, collocations and longer phrases that are used to form a dictionary by the compression algorithm. Each document D is thus represented by a vector with components $V(D)_i = sim_{D,D_i}$, therefore the dimensionality is equal to the total number of documents. Pre-processing in this case is restricted to stop list only, as most compression algorithms can handle word morphology themselves. The book on Kolmogorov algorithmic complexity [7] shows many applications of similarity based on such measures.

3 The Data - Evaluation Dataset

The three ways to generate numerical representation of texts have been compared on a set of articles selected from the Wikipedia. These articles belong to five different subcategories of the Wikipedia supercategory "Science" \hookrightarrow^5 : Chemistry \hookrightarrow **Chemical compounds**, Biology \hookrightarrow **Trees**, Mathematics \hookrightarrow **Algebra**, Computer science \hookrightarrow MS (Microsoft) **operating systems**, Geology \hookrightarrow **Volcanology**. Detailed information about selected documents is presented in Table 1 and Table 2. A total of 281 articles has been selected. For term-based representation only those terms that appeared in the whole collection of articles more than once (freq. > 1) have been kept. Also for the link-based representation references (features) that appear only one time have been removed. Table 1 explains colors and symbols used in Figure 1 to mark particular classes.

Table 1. Category names and the number of **Table 2.** Size of feature spaces for different articles used to construct data sets representation methods

Category name	Number of the articles	Color and Symbol
Chemical compounds	115	red *
Trees	69	green +
Algebra	21	blue □
MS operating systems	19	black ·
Volcanology	57	magenta ◇

Features space size				
terms		links		complexity
raw data	freq. > 1	raw data	freq. > 1	
12358	3658	1817	650	281

4 Comparison of Text Representations

The rough view of the class distribution in different representations can be made using two principal components with the highest variance [1]. This is shown in Figure 1. It is clear that two PCA components are not sufficient to separate all data, although most documents from the "tree" category may be distinguished quite easily. This is also clear from analysis of eigenvalues showing that the first two eigenvectors capture only a small percentage of variance.

⁵ \hookrightarrow denotes hierarchical relation.

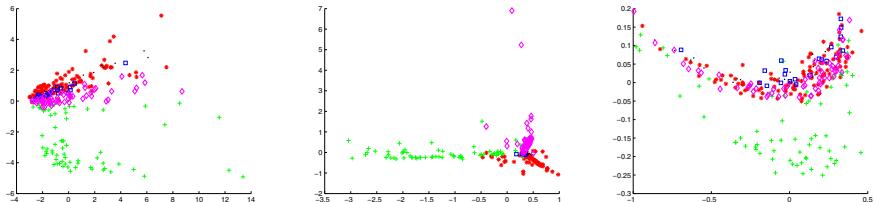


Fig. 1. Projection of dataset on two highest principal components for text representation based on terms, links and compression

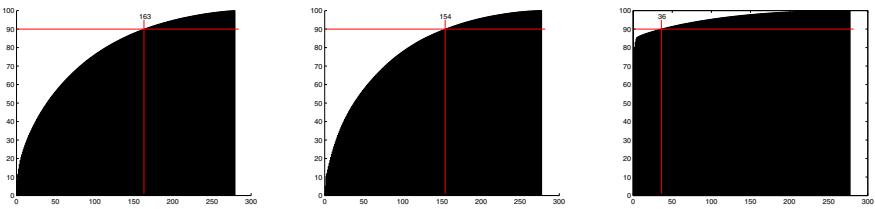


Fig. 2. Cumulative sum of primary components variance for different text representations: terms, links and algorithmic complexity

Using other combinations of principal components for scatterograms or using multidimensional scaling more structure can be observed, indicating that different methods of representation extract different information from texts. To see how much information is lost by performing PCA dimensionality reduction, in Figure 2 cumulative sums of the percentage of variance captured by the most important components for different representation methods is presented. For terms and links more than 100 components are needed to capture 80% of variance, but for algorithmic complexity very few components are needed, and 90% of variance is accounted for using only 36 components, about 5 times less than for terms or links (Fig. 2 where 163 and 154 components were needed).

Information extracted by different text representations may be estimated by comparing classifier errors in various feature spaces. Support Vector Machines classifier [8] have proved to be suitable for text categorization [9], [10]. To perform multiclass classification with SVM one-versus-other class approach has been used with two-fold crossvalidation repeated 50 times for accurate averaging of the results.

The detailed results of calculations are presented in Table 3. Standard deviations in all these calculations did not exceed 3%, and to limit the size of the tables are not reported here. Feature spaces have been generated using term, link and complexity-based document representations, and the results in the first column ("raw") are obtained by using linear SVM directly in these spaces. The second column, marked in Table 3 as $f. > 1$, shows results in reduced feature spaces, after removal of features that appear only with relation to a single document. In almost all cases this leads to improvement,

Table 3. Evaluation of the classification with SVM for different text representations

Category name	Text representation											
	terms				links				complexity			
	raw data	f.>1	cos	euclid	raw data	f.>1	cos	euclid	raw data	cos	euclid	
Chemical compounds	87.2	93.7	98.9	97.2	85.3	86.2	97.7	95.1	91.1	96.8	95.9	
Trees	90.4	92.9	98.7	96.1	87.9	92.1	96.8	95.5	92.7	98.5	95.3	
Algebra	94.1	98.8	99.3	97.8	88.2	91.9	98.7	97.6	95.9	94.9	93.5	
MS operating systems	98.6	98.6	99.9	98.6	97.4	98.6	99.7	99.6	99.3	99.7	98.9	
Volcanology	94.5	94.3	98.8	98.2	94.4	95.7	98.8	96.2	94.5	97.8	95.6	
Overall	92.9	95.6	99.0	97.5	90.6	92.9	98.3	96.8	94.7	97.5	95.8	

sometimes quite significant. Dimensionality of the original and reduced spaces is given in Table 2.

The next two columns contain results in the kernel spaces obtained by linear SVM. Instead of the original vectors \mathbf{X} kernel features $z_i(\mathbf{X}) = \{K(\mathbf{X}, \mathbf{X}_i)\}$ are generated using Euclidean distance $K(\mathbf{X}, \mathbf{X}_i) = \|\mathbf{X} - \mathbf{X}_i\|$ and cosine distance $K(\mathbf{X}, \mathbf{X}_i) = \mathbf{X} \cdot \mathbf{X}_i / \|\mathbf{X}\| \|\mathbf{X}_i\|$ as kernels. These kernel spaces have dimension equal to the number of all documents, in our experiments equal to 281, thus much smaller than the original feature spaces. We have used explicit representation of these kernel spaces with linear SVM instead of explicitly kernelized version of SVM because results of both approaches are essentially equivalent [11], but analysis of the discriminant functions is greatly simplified. For large number of documents selection of redundant kernel features by simple filters may reduce dimensionality in a similar manner to selection of support vectors.

Significant improvements of classification accuracy have been obtained in these kernel spaces. For terms and links Euclidean and cosine kernels replace original features by distances to all training data. Although our database is relatively small this is quite beneficial and should lead to even better results for larger sets of documents. Surprisingly, also for representation based on algorithmic complexity clear improvement in accuracy is noted, although this space is already based on similarity estimated using compression techniques. Transforming data in this representation by distance-type kernels amounts to second-order similarity transformation [12].

Table 4. Evaluation of the classification with SVM for different text representations, scaled with PCA

Category name	Text representation scaled with PCA											
	terms			links			complexity					
	PCA=163	cos	euclid	PCA=154	cos	euclid	PCA=36	cos	euclid			
Chemical compounds	77.4	98.8	97.3	75.3	94.4	96.2	94.1	96.7	95.2			
Trees	82.2	98.4	97.6	79.9	88.9	93.3	90.7	97.7	96.6			
Algebra	96.8	99.8	97.8	92.8	97.8	97.6	96.7	96.5	95.4			
MS operating systems	96.9	99.9	96.9	97.1	98.3	97.8	98.8	99.3	98.9			
Volcanology	86.2	98.9	98.7	86.8	97.1	96.8	93.4	98.1	95.6			
Overall	87.9	99.1	97.6	86.3	95.3	96.3	94.7	97.6	96.3			

In the second set of experiments (Table 4) dimensionality of representation space has been reduced even further by taking only the most important PCA components that cover 90% of the variance in the data. For the term and link representation that leads to some loss of accuracy, while the complexity based representation, with quite small (36) number of dimensions has not been degraded at all. Transforming 163 PCA vectors for term representation using cosine or Euclidean kernel recovered all information in this space, giving an insignificant improvement of the results. However, for links PCA reduction leads to decrease of classification accuracy by 3% for cosine kernel.

5 Discussion and Future Plans

Reading or listing to words neural activation in the brain spreads invoking additional concepts that support understanding and categorization of documents [2]. One should not expect perfect categorization without approximation of such processes with the help of extensive background knowledge and at least shallow inferences. However, it is important to know what kind of knowledge is most important and how to create useful features that would capture important information allowing for text categorization. In this paper we have compared three approaches to text representation, based on terms, links and similarity of their algorithmic complexity. Complexity measure allowed for much more compact representation, as seen from the cumulative contribution of principal components in Fig. 2 and achieved best accuracy in PCA-reduced space with only 36 dimensions, Tab. 4. However, after using cosine kernel term based representation is slightly more accurate. Explicit representation of kernel spaces and the use of linear SVM classifier allows to find important reference documents for a given category, as well as identify collocations and phrases that are important for characterization of each category.

Experiments presented here should be treated as a test-bed for large scale application of our methods for text categorization. The selection of Wikipedia articles from very different subcategories of articles in the supercategory "Science" used here for computational experiments made classification tasks perhaps too easy, as is evident from very high accuracy obtained by the two-fold crossvalidation. In future we plan to investigate more complex tasks, requiring hierarchical of classification, with articles more similar to each other. We can expect that for such more complex tasks the differences in usage of the text representations would be even larger, with more significant advantages coming from kernelization of feature spaces. We plan to run experiments on a much larger scale, on the whole Wikipedia, but this requires parallelisation of algorithms to run them on a powerful cluster instead of on single PC. We also plan to run unsupervised methods for clustering Wikipedia articles and provide tools to automatically create categories for this largest repository of human knowledge.

Different methods of text representation may be combined before such kernelization, and although we have not shown it here, combining term, link and complexity-based representations, followed by kernelization and aggregation of features using PCA leads to even better results with quite small feature spaces. Another idea to introduce more background knowledge and capture some semantics is to map articles on activations of a semantic network and then calculate distances between them. WordNet dictionary [13]

may be used for this purpose with word disambiguation techniques [14] that allow to map words to their proper synsets. We have made some research in this direction and the first results are very promising (in preparation). Representation methods based on neurolinguistic inspirations [2] that use natural concept semantics will also be investigated.

Acknowledgment. This work was supported by the Polish Ministry of Higher Education under research grant no. N N519 432 338.

References

1. Jolliffe, I.: Principal component analysis. Springer, Heidelberg (2002)
2. Duch, W., Matyiewicz, P., Pestian, J.: Neurolinguistic approach to natural language processing with applications to medical text analysis. *Neural Networks* 21(10), 1500–1510 (2008)
3. Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. *Science* 267, 843 (1995)
4. Landauer, T., Dumais, S.: A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge. *Psychological Review* 104, 211–240 (1997)
5. Wong, S.K.M., Ziarko, W., Wong, P.N.: Generalized vector spaces model in information retrieval. In: SIGIR 1985, pp. 18–25. ACM Press, New York (1985)
6. Korenius, T., Laurikkala, J., Juhola, M.: On principal component analysis, cosine and Euclidean measures in information retrieval. *Information Sciences* 177, 4893–4905 (2007)
7. Li, M., Vitányi, P.: An Introduction to Kolmogorov Complexity and its Applications, 3rd edn. Springer, Heidelberg (2008)
8. Schölkopf, B., Smola, A.: Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
9. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Machine Learning: ECML 1988, pp. 137–142 (1998)
10. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)* 34, 1–47 (2002)
11. Maszczyk, T., Duch, W.: Support feature machines: Support vectors are not enough. In: World Congress on Computational Intelligence, pp. 3852–3859. IEEE Press, Los Alamitos (2010)
12. Duch, W.: Towards comprehensive foundations of computational intelligence. In: Duch, W., Mandziuk, J. (eds.) Challenges for Computational Intelligence, Springer, Heidelberg (2007)
13. Miller, G.A., Beckitch, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: An On-line Lexical Database. Cognitive Science Laboratory, Princeton University Press, Princeton (1993)
14. Voorhees, E.: Using WordNet to disambiguate word senses for text retrieval. In: 16th ACM SIGIR Conference, pp. 171–180 (1993)

A Heuristic-Based Feature Selection Method for Clustering Spam Emails

Jungsuk Song, Masashi Eto, Hyung Chan Kim,
Daisuke Inoue, and Koji Nakao

National Institute of Information and Communications Technology, Tokyo, Japan
`{song,eto,hckim,dai,ko-nakao}@nict.go.jp`

Abstract. In recent years, in order to cope with spam based attacks, there have been many efforts made towards the clustering of spam emails. During the clustering process, many statistical features (*e.g.*, the size of emails) are used for calculating similarities between spam emails. In many cases, however, some of the features may be redundant or contribute little to the clustering process. Feature selection is one of the most typical methods used to identify a subset of key features from an initial set. In this paper, we propose a heuristic-based feature selection method for clustering spam emails. Unlike the existing methods in that they make the combinations of given features and evaluate them using data mining and machine learning techniques, our method focuses on evaluating each feature according to only its value distribution in spam clusters. With our method, we identified 4 significant features which yielded a clustering accuracy of 86.33% with low time complexity.

Keywords: spam, clustering, feature selection, time complexity.

1 Introduction

Most recent spam emails contain URLs that navigate spam receivers to malicious Web servers for the purpose of carrying out various cyber attacks such as malware infection, phishing attacks, etc. In order to cope with these spam based attacks, there have been many efforts made towards the clustering of spam emails based on similarities between them [1,2,3,4]. Since most recent spam emails are being sent by bots which often operate with others in the form of a botnet, it is possible to identify the infrastructure of spam sending systems and how they are grouped with each other by analyzing each spam cluster. Therefore, if we utilize botnet information obtained by clustering spam emails, it can be helpful to defend our computer systems and networks from potential cyber attacks.

On the other hand, the existing research applies many statistical features for calculating similarities between spam emails. For example, in [1,2,3], they used three criteria, *i.e.*, email content (mainly words), URLs and their domain names to calculate similarities between spam emails. In other words, if two emails contain similar contents or URLs or their domain names, then they become members of the same cluster. In many cases, however, some of the features

may be redundant or contribute little to the clustering process. Furthermore, considering we have to deal with a large amount of spam emails, it is very important to minimize the clustering time.

Feature selection is one of the most typical methods used to identify a subset of key features from an initial set. Most of the existing methods make the combinations of given features and evaluate them using clustering techniques, but doing so is extremely time-consuming. In this paper, we propose a heuristic-based feature selection method for clustering spam emails, which focuses on evaluating each feature according to its value distribution in spam clusters. By examining three weeks of spam emails captured in our SMTP server, we identified 4 significant features from the original 12 features which were defined in our previous work [6]. These 4 features yielded a clustering accuracy of 86.33%, and dramatically lowered the overall time needed for the clustering process.

The rest of the paper is organized as follows. In Section 2, we give a brief description of the existing spam clustering methods. In Section 3, we present our heuristic-based feature selection method and provide evaluation results. Finally, we make conclusions and suggestions for future study in Section 4.

2 Related Work

Zhuang et al. proposed a clustering method in that spam emails with similar content were regarded as the same cluster [1]. Li et al. used URLs in spam emails as the criterion for clustering [2]. In [3], Xie et al. applied polymorphic URLs which have the same domain name to grouping spam emails. Since there is a fatal weakness in that the three criteria, *i.e.*, content, URL and domain name, are easily influenced by changes in spam messages and trends, we defined 12 statistical features to calculate similarity between spam emails in [6]. Although we demonstrated that its performance is better than that of the domain name and URL based clustering methods, there is a limitation in that all 12 features are not essential for clustering spam emails. In this paper, we focus on removing redundant features and selecting key features from the original 12 features so that we are able to minimize the time required for clustering spam emails while maintaining a high clustering accuracy.

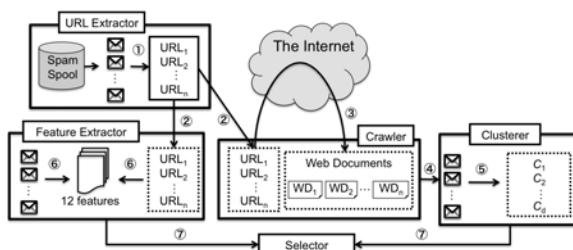


Fig. 1. Overall procedure of the proposed method

3 Proposed Method

3.1 Overall Procedure

Figure 1 shows the overall procedure of the proposed method, and it is composed of the following 5 main parts: URL Extractor, Crawler, Clusterer, Feature Extractor and Selector. The URL Extractor parses the email body, extracts URLs from it (①), and inserts them into the Crawler and the Feature Extractor (②). The Crawler accesses Web sites linked to the extracted URLs, downloads their HTML content (③), and inserts them into the Clusterer (④). The Clusterer generates clusters of spam emails according to similarities found in Web documents downloaded from the Crawler (⑤) as described in Section 3.2. The Feature Extractor extracts 12 statistical features from spam emails and the extracted URLs (⑥), which were defined in our previous work [6]. Finally, the Selector selects significant features from the 12 statistical features using the results of the Clusterer and the Feature Extractor (⑦).

3.2 Generating Clusters

Similar to the existing research [13, 5], we clustered spam emails according to similarities found in Web pages linked to URLs: if two spam emails share at least one URL whose Web content is quite similar to each other, then they become members of the same cluster. In order to examine the similarity between Web pages, we applied text shingling technique [3, 4, 7] to the corresponding Web pages. For example, assume that there are two Web documents, *e.g.*, A and B, which contain 30 (A_1, A_2, \dots, A_{30}) and 20 (B_1, B_2, \dots, B_{20}) words, respectively. If the size of a shingle is 5, then 26 shingles (*e.g.*, $A_1A_2A_3A_4A_5, A_2A_3A_4A_5A_6, \dots, A_{26}A_{27}A_{28}A_{29}A_{30}$) and 16 shingles (*e.g.*, $B_1B_2B_3B_4B_5, B_2B_3B_4B_5B_6, \dots, B_{16}B_{17}B_{18}B_{19}B_{20}$) are extracted from A and B, respectively. In this case, similarity between A and B is calculated as follows.

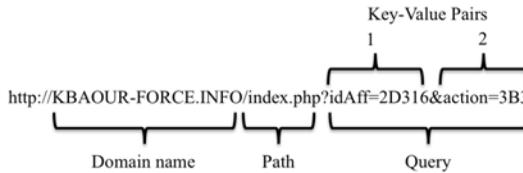
$$\frac{\text{the number of the same shingles}}{\text{the number of unique shingles in both } A \text{ and } B}$$

In this example, if A and B share 15 shingles with each other and the threshold to determine whether they are the same cluster or not is 50%, then we regard them as members of the same cluster, because similarity between A and B is $15/(11 + 15 + 1) = 56\%$ which is larger than 50%.

3.3 Extracting Features

In our previous work [6], we defined 12 statistical features to calculate similarity between spam emails as shown in Table 1. We describe 12 statistical features using the example of an URL as shown in Figure 2.

Assume that an email contains only one URL of Figure 2 in its body part. From the email, the size of emails (*i.e.*, 57) and the number of lines (*i.e.*, 1) are computed. After that, the URL is divided into 3 parts : domain name, path and

**Fig. 2.** Example of an URL**Table 1.** Description of the 12 statistical features

No.	Feature Name	Value
1	Size of emails	57
2	Number of lines	1
3	Number of unique URLs	1
4	Average length of unique URLs	57
5	Average length of domain names	17
6	Average length of path	10
7	Average length of query	22
8	Average number of key-value pairs	2
9	Average length of keys	5.5
10	Average length of values	4
11	Average number of dots(.) in domain names	1
12	Number of global top 100 URLs	0

query, and the query part is also partitioned into each key-value pair, because it can contain multiple key-value pairs. By using those parts, the values of 9 features (*i.e.*, No. 3 ~ No. 11) associated to the URL are computed as shown in Table 1. Finally, the number of global top 100 URLs, which is provided from Alexa.com, is computed.

3.4 Selecting Significant Features and Evaluation

In this section, we present our feature selection method based on heuristic analysis of spam emails. In our method, we first captured double bounce emails from our SMTP server (see subsection 3.4.1), and obtained 772 clusters using the clustering method in Section 3.2. After that, we selected the top 10 clusters from the original 772 clusters as shown in Table 2 (see subsection 3.4.2). Using the top 10 clusters, we evaluated 12 features heuristically and identified their degrees of contribution for improving the clustering accuracy of spam emails (see subsection 3.4.3).

3.4.1 Experimental Data

In our method, we used double bounce emails that were obtained by our own SMTP server. Double bounce emails indicate that they have no valid email addresses associated with spam senders and receivers in their header. This situation

Table 2. Statistics of Top 10 Clusters. A: Num. of emails, B: Num. of unique source IP addresses, C: Num. of unique source countries, D: Num. of entire URLs, E: Num. of unique URLs , F: Num. of unique domain names, G: Num. of unique destination IP addresses, H: Num. of unique destination countries.

ID	Top 10 Clusters									
	1	2	3	4	5	6	7	8	9	10
A	286,024	30,205	22,696	21,061	17,393	10,741	9,807	7,649	5,601	5,265
B	169,149	19,102	16,881	10,480	4,003	4,828	6,101	5,241	4,958	3,061
C	192	177	163	162	101	120	120	127	133	137
D	951,565	30,209	77,794	21,061	17,397	18,363	9,806	7,649	5,919	5,254
E	891,795	177	68,584	94	16,790	6,306	102	67	155	8
F	890,022	157	68,344	79	16,757	6,300	97	63	135	6
G	80	11	164	14	100	21	3	3	13	3
H	11	6	29	5	30	4	2	3	4	3

could be said that spammers intentionally forged his/her return-path address to conceal his/her activities and randomly generated recipient addresses in which some of them exist in the real world and others do not exist (*i.e.*, double bounce emails). Furthermore, a normal email has at least one valid return-path address in its email header, even if a sender mistyped the recipient address to his/her email. In this context, double bounce emails can be regarded as pure spam, and therefore we use double bounce emails for our analysis data, *i.e.*, spam emails.

We collected 596,526 double bounce emails that arrived at our SMTP server for three weeks (Jan. 25th – Feb. 20th, 2010). Among all emails, we observed that 526,544 emails contained one or more URLs in their body, and the total number of URLs were 1,405,950 of which downloadable URLs numbered 1,296,120.

3.4.2 Top 10 Clusters

Using the Clusterer in Section 3.2, we obtained 772 clusters from the above experimental data. During the clustering process, we set the size of a shingle to 5 and the threshold to 50%. This means that a shingle consists of 5 adjacent words, and if similarity between Web documents is larger than 50%, then they become members of the same cluster. From 772 clusters, we observed that most clusters contain only a few emails. In fact, 328 clusters contained only 1 email, 270 clusters contained 10 or less emails, and 168 clusters contained 10,000 or less.

After further investigation, we observed that there are 10 large clusters which are responsible for about 90% of spam emails with URLs arriving at our SMTP server. Table 2 shows their statistical information. From Table 2, we can see that each cluster has a lot of distinct spam sending systems, *i.e.*, 3,061 ~ 169,149, which represent the size of botnets or systems connected closely with each other and that they are distributed in many different countries, *i.e.*, 101 ~ 192. While in the case of URL destinations, we can see that the number of unique IP addresses is only 3 ~ 164 and that they are located in 2 ~ 29 countries. Also, we can observe that there are two patterns of URLs and domain names: in the case

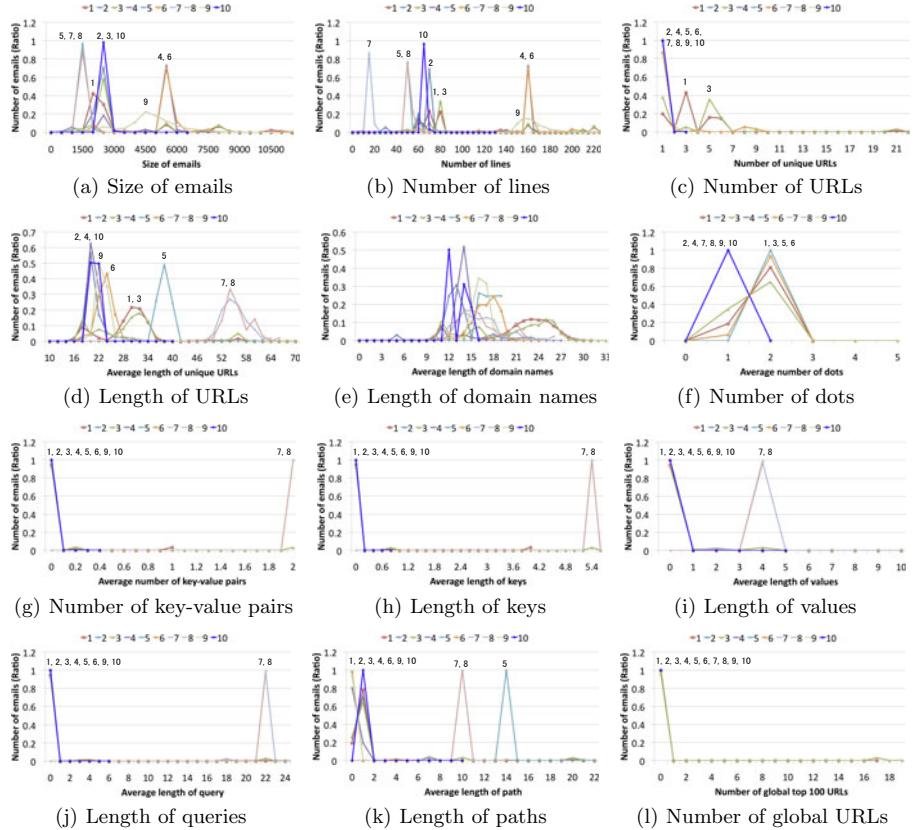


Fig. 3. Value distribution of 12 statistical features with respect to top 10 clusters

of clusters 1, 3 and 5, spammers or attackers frequently change their URLs and domain names, while in the case of the others, they almost always used the same URLs and domain names.

3.4.3 Selecting Significant Features

In order to discover a set of optimized features, it is best to investigate all the combinations of the 12 features: namely, the number of all cases to be estimated

Table 3. Comparison of the clustering accuracy and time complexity

	4 significant features	all features
Clustering Accuracy	86.33%	86.63%
Execution Time (sec.)	6,124	28,772

is $\sum_{i=1}^{12} C_i$, but doing so is extremely time-consuming. Thus, we carried out an alternative experiment, which is based on the following heuristic analysis of the top 10 clusters.

1. In each cluster, the number of spam emails that have the same value is counted with respect to all of the 12 features.
2. The number of spam emails is scaled to [0, 1] according to the size of each cluster, because the size of the top 10 clusters is different from each other.
3. Made 2-dimensional graphs for each feature where the horizontal axis indicates the values of each feature and the vertical axis indicates the ratio of the number of spam emails within each cluster as shown in Figure 3.
4. Using the results in Figure 3, we first selected three significant features: “Size of emails” (3(a)), “Number of lines” (3(b)), “Length of URLs” (3(d)), because most clusters can be distinguished from each other by using their values. In other words, since spam emails in each cluster have different value distribution in these three features, if we use the three features to calculate similarity between spam emails it is possible to accurately partition them into clusters where members within the same cluster have similar values with each other.
5. Second, we excluded “Number of global URLs” (3(l)) from the list of significant features, because the top 10 clusters have similar values in this feature.
6. Third, we did not regard “Length of domain names” (3(e)) as a significant feature, because it is unable to help distinguish each cluster at all: there is no cluster whose value distribution is independent from others.
7. Fourth, we also excluded 6 features, i.e., “Number of URLs” (3(c)), “Number of key-value pairs” (3(g)), “Length of keys” (3(h)), “Length of values” (3(i)), “Length of queries” (3(j)), “Length of paths” (3(k)) from the list of significant features, because they helped distinguish clusters 7, 8, 1, 3 and 5 from the others, but it is obvious that those clusters can also be obtained by using the three significant features in step 4.
8. Finally, since it is impossible to distinguish cluster 4 from cluster 5 using the three significant features in step 4, we added “Number of dots” (3(f)) to the list of significant features as it was able to separate them.

As a result, we selected four significant features, i.e., “Size of emails”, “Number of lines”, “Length of URLs” and “Number of dots”. Note that our method is a supervised feature selection, since it uses the label information of top 10 clusters.

3.4.4 Performance Evaluation

In order to demonstrate the effectiveness of the four significant features, we examined their clustering accuracy and time complexity. As a clustering algorithm, we used an optimized spam clustering method, called *O-means*[6], based on the K-means clustering method, which is one of the most widely used clustering methods. The evaluation results are shown in Table 3. From Table 3, we can see that almost the same clustering accuracy (i.e., 86.33%) was yielded by using only these four significant features. In addition, we can see that execution time was drastically reduced as a result of using only these four significant features,

enabling us to analyze spam emails more effectively. This measurement was performed on a machine running an Intel Core 2 Duo 2.8GHz CPU with 3 GB of RAM, and our program was written in the Perl programming language and Mysql.

4 Conclusion

In this paper, we have proposed a heuristic-based feature selection method for clustering spam emails. By examining three weeks of spam emails gathered in our SMTP server, we have identified 4 significant features which yielded a clustering accuracy of 86.33%. This is comparable with the clustering accuracy (*i.e.*, 86.63%) of our previous work. Furthermore, using these four features, the clustering time was drastically lowered from 28,772 seconds to 6,124 seconds, enabling us to analyze spam emails more effectively.

In our future work, it is a challenging task to evaluate the 12 features with respect to all their combinations using many other clustering methods, so that we are able to obtain an optimized set of statistical features for the clustering of spam emails. Also, we will evaluate the four significant features with a larger data set of spam obtained from various domain sources.

References

1. Zhuang, L., Dunagan, J., Simon, D.R., Wang, H.J., Tygar, J.D.: Characterizing botnets from email spam records. In: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, San Francisco, vol. (2), pp. 1–9 (2008)
2. Li, F., Hsieh, M.H.: An Empirical Study of Clustering Behavior of Spammers and Group-based Anti-Spam Strategies. In: Proceedings of 3rd Conference on Email and Anti-Spam (CEAS), Mountain View, CA, pp. 21–28 (2006)
3. Xie, Y., Yu, F., Achan, K., Panigrahy, R., Hulten, G., Osipkov, I.: Spamming botnets: signatures and characteristics. ACM SIGCOMM Computer Communication Review 38(4) (October 2008)
4. Song, J., Inoue, D., Eto, M., Kim, H., Nakao, K.: An Empirical Study of Spam: Analyzing Spam Sending Systems and Malicious Web Servers. In: 10th Annual International Symposium on Applications and the Internet (SAINT 2010), Seoul, Korea, pp. 19–23 (July 2010)
5. Anderson, D.S., Fleizach, C., Savage, S., Voelker, G.M.: Spamsifter: Characterizing Internet Scam Hosting Infrastructure. In: Proceedings of the USENIX Security Symposium, Boston (2007)
6. Song, J., Inoue, D., Eto, M., Kim, H., Nakao, K.: O-means: An Optimized Clustering Method for Analyzing Spam Based Attacks. IEICE Transactions on Fundamentals E94-A(1) (January 2010)
7. Fetterly, D., Manasse, M., Najork, M., Wiener, J.L.: A large-scale study of the evolution of web pages. Softw. Pract. Exper. 34(2) (2004)

Enhancement of Subjective Logic for Semantic Document Analysis Using Hierarchical Document Signature

Sukanya Manna, Tom Gedeon, and B. Sumudu U. Mendis

School of Computer Science, The Australian National University,
Canberra, ACT 0200, Australia
{sukanya.manna,tom.gedeon,sumudu.mendis}@anu.edu.au

Abstract. In this paper, an extension of Subjective Logic (SL) is presented which uses semantic information from a document to find ‘opinions’ about a sentence. This method computes semantic overlap of events (words or sentences) using Hierarchical Document Signature (HDS) and uses it as evidence to formulate SL belief measures to order sentences according to their importance. Stronger the opinion, more is the significance. These significant sentences then form extractive summaries of the document. The experimental results show that summaries generated by this method are more similar to human generated ones have outperformed the baseline summaries on average over all the data sets considered.

Keywords: subjective logic, opinions, evidence, Hierarchical Document Signature, semantic, summarization.

1 Introduction

Subjective logic is a type of probabilistic logic that explicitly takes uncertainty and belief ownership into account. In general, subjective logic is suitable for modeling and analysing situations involving uncertainty and incomplete knowledge. In subjective logic (SL) [4], first order measure of evidence is expressed as belief mass distribution functions over frame of discernment. All these belief measure representations in subjective logic, which are called ‘opinions’, also contain a base rate parameter which express the a priori belief in the absence of evidence. Philosophically, opinions are quantitative representations of evidence as perceived by humans or by other intelligent agents [5].

In document computing context, SL is used by Manna et al. [15], to find ‘opinion’ about a sentence where a sentence is regarded as a proposition; considering the fact that determining importance of a sentence from a document is very subjective and depends on a reader’s interest and motivation. The approach discussed in [15] is based on ‘bag of words’ concept where events are represented by words and their co-occurrences. This is enhanced in [14] where a theoretical framework is proposed using the semantic information.

Aim of this paper is to look at the practical application of the theoretical framework presented in [14] for extracting significant sentences from a document using the underlying semantic information of the context. Here, semantic overlap or similarity of concepts or events are computed using Hierarchical Document Signature (HDS) [12] and Fuzzy Word Similarity (FWS) measure [13] over simple ‘bag of words’ as in [15].

2 Measure of Semantic Similarity

Two words are contextually similar, if they share similar senses. To perform this automatically, an online lexicon database, WordNet [17] is required to compute this measure. Each word can have one or more synsets based on different senses of their existence also in different parts of speech like noun, verb, adjective, and adverb. Same word in different parts of speech convey different meaning to the context in which they are used. The word ‘hit’ when is a *verb*, then it means collision; again when it is a *noun*, then it means something popular as in cases of ‘movie hit’ and so on. Only considering root form of any word misses out the semantic meaning of it. To overcome this problem of ‘bag of words’ concept, a similarity measure α is introduced in this paper for finding semantic similarity of concepts.

Definition 1 (Semantic Similarity). Let Θ be a frame of discernment, and let $x, y \in 2^\Theta$. Then for any x and y , semantic similarity is the function $\alpha : \text{SimScore}(x, y) \rightarrow [0, 1]$ defined by

$$\alpha(x, y) = \text{SimScore}(x, y) \quad x, y \in 2^\Theta. \quad (1)$$

where $\text{SimScore}(x, y)$ is a function which determines the semantic similarity measure between x and y provided the elements of x and y are in the same parts of speech. This can be any kind of similarity score like gloss overlap [6], path based measures [11], [18], edge based measures, [13] or sentence similarity measure [12]. A threshold κ is used to define the degree of similarity. Thus, $\kappa = 1$, x and y are identical

$\psi \leq \kappa < 1$, x and y are similar

$0 \leq \kappa < \psi$, x and y are dissimilar, where $\psi \in (0, 1)$.

Generally $\kappa = 0.5$ is taken as a standard value for similarity scores [2]. But in this paper, similarity of events is computed using HDS and $\kappa = 0.4$ is used throughout due to fuzzy approximations in the model and by thorough testing on thousands of data.

2.1 Similarity Computation Using Hierarchical Document Signature (HDS)

Hierarchical Document Signature (HDS) [12] is a special form of Fuzzy Signature (FS) [16] meant for document analysis. For computing semantic similarity of contexts, HDS is generally used. Its levels are based on the natural hierarchy of

a document and can vary depending on applications. In this case, parts of speech (POS) is a level which deals with the semantic information of the sentences of a document. The aggregations at different level contribute to the final similarity score. It uses Fuzzy Word Similarity (FWS) [13] at the word level to deal with the similarity or relatedness of a word pair; which then propagates to the next higher level using proper aggregations and continues till it reaches the document level. In this application, sentence-sentence similarity, sentence-word similarity are computed using HDS; which can be called as the similarity between events where events can be either atomic or composite.

For this case, the membership functions of the fuzzy sets to compute FWS as used in [12] are further tuned. In [13] it is seen that triangular membership functions performed better than trapezoidal at the output. Here after tuning it is seen that trapezoidal can also be used at the output and it also gives reasonably good results. The new membership functions for the input and output are:

Table 1. Co-ordinates of trapezoidal membership functions of input and output fuzzy sets used here

INPUT low	(0, 1) (0.2, 1) (0.3, 0)
INPUT medium	(0.2, 0) (0.3, 1) (0.4, 1) (0.5, 0)
INPUT high	(0.4, 0) (0.5, 1) (1, 1)
OUTPUT low	(0, 1) (0.1, 1) (0.3, 0)
OUTPUT medium	(0.1, 0) (0.3, 1) (0.5, 1) (0.7, 0)
OUTPUT high	(0.5, 0) (0.7, 1) (1, 1)

Besides the membership functions, similarity threshold is determined to be 0.4 after repeated experiments.

3 Extension of SL Belief Measures Using Semantic Information

In this section, extension of subjective logic formulation for document analysis using semantic information. The equations of [15] are redefined using the similarity score as computed in (1).

Definition 2 (Belief Mass Assignment). Let Θ be a frame of discernment. If with each substate $x \in 2^\Theta$ a number $m_\Theta(x)$ is associated such that:

1. $m_\Theta(x) \geq 0$
2. $m_\Theta(\emptyset) = 0$
3. $\sum_{x \in 2^\Theta} m_\Theta(x) = 1$

then m_Θ is called a belief mass assignment in Θ , or BMA for short. For each substate $x \in 2^\Theta$, the number $m_\Theta(x)$ is called the belief mass of x .

The states/substates are also called events. In this paper, events can be atomic or composite depending on whether they are single words or a sentence (as a sentence is composed of many words).

3.1 BMA Calculation

BMA is explained in def[2]. Now, for a document, BMA for each event is calculated by,

$$m(x) = \frac{F(x)}{Z}, \quad (2)$$

where $F(x) = \sum_{k=1}^N f_{x_k}$, where N is the total number of sentences in the document, $x \in 2^\Theta$, and f_{x_k} is the frequency of occurrence of event x in sentence k . In words, it is the total frequency of that event in all the sentences (or the whole document).

$$Z = \sum_{\substack{\forall x \neq \emptyset \\ f_x \neq 0}} F(x), \quad x \in 2^\Theta \quad (3)$$

Z is the total frequency of the all the existing events whose frequency is non zero. In calculating frequency of atomic states, the POS of words are considered instead of root form as in [15]. For example, suppose the word ‘crash’ is in two different pars of speech (POS) in two different sentences, so these are considered to be two separate atomic events. Likewise, for the word ‘plane’ when it occurs in two sentences in the form of noun then a total count of 2 will be considered for that state.

Definition 3 (Semantic Belief Function). Let Θ be a frame of discernment, m_Θ be a BMA and α be semantic similarity on Θ respectively. Then the belief function corresponding with m_Θ and alpha is the function $b^s : 2^\Theta \rightarrow [0, 1)$ defined by:

$$b^s(x) = \sum_{\forall y | \alpha(x, y) \leq 1} m_\Theta(y), \quad x, y \in 2^\Theta, y \subseteq x \quad (4)$$

Definition 4 (Semantic Disbelief Function). Let Θ be a frame of discernment, m_Θ be a BMA and α be semantic similarity on Θ respectively. Then the disbelief function corresponding with m_Θ and α is the function $d^s : 2^\Theta \rightarrow [0, 1)$ defined by:

$$d^s(x) = \sum_{\alpha(x, y) < \kappa} \alpha(x, y) m_\Theta(y), \quad x, y \in 2^\Theta. \quad (5)$$

Definition 5 (Semantic Uncertainty Function). Let Θ be a frame of discernment, m_Θ be a BMA and α be semantic similarity on Θ respectively. Then the disbelief function corresponding with m_Θ and α is the function $u^s : 2^\Theta \rightarrow [0, 1)$ defined by:

$$u^s(x) = \sum_{1 > \alpha(x, y) \geq \kappa} \alpha(x, y) m_\Theta(y), \quad x, y \in 2^\Theta. \quad (6)$$

In this situation, the **Semantic Belief Function** will no longer hold strict additivity like ([15]) and is thus expressed as:

$$b^s(x) + d^s(x) + u^s(x) \leq 1, \quad x \in 2^\Theta, x \neq \emptyset. \quad (7)$$

Definition 6 (Semantic Relative Atomicity). Let Θ be a frame of discernment, let $x, y \in 2^\Theta$, and let $\alpha(x, y)$ be semantic similarity of x and y . Then for any given $y \neq \emptyset$ the relative atomicity of x to y is the function $a : 2^\Theta \rightarrow [0, 1]$ defined by:

$$a^s(x/y) = \frac{\sum_{j=1}^{|y|} \bigvee_{i=1}^{|x|} \alpha(x_i, y_j)}{|y|}, \quad x, y \in 2^\Theta, \quad \alpha(x_i, y_j) \geq \kappa \quad (8)$$

where x_i and y_i are atomic elements of x and y respectively.

Definition 7 (Semantic Probability Expectation). Let Θ be a frame of discernment with BMA m_Θ then the semantic probability expectation function corresponding with m_Θ is the function $E : 2^\Theta \rightarrow [0, 1]$ defined by:

$$E^s(x) = \sum_y m_\Theta(y) a^s(x/y), \quad x, y \in 2^\Theta. \quad (9)$$

Thus semantic **Opinion** about a proposition can be written as,

$$\omega^s(x) \equiv (b^s(x), d^s(x), u^s(x), a^s(x)). \quad (10)$$

A sentence is considered significant if it has stronger opinion. In other words, if it has greater probability expectation and lower uncertainty, more it is significant in the document.

4 Experiment

This section presents the experiment and evaluation results from evidence based subjective logic model discussed above. The significant sentences are extracted and then they are compared with human generated summaries provided with the dataset. As a benchmark, baseline summaries are also provided, which are compared in the same way with the human assessors and difference of results are noticed.

4.1 Data Set

In this experiment DUC2001 data set [1] is used for evaluation. The documents are grouped based on a specific topic. DUC2001 comes with human generated summaries and baseline summaries, providing a good platform for evaluation.

4.2 Generation of Summaries

Summaries are broadly classified into text extraction and text abstraction [9], [7]. For text extraction, sentences from the documents are used as summaries and for text abstraction important pieces of information are extracted and then stitched together to form summaries following some linguistic rules. This evidence based model can be used as a text extraction as the original sentences are used for summary generation.

Semantic evidence based model (SPEU): Section 3 describes how to compute opinion about a sentence. All the SL belief measures are computed for each sentence. Then the sentences are arranged based on their descending order of semantic Probability Expectation (SPE) and ascending order of uncertainty (U). Then 30% [3] of the top ranked sentences are extracted and formed extractive summary of the document.

4.3 Evaluation by ROUGE

ROUGE [8] stands for Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. In this experiment, the result with ROUGE-1 (n-gram, where n=1) at 95% confidence level are presented. ROUGE is sensitive to the length of the summaries [10]; hence the length is fixed to 100 words for the evaluation.

4.4 Results

DUC2001 dataset is used for this experiment. Among different document sets, the evaluation with ‘daycare’, ‘healthcare’, ‘pres92’ and ‘robert.gates’ data sets are presented. Our method (SPEU) and baseline summaries (denoted by LP) with two different human assessors are compared. For each set the assessors are different. The average table. 2 results show that our method out performs the baseline summaries which is rather good.

Figures 1 to 4 illustrate per document wise comparison for each data set. It is known that human judgements are subjective, whose evidence is seen through the summaries generated by human assessors. For *daycare* data, SPEU performs better than LP. Though in some documents, the results varied slightly.

For *healthcare* data similar results are noticed except for assessor j, where SPEU generated summaries are not as good as baseline summaries.

Again, for figures 3 and 4, SPEU performs better than LP for most of the cases showing higher similarity with the human generated summaries. For some documents, it is seen that SPEU’s performance dropped. The main reason for this is the coverage pitfall of the WordNet used. If a document has lots of unknown

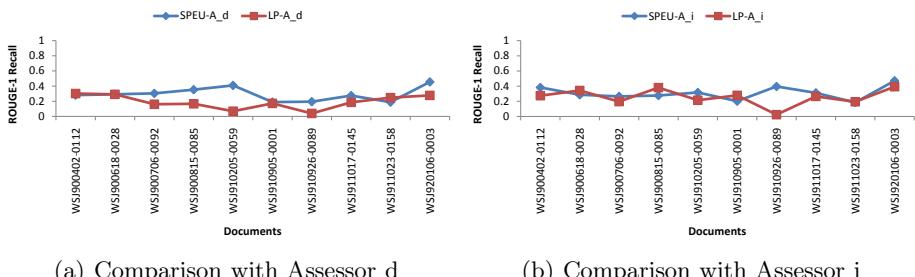
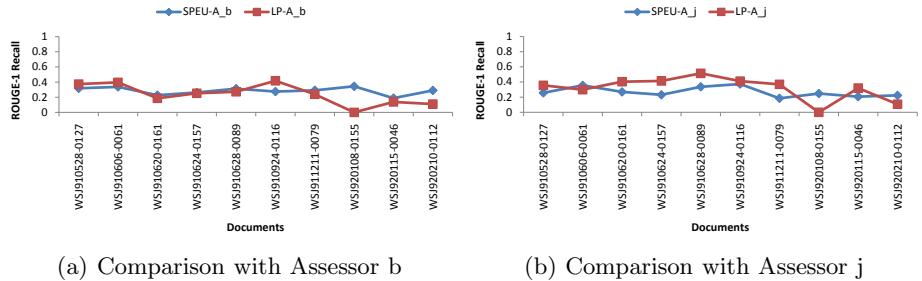
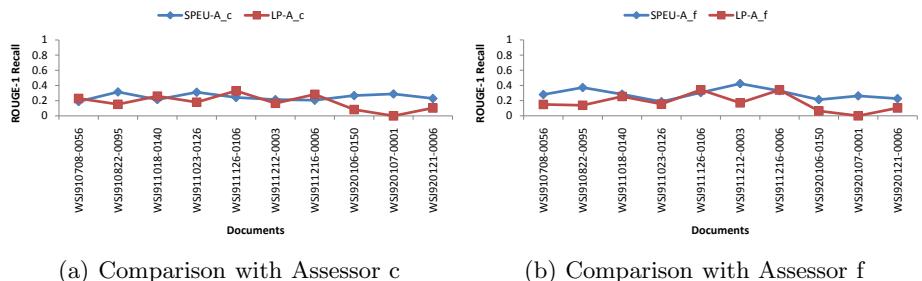
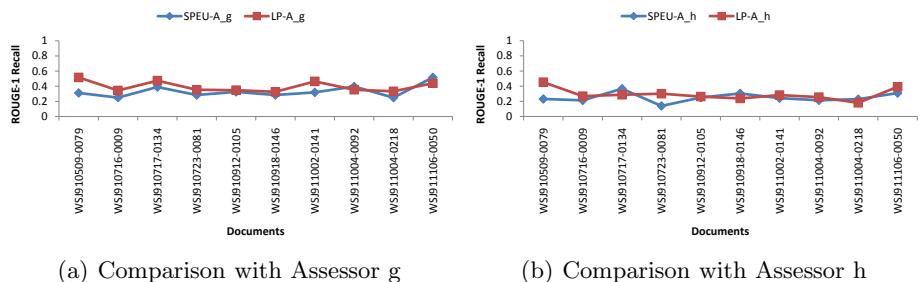


Fig. 1. Daycare dataset

**Fig. 2.** Healthcare dataset**Fig. 3.** Pres92 dataset**Fig. 4.** Robert.Gates dataset

words, and if they are not found in WordNet, then SPEU cannot function properly as HDS and FWS are WordNet dependent for its semantic analysis. Besides this, it is also noticed that POS tagger which is used here failed to tag properly in many cases, as a result of which it affected HDS' performance, which directly affected SPEU method described here.

Now, in table 2, the results are averaged over the whole data sets. Here SPEU shows higher similarity with human assessors than LP.

Table 2. Summary of all four sets of results (ROUGE-1 Recall)

	LP-A_1	LP-A_2	SPEU-A_1	SPEU-A_2
daycare(d,i)	0.19	0.26	0.29	0.31
healthcare(b,j)	0.27	0.35	0.28	0.27
pres92(c,f)	0.20	0.19	0.24	0.29
robert.gates(g,h)	0.40	0.29	0.33	0.25
Average	0.26	0.27	0.29	0.28

5 Conclusion

This paper presented a semantic extension of Subjective Logic for document analysis. To formulate this, HDS was used to compute similarity between composite events. Using the semantic information, it is noticed that the opinions about a sentence are more subjective and similar to human decisions. The evaluation of the highly opinionated sentences when compared with baseline summaries of the DUC data set shows that they are more similar to human assessors. The possible reason is the use of semantic information for formulating. Noise entered in the model while using the NLP tools at the same time less coverage of information by WordNet have influenced the performance to a greater extent for some documents. Thus as a future work, these aspects needs to be taken care of by using fuzzy methods to reduce noise in using HDS, at the same time membership functions of FIS needs tuning at word level for further refinement of the fuzzy outputs using different machine learning methods.

References

1. DUC, The document understanding conference (2001), <http://duc.nist.gov/>
2. Achananuparp, P., Hu, X., Shen, X.: The evaluation of sentence similarity measures. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 305–316. Springer, Heidelberg (2008)
3. Dalianis, H.: SweSum-A Text Summarizer for Swedish (2000), <http://www.dsv.su.se/%7Ehercules/papers.Textsumsummary.html>
4. Jøsang, A.: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 279–311 (2001)
5. Josang, A.: Belief Calculus. ArXiv Computer Science e-prints (June 2006)
6. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation, pp. 24–26. ACM Press, New York (1986)
7. Liddy, E.D.: The discourse-level structure of empirical abstracts: an exploratory study. Information Processing and Management 27(1), 55–81 (1991)
8. Lin, C.-Y.: Rouge: A package for automatic evaluation of summaries. Association for Computational Linguistics, Barcelona, Spain, pp. 74–81 (July 2004)

9. Lin, C.Y., Hovy, E.: Identifying topics by position. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 283–290. Morgan Kaufmann Publishers Inc, San Francisco (1997)
10. Lin, C.Y., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, P. 78. Association for Computational Linguistics (2003)
11. Lin, D.: Using syntactic dependency as local context to resolve word sense ambiguity. In: Annual Meeting-Association For Computational Linguistics, vol. 35, pp. 64–71. Association For Computational Linguistics (1997)
12. Manna, S., Gedeon, T.: Semantic Hierarchical Document Signature In Determining Sentence Similarity. In: 2010 IEEE World Congress On Computational Intelligence, 19th International Conference on Fuzzy Systems, pp. 2780–2787. IEEE, Los Alamitos (2010)
13. Manna, S., Mendis, B.S.U.: Fuzzy Word Similarity: A Semantic Approach Using WordNet. In: 2010 IEEE World Congress On Computational Intelligence, 19th International Conference on Fuzzy Systems, pp. 1761–1768. IEEE, Los Alamitos (2010)
14. Manna, S., Mendis, B.S.U., Gedeon, T.: An Enhanced Framework Of Subjective Logic For Semantic Document Analysis (to appear). In: The 7th International Conference on Modeling Decisions for Artificial Intelligence, Springer, Heidelberg (2010)
15. Manna, S., Mendis, B.S.U., Gedeon, T.: Advances in knowledge discovery and data mining, 14th pacific-asia conference, pakdd 2010, hyderabad, india, proceedings. part ii. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) Advances in Knowledge Discovery and Data Mining. LNCS, vol. 6119, Springer, Heidelberg (2010)
16. Mendis, B.S.U.: Fuzzy Signatures: Hierarchical Fuzzy Systems and Applications (PhD thesis). PhD thesis, College of Engineering and Computer Science, The Australian National University, Australia (2008)
17. Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM 38(11), 41 (1995)
18. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: International Joint Conference on Artificial Intelligence, vol. 14, pp. 448–453. Citeseer (1995)

Is Comprehension Useful for Mobile Semantic Search Engines?

Ahmad Ali Iqbal¹ and Aruna Seneviratne²

¹ School of EE&T, University of New South Wales (UNSW), Australia

² National Information and Communication Technology Australia (NICTA)

Abstract. Semantic web is gaining popularity as a candidate for next generation World Wide Web. In recent years, there has been a tremendous increase of using Internet on mobile devices and search engines are considered essential for Internet users. The existing search engines have been designed for powerful computers and are highly resource hungry, while mobiles have limited computational resources. In this paper, we study the use of comprehension for aiding the search engine results for mobile users. Our preliminary evaluation shows the promising results for comprehension generation.

1 Introduction

Semantic search engines are in action since last couple of years. They have focused on improving the quality of search in terms of data availability, search delay and ranking of the results [4]. However, it is evident that on average, users get much higher results than they actually visit from the search results. This reduces the viability of showing all results against the user query, the resource usage for generating ranked results and the data transfer cost specially when search engines are accessed by the mobile users.

In recent years, mobile search engine optimization focuses on reducing the content size by either lessening the content itself or by summarizing the content through major heading in the content. Such optimizations are effective to a certain extend by trading off the detailed information to the end user. Also such optimizations are server centric and are more effective for content server than the mobile device where content servers are full fledged and powerful machines connected through gigabit Internet connection. Instead we believe that further optimization can be achieved by removing the data redundancy from the search results of a user query and deliver the contents to a user by transferring only the redundancy removed data through wireless network.

1.1 Semantic Web and Search

Current web is for communication between human and machines through Internet whereas the goal of semantic web is to integrate universally available data and make it useful for inter machine communication. It improves the interoperability between systems, applications and information sources thus enables the

higher degree of automation of many tasks that would otherwise requires human intervention [9]. In data access from web, the main driver is search engine. Similarly, semantic search engines are the enablers to the semantic web data access. To get advantage from exponentially growing semantic web data, various semantic web search engines are constructed in recent years. Few of them are Swoogle [4], Watson¹, Aqualog [10], Hakia [11][13] and SenseBot².

1.2 Mobile Search

Mobile users that use data services on regular basis are in seek of efficient access to the data they want. Like for traditional Internet users, search engines also play a vital role for mobile users for locating the data sources. The search performed by mobile users are most likely to fulfill immediate requirement like downloading a ringtone or to find the route to a desired destination. The query typed in is short containing only few words but the traditional search engines return thousands of results. It is shown that 50, 33.9, 10.5 and 3.7 percent of mobile searches consist of 1, 2, 3 and 4 word queries respectively³. Many of these pages contain redundant or irrelevant data or many times, users end up by searching again with different single / combination of keywords for determining the sources of data of their interest. Also it is trivial that people normally do not navigate away from the first few pages of search results. In this way more useful and relevant information might be overlooked and the user may begin the search again by typing different keywords and in this way inefficiency takes over by waiting the resources. As in case of mobile phones many constraints are involved like battery life, limited available data rate and small mobile screen so above mentioned problem will cause wastage of resources like time and mobile battery. There is a room to improve existing search engines for end users interest in order to address these problems. Despite giving a list of thousands of links to the actual data, there must also be a concise summary of search results so that user can immediately determine the contents in search results. The resultant pages that contain redundant data should be removed and a single concise summary of the result data should be offered so that users quickly determine whether their searched contents are what they are actually looking for or not. Latter in this paper we refer this concise summary as a comprehension.

In order to achieve this, we present a proxy server based architecture that employs a proxy server⁴ for acting as an agent for mobile user, fetches the contents, perform data redundancy removal and offer the comprehension as a result of a user query. In order to remove the data redundancy, we present an architecture that makes use of a Bloom filter. These proxies get the contents according to user queries, perform the redundancy removal and deliver these redundancy removed contents to mobile users.

¹ <http://kmi-web05.open.ac.uk/WatsonWUI/>

² <http://www.sensebot.net/>

³ <http://searchenginewatch.com/3624288>

⁴ It is a resourceful machine operating with a regular power and connected through a broadband Internet

Rest of this paper is structured as follows; Related work is presented in section 2. Section 3 describes the architecture for generating comprehension for mobile search. In section 4, we setup the evaluation environment and describe the preliminary performance in terms of data redundancy elimination. Finally, we conclude this paper in section 5.

2 Related Work

Previous work has looked at various aspects of searching through Semantic Web stores and summarization aspects through text mining [7][11] and Semantic Web data [15] but to the best of our knowledge no integrated work that involves Ontology summarization for mobile search engines is presented in literature. These summary generation approaches can not be as useful as employing a Bloom filter because the summary size they create are still much larger. However, Bloom filter generates the summary for the sake of element membership testing only and is already widely used in various networking and database applications [3][5][14]. In this section, we cover related aspects from summarizing the semantic web data, intelligent summary generation through natural language processing and summarization by the search engines.

SenseBot is a web search engine which summarizes search results into one concise digest on the topic of your query. It uses text mining for generating the summary. Our work differs from SenseBot in a way that we assign the data redundancy elimination to a proxy server. This way, our approach can be used with most of the existing search engines. Also we employ Bloom filter to determine the data redundancy between the Resource Description Framework (RDF) statements.

The architectures like [8][12] and [2] are the foundation and made the basis for contribution on redundancy removal approaches on semantic web data as is presented in this paper. Our algorithms employ summary exchange similar to [5] between peers to make them aware of distributed data redundancy thus avoid transmitting redundant data transmission on the network. In literature, great amount of work on creating the summary of the natural language text [7][11] and RDF [15] is presented. However, such summary generation approaches can not be as useful for redundancy removal as Bloom filter, due to the summary size they create. Data redundancy removal approaches only needs the summary for membership testing of data elements and Bloom filter is widely accepted in various applications [3][5][14].

A well known example which employs the Bloom filters for summary exchange is distributed web caching [5] where each web proxy periodically exchange the Bloom filter of its cache with other proxies. This way proxies become aware of the contents of their neighbors. They only send the query to remote peer, if it can not be resolved from local cache. In [14], authors compared the performance of full text search in peer-to-peer networks using Bloom filters and DHTs along with inverted lists. *Yang et. al.* showed that structured peer-to-peer networks depict the best response time at the higher document publishing cost.

3 Architecture

Our architecture is based on a proxy server namely *Mobile Proxy Server (MPS)* that is responsible to remove the data redundancy for generating a concise summary before providing it to the mobile users. Proxy server is a powerful machine with a gigabit Internet connectivity that performs the major task and mobile users get the benefit from it. MPS provides the WiFi interface for such a connectivity. As stated earlier, MPS uses the Bloom filters for data redundancy elimination as follows;

1. It gets the user query from a mobile, uses a traditional search engine (say swoogle^[4], a semantic search engine),
2. It gets the RDF triples from first search result link,
3. It creates a Bloom filter on it using a pre-defined parameters (filter size, number of hash functions etc.),
4. It gets RDF triples from next result link,
5. It discards those RDF triples that previously created Bloom filter determines as redundant entries,
6. It updates Bloom filter for all new RDF triples,
7. It repeats the process from step 2 to step 6 for next few search result links and finally deliver redundancy eliminated set of RDF triples to a mobile user.

When a keyword is searched through a search engine, the RDF documents fetched by the search engine contains a significant portion of data redundancy. However, these document authors use different ontologies thus we adopt Ontology alignment algorithm first and then remove the data redundancy. This way the RDF ontologies generated by two different authors get aligned.

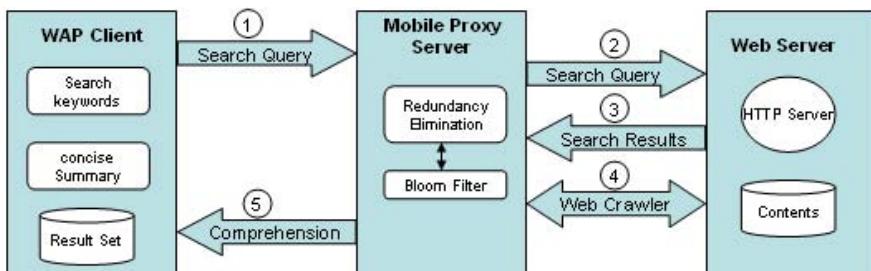


Fig. 1. Mobile Proxy Model

4 Evaluation

This section presents the preliminary evaluation results drawn based on processing delay for redundancy removal and number of duplicate RDF statements

removal. For the sake of our preliminary evaluation, we used Swoogle⁵ semantic search engine for getting the search links and use RDF documents from these links for our experiment. Like *Iterative Summary Exchange* algorithm⁶, documents are used one after the other from the search result links and RDF statements from firstly appeared search link are used for creating a Bloom filter such that all statements from this document are included in a new document. Non redundant RDF statements from other search result links are appended to this new document. We set a threshold to a maximum of 10 documents for redundancy elimination. We use Sesame API⁷ for semantic web data access and SPARQL query. Further, we used the following five keywords; *Bloom filter*, *knowledge*, *wireless*, *Internet* and *data redundancy* for determining the processing delay for comprehension generation⁸ and then choose two keywords *Bloom filter* and *Internet* for determining the number of duplicate statements between the documents that can be eliminated for comprehension generation from search result documents.

Table 1 shows the data redundancy elimination delay between the statements of documents for individual keywords. This experiment first retrieve the RDF document by searching each keyword using a Swoogle search engine and remove the data redundancy using a Bloom filter. Our experiment truncate the keyword search results to 10 if it finds more than 10 documents e.g. for *Internet* keyword otherwise it uses all the documents e.g. for *Bloom filter* keyword.

Table 1. Data redundancy elimination delay for five different keywords

No.	Keyword	Elimination delay (msec)
1	Bloom filter	21.8
2	Knowledge	805.0
3	Wireless	574.3
4	Internet	863.8
5	Data redundancy	339.5

Table 2 and 3 compares the number of redundant statements that are eliminated for comprehension generation with the total statements if they contain redundancy. *No of statements* is the total number of statements in a corresponding document fetched by Swoogle, *Statements added* is number of RDF statements added in comprehension after redundancy elimination, *Total statements* refers to the aggregated statements in a comprehension from a number of documents and *% Improvement* shows the percentage of statements eliminated.

⁵ <http://www.openrdf.org/doc/api/index.html>

⁶ In table 1, it is referred as *elimination delay*

Table 2. Swoogle Results for a *Bloom Filter* keyword

Document	No of statements	Statements added	Total statements	% Improvement
1	25219	25219	25219	0
2	21619	3181	28400	85.3
3	195	6	28406	96.9

Table 3. Swoogle Results for a *Internet* keyword

Document	No of statements	Statements added	Total statements	% Improvement
1	12853	12853	12853	0
2	24415	5029	17882	79.4
3	16139	2042	19924	87.3
4	27518	4088	24012	85.1
5	9630	1137	25149	88.2
6	14505	1961	27110	86.5
7	15952	1758	28868	89.0
8	2087	253	29121	87.9
9	23294	1079	30200	95.4
10	9626	392	30592	95.9

5 Conclusion

Mobile phones are limited in resources whereas existing trend of Internet usage on mobile phones requires an optimized content delivery from search engines. We present a web proxy based architecture that eliminates the redundant data and gives the concise results of a user query so that user can promptly determine the contents of search results. We employ Bloom filter for data redundancy elimination and evaluated its performance using swoogle search engine.

Acknowledgments

NICTA is funded by the Australian Government's Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Research Centre of Excellence programs.

References

1. Hakia: Semantic search technology. White Paper (2010)
2. Adamku, G., Stuckenschmidt, H.: Implementation and evaluation of a distributed rdf storage and retrieval system. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI), USA (2005)

3. Broder, A., Mitzenmacher, M.: Network applications of bloom filter: A survey. *Internet Mathematics* I(4), 485–509 (2003)
4. Ding, L., Finin, T.: Boosting semantic web data access using swoogle. In: AAAI 2005: Proceedings of the 20th National Conference on Artificial Intelligence, pp. 1604–1605. AAAI Press, Menlo Park (2005)
5. Fan, L., Cao, P., Almeida, J., Broder, A.Z.: Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networks* 8(3) (2000)
6. Iqbal, A.A., Ott, M., Seneviratne, A.: Removing the redundancy from distributed semantic web data. In: Bringas, P.G., Hameurlain, A., Quirchmayr, G. (eds.) *Database and Expert Systems Applications. LNCS*, vol. 6261, pp. 512–519. Springer, Heidelberg (2010)
7. Jing, H.: Sentence reduction for automatic text summarization. In: 6th Conference on Applied Natural Language Processing, pp. 310–315. Association for Computational Linguistics (2000)
8. Karnstedt, M., Sattler, K.-U., Richtarsky, M., Müller, J., Hauswirth, M., Schmidt, R., John, R.: Unistore: Querying a dht-based universal storage. In: ICDE (2007)
9. Lassila, O.: Using the semantic web in mobile and ubiquitous computing. In: 1st IFIP WG12.5 Working Conference on Industrial Applications of Semantic Web (2005)
10. Lopez, V., Uren, V., Motta, E., Pasin, M.: Aqualog: An ontology-driven question answering system for organizational semantic intranets. *Web. Semant.* 5(2), 72–105 (2007)
11. Silla, C.N., Pappa, G.L., Freitas, A.A., Kaestner, C.A.A.: Automatic text summarization with genetic algorithm-based attribute selection. *Advances in Artificial Intelligence (IBERAMIA)*, 305–314 (2004)
12. Solomou, G.D., Koutsomitropoulos, D.A., Papatheodorou, T.S.: Semantic-aware querying of web-distributed rdf(s) repositories. In: 1st Workshop on Semantic Interoperability in the European Digital Library (SIEDL 2008) with ESWC (2008)
13. Turner, D., Shah, M.A., Bitirim, Y.: An empirical evaluation on semantic search performance of keyword-based and semantic search engines: Google, yahoo, msn and hakia. In: ICIMP 2009: Proceedings of the 2009 Fourth International Conference on Internet Monitoring and Protection, pp. 51–55. IEEE Computer Society, Washington (2009)
14. Yang, Y., Dunlap, R., Rexroad, M., Cooper, B.F.: Performance of full text search in structured and unstructured peer-to-peer systems. In: IEEE INFOCOM (2006)
15. Zhang, X., Cheng, G., Qu, Y.: Ontology summarization based on rdf sentence graph. In: 16th International Conference on World Wide Web (WWW), USA (2007)

A Novel Text Classification Approach Based on Deep Belief Network

Tao Liu^{1,2}

¹ Key Laboratory of Data Engineering and Knowledge Engineering
(Renmin University of China), MOE, 100872 Beijing, China

² School of Information, Renmin University of China,
100872 Beijing, China
tliuruc@google.com

Abstract. A novel text classification approach is proposed in this paper based on deep belief network. Deep belief network constructs a deep architecture to obtain the high level abstraction of input data, which can be used to model the semantic correlation among words of documents. After basic features are selected by statistical feature selection measures, a deep belief network with discriminative fine tuning strategy is built on basic features to learn high level deep features. A support vector machine is then trained on the learned deep features. The proposed method outperforms traditional classifier based on support vector machine. As a dimension reduction strategy, the deep belief network also outperforms the traditional latent semantic indexing method. Detailed experiments are also made to show the effect of different fine tuning strategies and network structures on the performance of deep belief network.

Keywords: text classification, deep belief network, support vector machine.

1 Introduction

Text classification is the task of assigning free text documents to predefined categories based on their content. With the explosion of web information, efficient organization and management of web information has becomes more and more important to help users find their needed resources. Automatic text classification [1] is a key technology in organizing large scale online resources. Many efforts have been paid on this problem. A number of statistical and machine learning approaches [2] have been proposed for text classification, such as K-nearest neighbor, Bayesian classifier, neural networks and support vector machine. These techniques obtain satisfying performance in some text classification tasks. However, for the problem of text classification with large number of classes, the performance of text classification is still not satisfying. More effective algorithms need to be proposed.

Documents should be represented in some form which is the input of supervised learning algorithm. The most common document representation model, vector space model [3], assumes different words are independent from each other. In fact, there exists semantic relatedness among different words. Two documents may be related to each other though they don't share common words. Deep belief network proposed

recently (DBN) [4] can be used to model the words correlation of documents. It learns high level representation of data by constructing a deep architecture [5], which overcomes the limitations of traditional methods with shallow architecture, and also avoids the “explaining away” effect existed in dimension reduction method based on latent dirichlet allocation [6]. Though DBN has been successively applied to many fields such as digit recognition [4], language model [7] and semantic search [8], to the best of our knowledge, the research of it on text classification task is rarely reported until now. In this paper, we make an extensive study on using DBN to solve the text classification problem, and propose a novel approach for text classification based on deep belief network, which outperforms traditional state-of-art methods.

2 Method

Automatic text classification is a supervised learning task, which is to find a mapping function $f : D \times C \rightarrow \text{Boolean}$ between a set of documents D and a set of predefined categories $C = \{c_1, c_2, \dots, c_m\}$. The mapping function f is learnt from a training documents set T , and each document of T is labeled according C . The unstructured document should be represented as a machine readable form before supervised learning method is built.

2.1 Document Representation with Feature Selection

In the classical vector space model [3], each document is represented as a vector of words counts. Large amount of words forms a high dimensional sparse feature space. Feature selection is required to reduce the computation complexity and make text classification more efficient. We select features based on our previously proposed statistical measures [9]: normalized corpus impurity (NCI) and normalized domain impurity (NDI). Features are selected based on two principles using the categorized training corpus: within the categorized corpus, representative words tend to have biased occurrence distribution among different categories, and have uniform occurrence distribution among different documents of its relevant category. NCI for a word w is defined as follows:

$$NCI(w) = - \sum_{i=1}^m P'(C_i | w) \log P'(C_i | w) \quad (1)$$

$$P'(C_i | w) = \frac{P(C_i | w) / L_i}{\sum_{j=1}^m [P(C_j | w) / L_j]} \quad (2)$$

where $P(C_i | w)$ is the probability that a word w occurs in category C_i , and $P'(C_i | w)$ is the normalized form of $P(C_i | w)$ by L_i , the sum of lengths of training documents belong to C_i . NDI for a word w in a certain category C_i is defined similarly as NCI to measure the distribution of w in different documents of C_i . Words with lower NCI value and higher NDI in its most related category are taken as features. The weight of feature i in document j is assigned using the classical OKAPI weighting formula [10].

Compared with traditional feature selection methods such as TFIDF, document frequency (DF), information gain (IG), mutual information(MI), χ^2 statistic (CHI) and expected cross entropy(ECE) [11], NCI and NDI can make best use of the word distribution among different categories and within its relevant category.

2.2 Deep Feature Learning

Based on vector space model, documents are represented as vectors of features. This representation mechanism assumes words are independent from each other. In fact, the phenomena of synonymy and polysemy commonly exist in text corpora, which makes different words are correlated. We use deep belief network to learn deep features of texts and to capture the correlation among different words.

Deep belief network (DBN) constructs a deep architecture to obtain the high level abstraction of input data [12]. DBN is a multi-level neural network, which includes one visible layer and several hidden layers. Every layer is composed of certain number of stochastic neurons. There are full connections between neurons of two adjacent layers, and no intra-layer connections. The learning goal of DBN is adjusting weights between neurons to make the network more likely to generate the input data. In such a densely connected net, learning all parameters at a time is very difficult and inefficient. A greedy layer-wise algorithm [13] is used for learning, which improves a variational lower bound on the log probability of the training data.

2.2.1 Learning Method

Each pair of two adjacent layers of DBN forms a Restricted Boltzmann Machines (RBM) [14]. The DBN is constructed by training one RBM at a time in a bottom up fashion. Each RBM includes one hidden layer h and one visible layer v , and there are undirected, symmetric connections between them. The hidden units of the i^{th} RBM are taken as the input visible units of the $(i+1)^{\text{th}}$ RBM. A combination of v and h in one RBM has energy:

$$E(v, h) = -\sum_{s=1}^{n^v} b_s^v v_s - \sum_{t=1}^{n^h} b_t^h h_t - \sum_{s=1}^{n^v} \sum_{t=1}^{n^h} v_s h_t w_{st} \quad (3)$$

where v_s and h_t are the s^{th} and t^{th} units of v and h , b_s^v and b_t^h are the biases of unit v_s and h_t , and w_{st} is the weight between v_s and h_t . The probability of generating data is proportional to the energy $E(v, h)$. Notice that $\partial E(v, h) / \partial w_{st} = -v_s h_t$ and $p(v) = \sum_h e^{-E(v, h)} / \sum_{u, g} e^{-E(u, g)}$, then the derivative of the log probability of visible data is $\partial \log p(v) / \partial w_{st} = \langle v_s h_t \rangle^0 - \langle v_s h_t \rangle^\infty$.

Given a visible vector v of one RBM, the output of h_t is:

$$p(h_t = 1 | v) = \sigma(b_t^h + \sum_{s=1}^{n^v} w_{st} v_s) \quad (4)$$

where $\sigma(x) = 1/(1 + e^{-x})$. New visible units are then generated using the state of the hidden units by:

$$p(v_s = 1 | h) = \sigma(b_s^v + \sum_{t=1}^{n^h} w_{st} h_t) \quad (5)$$

Contrastive divergence is used to update parameters by:
 $\Delta w_{st} = \epsilon (\langle v_s h_t \rangle^0 - \langle v_s h_t \rangle^1)$, which approximately follows gradient objective.

In a bottom up recognition procedure of the multi-layer network, symmetric weights can be used to infer more abstract representation of data; in a top down generative procedure, these weights are used to generate data.

2.2.2 Fine Tuning the Weights

In the layer-wise learning process of DBN, the weights updating for every RBM is independent: the weights of lower layer RBMs aren't updated after the weights of higher layer RBMs are obtained. It is necessary to fine tune all the parameters after the layer-wise learning to make the whole network to work much better. There are two kinds of fine-tuning methods based on different optimizing goals: one is to maximize the probability for the current model to generate the training data, the other is to minimize the classification error on training set of the current model. These two kinds of fine tuning strategies are both implemented by back propagation algorithm. We explore the use of these two methods on the text classification problem, and analyze the difference between these two kinds of methods. Optimizing the first generative goal can make the model to fit with the training data much better. This optimizing strategy doesn't definitely guarantee the model working very well on certain classification task.

When confronted with different classification objectives, features in distinguishing the same training data may be very different. For example, features for text classification according to academic domains or genre are different. Therefore, fine tuning the network according to certain classification goal is important to make the learned deep features adapted to different classification objective. There are two ways to get the classification error: One is to take the last layer of deep network as class label vector, and the other is to combine the deep network with a classifier. In the experiment part, we attempt different fine tuning methods.

2.3 Text Classifier

After the deep feature representation, we use the support vector machine (SVM) as a classifier. SVM has shown good performance on text classification tasks [15, 16]. SVM operates by transforming samples into a high-dimensional space and finding an optimal hyperplane in this space. The optimal hyperplane is chosen to separate as more as possible samples correctly and maximize its distance from the closest training samples while keeping the largest margin simultaneously. SVM is a classifier for binary classification problem and we use the one verse one conversion strategy to make it be adapted to multiclass problem.

3 Experiment

3.1 Dataset and Evaluation Criteria

We evaluate the proposed method on Chinese corpora, which are constructed using test datasets of the text classification evaluation task of High Technology Research

and Development Program (863) in 2003 and 2004. These datasets both include 36 categories which accord with the Chinese Library Classification (CLC) standard except omitting two categories (T and Z). The number of documents of different categories is nearly the same. There are totally 7200 documents after combining these two datasets. We randomly select 6000 documents of them as training data, and take the left 1200 documents as test data. In the experiment, each document belongs to one category. For all categories, we use accuracy (Micro-Precision), Macro-recall, Macro-precision and Macro-F1 score to evaluate the performance of text classification.

3.2 System Implementation

There are no explicit delimiters to indicate word boundaries in Chinese texts. Therefore we preprocess the text classification corpus by word segmentation using a Chinese lexical analysis tool ICTCLAS¹ developed by Institute of Computing Technology of Chinese Academy of Sciences. Then according to the statistics of words among the training corpus, original features are defined by the *NCI* and *NDI* measures. Threshold α and β are empirically set to 3.0 and 0.5 respectively. There are totally 1800 selected features. Each document is represented as a vector of feature weights which are computed by the OKAPI formula.

LIBSVM² package is used as the implementation of SVM. SVM is a classifier for binary classes. We use the one verse one strategy to convert the multiclass categorization problem to binary classification problem. We conducted experiments by taking the linear function and radial basis function (RBF) as kernels respectively. For the RBF kernel, we use grid search to seek the best choice of penalty factor c and the RBF parameter gamma. We discover that the optimized value of gamma is very small such as 2^{-20} , where the RBF kernel is more like a linear kernel. Therefore we only listed the experimental results obtained by the linear kernel in the following sections, and the penalty factor c is set to 1.

3.3 The Effect of Fine Tuning of DBN

We implemented the DBN with three settings to show the effect of fine tuning: 1) Without the use of fine tuning; 2) Using fine tuning with generative optimizing goal; 3) Using fine tuning with discriminative optimizing goal. In the first scenario, the weights on connections of all layers won't be updated after all RBMs have been trained layer-by-layer. In the second scenario, after the layer-wise training of DBN, the weights of the whole network are fine tuned using back propagation to maximize the probability of generating data. In the third scenario, the optimizing goal of fine tuning is to minimize the classification error, which is different from the one in the second scenario.

The number of hidden layers is set to three, and the number of units in the first, second and last hidden layer is set to 1500, 1200 and 900 respectively. The number of the visible units is 1800, which is same as the number of the input features. As shown in Table 1, fine tuning can improve the micro-precision of the proposed method on all settings of epoch. The best micro-precision obtained by generative fine tuning is 5.33

¹ <http://ictclas.org/>

² LIBSVM: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

percent higher than the best one obtained by method not using fine tuning. The discriminative tuning obtains better micro-precision than generative fine tuning method. We attempt two different iterative epochs (200 and 100) for fine tuning. The scenario of epoch 100 obtains more steady performance when the epoch of training RBM varies. When the iterative epochs for RBM takes 300, the discriminative fine tune method obtains highest accuracy 76.83%.

Table 1. Effect of training epoch/fine tuning epoch on micro-precision of the DBN+SVM

Initial training epoch	No fine tune	Fine tune		Discriminative fine tune	
		200 epoch	100 epoch	200 epoch	100 epoch
2000	69.92%	73.34%	74.75%	75.33%	75.58%
1000	67.75%	74.42%	75.08%	75.75%	75.92%
500	66.67%	74.83%	75.17%	75.17%	75.25%
300	67.33%	75.25%	74.92%	76.00%	76.83%
200	65.58%	74.50%	74.42%	74.92%	75.92%

3.4 The Effect of Different Network Structure

To show the influence of different network structure on the performance of DBN in the text classification task, we attempted eight different network structures which have different number of hidden units. As shown in Table 2, different network structures influence the text classification performance to some extent, but the change is not very large. The best micro-precision 76.83% is obtained using the “1800-1500-1200-900” structure on the method combining DBN with SVM, which is 1.75 percent higher than the lowest one obtained using the “1800-1800-1800-1800” structure. The best micro-precision obtained by using both DBN and SVM is 1.49 percent higher than that obtained by only using DBN.

Table 2. Effect of different network structure on DBN without /with the combination of SVM

Network structure	DBN	DBN + SVM			
	Micro-P	Micro-P	Macro-P	Macro-R	Macro-F
1800-1800-1800-1800	75.00%	75.08%	74.82%	74.78%	74.80%
1800-1700-1600-1500	74.67%	75.50%	75.45%	75.17%	75.31%
1800-1600-1400-1200	74.08%	76.08%	76.28%	75.72%	76.00%
1800-1540-1280-1000	73.67%	75.33%	75.02%	74.96%	74.99%
1800-1500-1200-900	75.00%	76.83%	76.81%	76.50%	76.66%
1800-1470-1140-800	74.67%	75.92%	75.81%	75.54%	75.68%
1800-1400-1000-600	75.17%	75.42%	75.33%	75.05%	75.19%
1800-1300-800-300	74.00%	76.33%	75.71%	75.94%	75.83%

3.5 Comparison with Other Methods

In the proposed method, DBN learns deep features for SVM and it performs as dimension reduction. Latent semantic indexing (LSI) [17] is a classical dimension reduction method. We compare the proposed method with two traditional methods: SVM only; LSI + SVM. NCI and NDI based feature selection methods are also used

in these methods. In the second method, the document feature matrix is taken as the input of the singular value decomposition (SVD) of LSI. SVD decomposes the input matrix M ($m \times n$) into three matrices $M = U\Sigma V^T$, where Σ ($r \times r$) is the diagonal matrix with rank r ($r \leq \min(m, n)$), and U, V satisfy $U^T U = V^T V = I_r$. An approximate decomposition $M \approx M_k = U_k \Sigma_k V_k^T$ is used which only keep the k largest singular values and delete corresponding columns of U and V . We define k as 900 which is the same as the number of output units of DBN. As shown in Table 3, the proposed method outperforms these two methods by 2.06 percent and 2.14 percent respectively. In this experiment, LSI decreases the performance of SVM classifier.

Table 3. Comparative results (DBN structure: 1800-1500-1200-900)

Method	Micro-P	Macro-P	Macro-R	Macro-F1
SVM	74.75%	74.83%	74.38%	74.60%
LSI + SVM	74.58%	74.82%	74.22%	74.52%
DBN + SVM	76.83%	76.81%	76.50%	76.66%

4 Conclusion

In this paper, we have proposed a novel text classification method using DBN for Chinese text classification. DBN develops high level abstractions of documents, which improves the F1 measure of text classification when it is combined with a support vector machine. Discriminative fine tuning is very important in making the learned deep features be more adapted to the classification goal. Different network structures are attempted to show the influence on the performance of classification. Only using DBN with discriminative fine tuning (without SVM) can obtain comparable result with only using support vector machine. Building support vector machine on the basis of deep features, can obtain better accuracy and F1 measure in the text classification problem. As a dimension reduction strategy, DBN outperforms the traditional latent semantic indexing method in the experiment.

During the process of peer review, a reviewer suggests a more effective feature dimension reduction method [18], which will be compared with our method in future.

Acknowledgments. This work was supported by the National Natural Science Foundation of China with grant No. 60873017, 60803092 and 61003204, and Scientific Research Fund of Renmin University of China with a grant No. 22382074.

References

1. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computer Surveys 34(1), 1–47 (2002)
2. Yang, Y., Liu, X.: A Re-examination of Text Categorization Methods. In: 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 42–49. ACM Press, New York (1999)
3. Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Processing and Management: an International Journal 24(5), 513–523 (1988)

4. Hinton, G.E., Salakhutdinov, R.R.: Reducing the Dimensionality of Data with Neural Networks. *Science* 313(5786), 504–507 (2006)
5. Bengio, Y.: Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning* 2(1), 121–127 (2009)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3(1), 993–1022 (2003)
7. Mnih, A., Hinton, G.E.: A Scalable Hierarchical Distributed Language Model. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1081–1088 (2008)
8. Salakhutdinov, R., Hinton, G.: Semantic Hashing. *International Journal of Approximate Reasoning* archive 50(7), 969–978 (2009)
9. Liu, T., Wang, X.L., Guan, Y., Xu, Z.M., et al.: Domain-specific Term Extraction and its Application in Text Classification. In: *8th Joint Conference on Information Sciences*, pp. 1481–1484 (2005)
10. Yang, Y., Ault, T.: kNN at TREC-9. In: *9th Text REtrieval Conference (TREC 1999)*, pp. 127–134 (1999)
11. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: *14th International Conference on Machine Learning*, pp. 412–420 (1997)
12. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy Layer-wise Training of Deep Networks. In: *Advances in Neural Information Processing Systems*, pp. 153–160 (2007)
13. Hinton, G.E., Osindero, S., Teh, Y.W.: A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation* 18, 1527–1554 (2006)
14. Hinton, G.E.: Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation* 14(8), 1771–1800 (2002)
15. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: *10th European Conference on Machine Learning*, pp. 137–142 (1998)
16. Kwok, J.-Y.: Automatic Text Categorization Using Support Vector Machine. In: *International Conference on Neural Information Processing*, pp. 347–351 (1998)
17. Deerwester, S., Dumais, S., Landauer, T., et al.: Indexing by Latent Semantic Analysis. *Journal of American Society of Information Science* 41(6), 391–407 (1990)
18. Manna, S., Petres, Z., Gedeon, T.D.: Tensor Term Indexing: An Application of HOSVD for Document Summarization. In: *4th International Symposium on Computational Intelligence and Intelligent Informatics*, pp. 135–141 (2009)

A Probability Click Tracking Model Analysis of Web Search Results

Yujiu Yang*, Xinyi Shu*, and Wenhua Liu

Information Science and Technology Division,
Graduate School at Shenzhen,Tsinghua University
{yang.yujiu.liuwh}@sz.tsinghua.edu.cn,
shuxy08@mails.tsinghua.edu.cn

Abstract. User click behaviors reflect his preference in Web search processing objectively, and it is very important to give a proper interpretation of user click for improving search results. Previous click models explore the relationship between user examines and latent clicks web document obtained by search result page via multiple-click model, such as the independent click model(ICM) or the dependent click model(DCM),which the examining-next probability only depends on the current click. However, user examination on a search result page is a continuous and relevant procedure. In this paper, we attempt to explore the historical clicked data using a probability click tracking model(PCTM). In our approach, the examine-next probability is decided by the click variables of each clicked result. We evaluate the proposed model on a real-world data set obtained from a commercial search engine. The experiment results illustrate that PCTM can achieve the competitive performance compared with the existing click models under standard metrics.

1 Introduction

Larger and larger amounts of user click data collected and stored in web search increases the demand for effective analysis methods to find the user interests and preferences. Web search click is probably one of the most extensive and direct user behaviors [6], and click logs can be leveraged for many tasks including personalization [9], query recommendation [3], and optimizing search engines [5]. An challenging problem of click log analysis is to infer user-perceived relevance for each query-URL pair. In general, user clicks are also influenced by the rank of results, not only the results themselves. Users tend to click results in top positions more, even though the results at the bottom are more relevant, which results in the so-called *position bias* [5] problem.

A representative work of click model by Craswell et al.[1] conducted *cascade model* adopted by the state-of-the-art click models[2,3,4,8]. Among these models, the probability that a user continues to see the next result (*i.e. examine-next probability*) only depends on the current result or the preceding last clicked result,

* The first two authors contributed equally to this study.

which is some similar with first-order Markov chain. Intuitively, the examine-next probability should be determined by whether the user is satisfied with the information got from the query session. For example, if a user feels that all the clicked documents provide enough information, then the user will quit the search session. Therefore, the examine-next probability is supposed to depend on all the preceding clicked positions, not only one position.

In this paper, we propose the *Probability Click Tracking Model* (PCTM) which takes the information of all the preceding clicked results into consideration, including the examination and click events, and document relevance. This is the primary advantage of PCTM over other existing click models. The experiments verify the feasibility and effectiveness of the proposed model.

The rest of this paper is organized as follows. We first introduce preliminaries and the previous click models in Section 2. Then we present our new model PCTM in Section 3. Section 4 is devoted to experiment studies, comparing PCTM with two state-of-the-art models(DCM and CCM). Finally, the paper is concluded in Section 5.

2 Preliminaries and Related Works

To clarify the click model clearly, we give some definitions and notations here. If the user reads the abstract of a document in the result page, then we say that this document is *examined* by the user. The relevance of each query-URL pair (*document relevance*) is a continuous variable which measures the relevance between queries and documents, and it ranges in [0,1]. In a query session, the result pages only consider the top M (usually $M = 10$) documents which can be represented as d_1, \dots, d_M , and $n = 1, \dots, M$ is the *ranks (positions)* of documents. The corresponding document relevance can be represented as R_{d_1}, \dots, R_{d_M} . Note that R_{d_i} is a continuous variable which ranges in [0, 1], and the examination and click events are E_1, \dots, E_M and C_1, \dots, C_M , where $E_i = 1$ means that the document in position i is examined, $E_i = 0$ means not examined. C_i is the same.

In previous models, there exist two popular assumptions:

Assumption 1. [4] *Cascade hypothesis is summarized as follows :*

$$\begin{aligned} P(E_1 = 1) &= 1 \\ P(E_{i+1} = 1 | E_i = 0) &= 0 \end{aligned} \tag{1}$$

The examination is strictly linear to the position, and it is possible for a user to examine a document only if the preceding document is examined.

Assumption 2. [7] *Examination hypothesis is summarized as follows:*

$$\begin{aligned} P(C_i = 1 | E_i = 0) &= 0 \\ P(C_i = 1 | E_i = 1) &= R_{d_i} \end{aligned} \tag{2}$$

The user will not click a document if the document is not examined. If the user examines the document, then the click probability equals the document relevance.

Follow these notations, the *cascade model* [1] can be represented as:

$$P(E_{i+1} = 1 | E_i = 1) = 1 - C_i \quad (3)$$

This model implies that the number of clicked documents in a user query session is one or zero. Beyond a single click model, Fan Guo generalized cascade model to Multi-click model, and proposed DCM [4]. In DCM, the examine-next probability is calculated as follows:

$$\begin{aligned} P(E_{i+1} = 1 | E_i = 1, C_i = 1) &= \lambda_i \\ P(E_{i+1} = 1 | E_i = 1, C_i = 0) &= 1 \end{aligned} \quad (4)$$

CCM [3] describes the examine-next probability as follows:

$$\begin{aligned} P(E_{i+1} = 1 | E_i = 1, C_i = 1) &= \alpha_2(1 - R_{d_i}) + \alpha_3 R_{d_i} \\ P(E_{i+1} = 1 | E_i = 1, C_i = 0) &= \alpha_1 \end{aligned} \quad (5)$$

Furthermore, User Browsing Model(UBM)[2] argues that the examine-next probability is determined by the preceding clicked position r and the distance to this position d , that is, $P(E_i = 1 | C_{1:i-1}) = \beta_{r_i, d_i}$, where $r_i = \max\{j | j < i, C_j = 1\}$, $d_i = i - r_i$. UBM breaks the cascade hypothesis, but still follows the examination hypothesis.

3 Probability Click Tracking Model

In this section, we introduce our new model *Probability Click Tracking Model* (PCTM). Differing from the existing models, PCTM takes all the clicked documents of a query session into consideration, and the examine-next probability is a function of the examination and click events and relevance of the preceding documents, represented as :

$$P(E_{i+1} = 1 | E_i = 1) = f(C_1, \dots, C_i; E_1, \dots, E_i) \quad (6)$$

Intuitively, the previous click and examination information can be useful for the current click decision. Now, a new question is raised - how to choose an appropriate function to measure the relationships of $C_{1,\dots,i}$ and $E_{1,\dots,i}$. The estimation of document relevance can be done by using *maximum likelihood estimation* (MLE). By contrast, the inference of model parameters is difficult, so we set the parameters by the trial-and-error method.

3.1 Model Specification

Note that, PCTM still follows the cascade hypothesis and examination hypothesis. Firstly, we assume that the first document is always examined by the user. That is to say, the examine probability of the first document is 1, shown in Eq[1]

When the user examines a document d_i , he or she may be attracted by d_i and then clicks it. This conditional probability equals the document relevance

R_{d_i} . The examine-next probability $P(E_{i+1} = 1|E_i = 1)$ is determined by the function f in Eq.6. If the user does not examine d_{i+1} , we consider that the user will stop examining the following documents and quit the query session.

In our new model, $f(E_1, \dots, E_i; C_1, \dots, C_i)$ is defined as a function which ranges in [0,1]. We test several definitions of this function, and find that the following one performs the best:

$$\begin{aligned} P(E_{i+1} = 1|E_i = 1) &= f(C_1, \dots, C_i; E_1, \dots, E_i) \\ &= \prod_{j=1}^i \{C_j[\alpha(1 - R_{d_j}) + \beta R_{d_j}] + (1 - C_j)\} \end{aligned} \quad (7)$$

Like most other click models, PCTM only takes the top M ($M = 10$) documents into consideration. We assume that the user quits the query session immediately after d_M is examined or clicked.

3.2 Parameters Extimation

There are document relevances R_{d_1}, \dots, R_{d_M} and model parameters α, β to be inferred in PCTM. We use MLE to estimate them. For one query session, we can calculate the likelihood using the following equation:

$$\begin{aligned} L(\text{session}) &= P(C_1, \dots, C_M) \\ &= \prod_{i=1}^l P(E_i = 1)P(C_i|E_i = 1) \cdot P(C_{l+1}, \dots, C_M) \end{aligned} \quad (8)$$

where l is the last clicked position of the session.

Therefore, the log-likelihood for one query session is:

$$\begin{aligned} \log(L(\text{session})) &= \sum_l^{i=1} \log(P(E_i = 1)) + \sum_l^{i=1} \log(P(C_i|E_i = 1)) \\ &\quad + \log(P(C_{l+1}, \dots, C_M)) \end{aligned} \quad (9)$$

For document relevance, we pick out all the sessions which belong to a certain query and add the log-likelihood of each session to calculate the total log-likelihood of this query. Then we maximize the total log-likelihood to estimate the relevance of documents which appear in these query sessions. As the number of query-URL pairs for one query is usually not very large, computing document relevance for one query is not very difficult.

By contrast, the inference of model parameters α, β is complicated due to their appearance in all sessions of each query. There are so many query-URL pairs in click logs that it is almost impossible to solve the optimization problem directly. To simplify the inference, we set the values of α, β as constant selecting from trial-and-error test.

As we mentioned, MLE is used to estimate document relevance. Therefore, we have to maximize the log-likelihood function for each query. Obviously, the previous problem can be described as the following optimization problem:

$$\begin{aligned}
\min \quad & - \sum_{k=1}^n l(\text{session } k) \\
\text{s.t.} \quad & 0 \leq R_{d_i} \leq 1
\end{aligned} \tag{10}$$

where $k = 1, \dots, n$ are the indices of sessions of one query, and $l(\text{session } k)$ is the log-likelihood of session k . To attack this optimization problem, we solve it via a subspace trust region method, which is based on the interior-reflective Newton method.

4 Experiments

In this section, we compare PCTM with two state-of-the-art models CCM and DCM. All experiments are conducted on click logs provided by Sogou.

4.1 Experiment Setting

Data Set. The raw data set is sampled from the Sogou search engine logs¹ in August, 2006. The data format is defined as:

User ID\Query Words\URL Order\Item Order\Clicked URL

As we can see, every item of this data set corresponds to one click event, and the URLs not clicked will not be recorded in click logs. After preprocessing the raw data set, we get a data set consisting of 3,019,358 sessions and 4,044 queries. For each query, we divide all the sessions into two parts equally. Finally, we get a training set with 1,508,695 sessions and a testing set with 1,510,663 sessions.

Parameter Setting. In the training and testing data set, only sessions with at least one click are kept, because clicks on ads, query suggestions or other elements are much more likely to appear for the ignored sessions with no clicks [4]. In order to deal with documents that appear in the testing set but do not appear in the training set, we compute *position relevance* [3] for each query. All the experiments were carried out on a PC with 2GB RAM and 3GHz CPU. Model training was done with MATLAB R2008a because we used the optimization toolbox, and testing was done with C language for the sake of time cost.

As aforementioned, it is not easy to calculate the analytical expression of model parameters α, β . In practice, we chose the parameters by the trial-and-error method. An analysis of parameter sensitivity is shown Fig. II. Firstly, we set $\beta = 0$ and change α , as Fig. I(a) shows. α ranges in [0,1], and the interval is 0.1. We find that the average LL reaches maximum when $\alpha = 0.9$. Therefore, we take 0.9 as the optimal value of α . Then we set $\alpha = 0.9$ and change β . Fig. I(b) shows the average LL curves when β changes, the optimal value of β is 0.

¹ Available at <http://labs.blog.sohu.com/103180971.html>

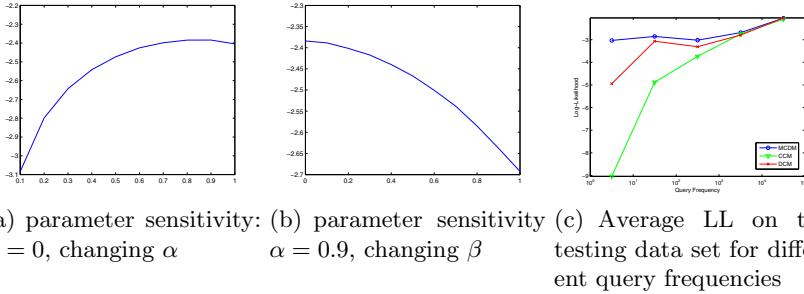


Fig. 1. Parameter sensitivity and the average log-likelihood curves

4.2 Evaluation

Average Log-likelihood. *Log-likelihood* (LL) is a widely used metrics in click model evaluation. A larger LL indicates better model performance, the theoretical optimal value is 0. and Eq.9 shows how to compute the log-likelihood of one session. Given the testing set, we use LL per session (average LL) to measure model performance. The average LL of three models is in Table II. From it, we observe that PCTM outperforms the other two models significantly. The improvement of log-likelihood l_1 over l_2 is computed as $(\exp(l_1 - l_2) - 1) \times 100\%$ [3]. Therefore, the improvement of PCTM over CCM is with 17.48 percentage, and the improvement over DCM is more distinct with increasing 23.17 percentage points. Fig. 1(c) shows LL per session for different query frequencies. We find that PCTM outperforms DCM and CCM obviously on low frequency queries. And on high frequency queries, their performances are almost the same. This is a significant characteristic of PCTM because many existing click models perform well on high frequency queries but not very well on low frequency queries.

Table 1. Comparison of the different models in average LL and perplexity index

Evaluation Index	DCM	CCM	PCTM
Average LL	-2.592459	-2.545192	-2.384101
Perplexity	1.300249	1.270166	1.259565

Click Perplexity. The *perplexity* [2] is defined as : $2^{-\frac{1}{N} \sum_{i=1}^N \log_2 p_i}$. where N is the number of observations in the testing set and p_i is the probability of observation i as predicted by the model. Perplexity measures how "surprised" the model is upon observing i . The lower the value of perplexity is, the better the model performs. The theoretical optimal value of perplexity is 1, and the improvement of perplexity p_1 over p_2 is: $(p_2 - p_1)/(p_2 - 1) \times 100\%$.

For click models, click perplexity [3] can be derived from the original definition of perplexity:

$$p_i = 2^{-\frac{1}{N} \sum_{i=1}^N (C_i^n \log_2 q_i^n + (1-C_i^n) \log_2 (1-q_i^n))} \quad (11)$$

where the query sessions in the testing set are indexed by $n = 1, \dots, N$, and q_i^n denotes the click probability as predicted by the click model on position i of session n , and C_i^n denotes the corresponding click event. The average click perplexities over all query sessions and positions for three models are in Table II. Then, we can calculate the click perplexity improvement of PCTM over DCM: 13.55%, and CCM: 3.92%. Fig. 2(a) is the click perplexity over all positions for different query frequencies, and Fig. 2(b) is over all query sessions for different positions. PCTM performs much better than DCM and CCM on high frequency queries. Model performances in click perplexity are consistent with log-likelihood case.

First and Last Click Predicting Error. We use *Root-mean-square (RMS) error* as evaluation metric of click predicting errors [4], and RMS error of click position prediction can be computed as: $RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (p_n - o_n)^2}$. where, $n = 1, \dots, N$ is the indices of query sessions, and p_n, o_n is the predicted and observed click position.

Fig. 2 is the first and last click prediction error on the testing set. PCTM performs a little worse than DCM and CCM on first click prediction but obviously better on last click prediction. This characteristic of PCTM is also significant because predicting last click position is challenging for click models while first click is relatively easy. Intuitively, We interpret the phenomena as a result of the previous assumptions. PCTM performs better on bottom positions because it takes all the previous clicked documents into consideration.

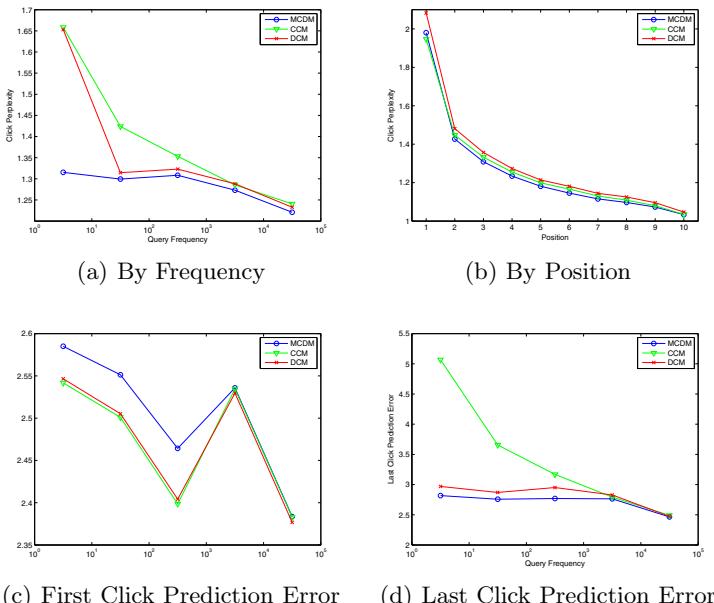


Fig. 2. Click perplexity on testing set and click prediction error

5 Conclusion and Future Work

In a word, the proposed PCTM makes use of the information of previous clicked documents to improve the performance. Experimentally PCTM outperforms the state-of-the-art click models, especially over bottom positions of query sessions. Besides, we also find that PCTM performs well on less frequency queries while other models performs well on high frequency queries but not very well on less frequency queries. In future work, the mechanism behind the model and automatical parameter setting will be studied.

References

1. Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: Proceedings of the International Conference on Web Search and Web Data Mining, pp. 87–94. ACM, New York (2008)
2. Dupret, G.E., Piwowarski, B.: A user browsing model to predict search engine click data from past observations. In: Proceedings of the 31st Annual International ACM SIGIR, pp. 331–338. ACM, New York (2008)
3. Guo, F., Liu, C., Kannan, A., Minka, T., Taylor, M., Wang, Y.M., Faloutsos, C.: Click chain model in web search. In: Proceedings of the 18th International Conference on World Wide Web, pp. 11–20. ACM, New York (2009)
4. Guo, F., Liu, C., Wang, Y.M.: Efficient multiple-click models in web search. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 124–131. ACM, New York (2009)
5. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., Gay, G.: Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Transactions on Information Systems (TOIS) 25(2), 7 (2007)
6. Rho, Y., Gedeon, T.D.: A Link-Click Life-cycle on the Screen. In: Web Technologies and Applications, Proceedings Asia Pacific Web Conference (APWeb 1998), pp. 307–312 (1998)
7. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. In: The 10th IEEE International Conference on Data Mining, p. 530 (2007)
8. Shu, X.Y., Yang, Y.J., Liu, W.H.: Multi-click dependent model to estimate document relevance in web search (2010)
9. Tan, B., Shen, X., Zhai, C.X.: Mining long-term search history to improve search accuracy. In: Proceedings of the 12th ACM SIGKDD, p. 723. ACM, New York (2006)

Intention Extraction from Text Messages

Insu Song and Joachim Diederich

School of Business & Information Technology

James Cook University Australia

{insu.song, joachim.diederich}@jcu.edu.sg

Abstract. Identifying intentions of users plays a crucial role in providing better user services, such as web-search and automated message-handling. There is a significant literature on extracting speakers' intentions and speech acts from spoken words, and this paper proposes a novel approach on extracting intentions from non-spoken words, such as web-search query texts, and text messages. Unlike spoken words, such as in a telephone conversation, text messages often contain longer and more descriptive sentences than conversational speech. In addition, text messages contain a mix of conversational speech and non-conversational contents such as documents.

The experiments describe a first attempt to extracting writers' intentions from Usenet text messages. Messages are segmented into sentences, and then each sentence is converted into a tuple (*performative, proposition*) using a dialogue act classifier. The writers' intentions are then formulated from the tuples using constraints on felicitous human communication.

1 Introduction

1.1 Motivation

In recent years, machine learning techniques have shown significant potential to the practice of automated message processing, in particular dialog act classification for spoken words [5913], but the use of such classifiers on non-spoken words has not been investigated. Almost every user of the Internet is using one or more text messaging services. These include email, Newsgroups, and instant messaging. In 1999, an estimated 3 billion email messages were sent every day in the USA [6]. The study by Kraut *et al.* [7] showed that interpersonal communication is a stronger driver of Internet use than are information and entertainment applications. That is, text messaging services are virtually everywhere and constantly demanding our attention as messages arrive. This ubiquity and the accelerating growth in the number of text messages make it important that we develop automated text message processing and text message based query processing in general.

However, conversational speeches (e.g. telephone conversation) are different from electronic text messages (e.g. email, Newsgroups articles). The main difference is that the speaker turns (i.e. who is talking) are explicitly specified by the non-text field (i.e. sender) of the messages. Therefore, there is no need to identify the speaker turns. Another difference is that text messages contain longer and more descriptive sentences than conversational speech. That is, a text message is a mix of conversational speech

and normal text documents. This paper proposes a method of extracting intentions from online text messages, such as web query strings and instance messages. The experiments describe a first attempt to extract intended acts from Usenet newsgroup messages. Dialog-act classifiers were constructed to extract dialog-acts, and then senders' intentions were deduced based on intentional speech act theories [2][14][10]. Predicate forms (verb-object-subject) of each dialog-act units together with the extracted dialog-acts were then used to represent a set of intended acts (speech acts) of message senders.

1.2 Background

This work is based on text mining and in particular text classification, dialog-act classification, and intention based theory of speech acts. The following section provides a brief overview of the core techniques, focussing on dialog act classification and speech act theory.

Dialog Act Classification. Traditionally syntactic and semantic processing approaches have been used for dialogue act classification, but they are error prone [9] and require intensive human effort in defining linguistic structures and developing grammar [13]. Recently, many statistical understanding approaches for dialog act classification have been proposed to overcome these problems [5][9][13]. All of these approaches use unigram Naive Bayesian classifiers, except [13], which uses the ngram model with dynamic programming algorithm. Another advantage of the probabilistic approaches over the traditional syntactic and semantic processing approaches is that no expensive and error-prone deep processing is involved.

Intention Based Theory of Speech Acts. In 1990, Cohen and Levesque [2] developed a theory in which speech acts were modelled as actions performed by rational agents in the furtherance of their intentions, based on their intentional logic [1]. Unlike their previous plan-based speech act theory, this theory is rooted in a more general theory of rational action [14]. The foundation upon which they built this model of rational action was their theory of intention [1]. The theory was formalized in modal logic using their logic of intention.

Although semantics of intentional logic itself were given in possible worlds semantics in [18], it was Singh [10] who proposed semantics of speech acts, i.e., their conditions of satisfaction. He proposed several useful constraints on communication. One of the interesting constraints proposed is the normative constraints for felicitous communication. An example is shown below:

$$\text{comm}(x,y,\langle \text{directive}, p \rangle) \rightarrow \text{intends}(x,p)$$

This says that if x directs y to achieve p , x intends p . He proposes six such constraints arguing that these constraints restrict the possible models to the agents that we can assume to be rational. That is, if an agent is rational, the agent saying a directive speech act must be intending what the agent is saying. We develop an intention extraction function for message handling assistants based on these constraints in Section 3.2.

2 Experimental Evaluation

2.1 Methodology

A preliminary study has been undertaken on extracting intentions from online text messages. Three sets of online text messages were obtained from Usenet news groups. Each data set comprises of DA-annotated (dialog-act annotated) units, called dialog-act units (DUs), as shown in Figure 1. Dialog-act units (DUs) in the data set A are either a question for information or a yes-no question. The question dialog-act units are collected from various Usenet FAQ forums. Dialog-act units in data set B are collected from the same Usenet discussion group. Dialog-act units in data set C are collected from another Usenet discussion group (comp.ai.neural-nets) that is different from the data set B. The unknown words for the data sets A, B, and C in the test sets were 19.9%, 14.8%, and 33.5%, respectively.

The resulting dialog-act units are represented as attribute-value vectors (“bag of words” representation) where each distinct word corresponds to a feature whose value is the frequency of the word in the text sample. Values were transformed with regard to the length of the sample. Some functional words (e.g., the and a) were removed and stemming were performed on each extracted dialog-act unit. In summary, input vectors for machine learning consist of attributes (the words used in the sample) and values (the transformed frequency of the words). Outputs are dialog-acts of each dialog-act units, that is, multiclass decision tasks were learned.

All the dialog units in the data sets were labelled with 21 dialog acts. Among 21 dialog acts, we used only 11 dialog acts because other dialog acts are not useful for practical uses, such as message handling. For example, dialog acts that are realized with some fixed phrases like THANK or GREETING do not provide any essential information in message handling and trivial in recognizing them. Compound sentences and unreadable sentences that couldn’t be automatically segmented are tagged as NOTAG and not included in the data set. The assertive dialog act is further classified into four dialog acts to test classification of the combinations of performatives and concepts. The chosen dialog acts for our experiments are listed on the left column in Table 2. The types of the dialog acts and example dialogs are also shown.

A total of four classifiers were implemented. Two of them were based on Naive Bayesian approaches: multivariate-Bernoulli (NB-B) and multinomial (NB-M). The other two were based on the neural network approach: single layer perceptron classifier (SLP) and multi-layer perceptron classifier (MLP). Only the MLP classifier was a non-linear classifier. All other classifiers were linear classifiers.

Data set	DU # in training set (Word #)	DU # in test set (Word #)	Unknown %
A	233 (2069)	124 (1177)	19.9
B	456 (3938)	237 (2042)	14.8
C	.	92 (877)	33.5

Fig. 1. The number of dialog units and the number of words in each data set

Dialog Act	Type of Illocutionary Force	Example
0: DECLARATIVE	DECLARATIVE	I name this door the Golden Gate.
1: COMMISSIVE	COMMISSIVE	I promise to check the price.
2: PERMISSIVE	PERMISSIVE	You can download files from the server.
3: PROHIBITIVE	PROHIBITIVE	You are not allowed to log in.
4: DIR_QUESTION	Directive	How does it cost?
5: DIR_ACTION	Directive	Could you open the door?
6: A_FACT	Assertive	It will work via IR.
7: A_SELF_SITUA	Assertive	I bought mine on Thursday.
8: A_PREFERENCE	Assertive	I have liked it so far.
9: A_OPINION	Assertive	I don't see that ever changing.
10: CONDITION	-	When I was connected into this TAP

Fig. 2. Dialog acts for the experiment

Feature set	% Correct								Hidden Unit #	
	NB-B		NB-M		SLP		MLP			
	Train	Test	Train	Test	Train	Test	Train	Test		
A	91	90	92	91	99	94	94	83	1	
B	68	59	82	62	93	57	72	48	4	
C		51		56		66		44	4	

Fig. 3. Performance comparison

Unlike Naive Bayes classifiers, the neural network classifiers (i.e. SLP and MLP) have a few parameters that have to be set manually from some empirical study. We found that both SLP and MLP classifiers converge reasonably within 1000 iterations. We set both the learning rate η , the momentum α to 0.1 in all our tests.

The number of hidden units used for each data set is shown in Figure 3. For neural network classifiers, each test was conducted ten times and the results were averaged.

2.2 Results

In Figure 3 accuracy of the classifiers are shown. Test results of neural network classifiers are averages over ten tests. As we can see from Figure 3, in overall, the SLP classifier outperformed all other classifiers in accuracy. The highest accuracy in each feature set is highlighted.

The SLP classifier recognized 60 dialog acts correctly from 91 dialog units in data set C, i.e., 66%. Even though the data set C was collected from a different site that is independent from the training data set B, the test result was better than the result of [12] which claims accuracy of 58% on performative classification and 39.7% in dialog act classification.

Figure 4 shows the recall (rec) and precision (prec) measures for dialog acts for the feature set B. As we can see from the table, the SLP classifier again performs the best having an average recall of 42%, with 48% precision. The NB-M classifier comes to the second having an average recall rate of 38%, with 42% precision. However, unlike the accuracy measure, the MLP classifier had higher recall and precision rate.

Dialog Act	DU		NB-B		NB-M		SLP		MLP	
	#	rec	prec	rec	prec	rec	prec	rec	prec	
0: DECLARATIVE	7	0	0	0	0	0	0	0	0	0
1: COMMISSIVE	13	0	0	31	44	23	27	23	27	
2: PERMISSIVE	2	0	0	0	0	50	100	50	100	
3: PROHIBITIVE	1	0	0	100	100	100	100	0	0	
4: DIR_QUESTION	33	51	68	63	68	58	70	48	47	
5: DIR_ACTION	10	0	0	10	25	0	0	0	0	
6: A_FACT	67	99	56	85	63	76	59	69	60	
7: A_SELF_SITUA	64	86	63	88	64	73	65	59	62	
8: A_PREFERENCE	5	0	0	0	0	0	0	0	0	
9: A_OPINION	20	5	33	10	33	20	33	15	16	
10: CONDITION	14	7	50	29	67	64	69	42	32	
Total/Average	236	23	25	38	42	42	48	28	31	

Fig. 4. Recall and precision rates for all dialog acts

Labeled	Assigned										Total	
	0	1	2	3	4	5	6	7	8	9	10	
0: DECLARATIVE	(0)	0	0	0	1	0	3	3	0	0	0	7
1: COMMISSIVE	0	(4)	0	0	0	0	0	8	0	0	1	13
2: PERMISSIVE	0	0	(0)	0	1	1	0	0	0	0	0	2
3: PROHIBITIVE	0	0	0	(1)	0	0	0	0	0	0	0	1
4: DIR_QUESTION	0	0	0	0	(21)	0	9	2	0	1	0	33
5: DIR_ACTION	0	0	0	0	1	(1)	6	2	0	0	0	10
6: A_FACT	0	0	0	0	2	1	(57)	5	0	2	0	67
7: A_SELF_SITUA	0	1	0	0	0	0	7	(56)	0	0	0	64
8: A_PREFERENCE	0	0	0	0	1	0	0	4	(0)	0	0	5
9: A_OPINION	0	4	0	0	2	1	6	4	0	(2)	1	20
10: CONDITION	0	0	0	0	2	0	3	4	0	1	(4)	14
Total	0	9	0	1	31	4	91	88	0	6	6	236

Fig. 5. Confusion matrix for dialog acts (Multinomial Naive-Bayes on data set B)

Figure 5 shows a confusion matrix for the classification experiment using the multinomial Naive Bayes classifier on the feature set B. The tables show that the classifiers tends to assign A_FACT (assert facts) to directive dialog units and A_SELF_SITUA (assert self situation) to commissive dialog units. An_OPINION dialog units are equally misclassified as either commissive or assertive. A_SELF_SITUA and A_FACT tends mob up dialog units of other classes. This tendency was so severe in the NB-B classifier that most of dialog units belonging to infrequent classes are classified either as A_SELF_SITUA or A_FACT.

3 Extracting Intentions from Messages

3.1 Intention Extraction Function: I_{ext}

We describe an intension extract function defined in [11]. The intention extraction function, I_{ext} , maps human communication messages (e.g., emails, SMS messages) to a set of intentions. It defines a relationship between messages and sender's intentions. Let

us suppose that each sentence in a message contains only one performative verb, that is, we do not consider compound sentences that contain several performatives. Then, since all sentences in a message could be put in the form of performatives by using appropriate performative verbs [10], a sentence can be written as a pair $\langle a, p \rangle$, where a is a performative (e.g., a dialog-act of a DU extracted using a dialog-act classifier) and p is a proposition (e.g., a DU in a form of verb-object-subject). A message can contain several sentences. Thus, a message, m , can be represented as an ordered set of $\langle a_i, p_i \rangle$, i.e., $m = (\langle a_1, p_1 \rangle, \dots, \langle a_n, p_n \rangle)$, where n is the number of sentences in the message. If we assume that messages are independent, ignoring the order of sentences, we can then define an intention extraction function $Iext$ as follows:

$$Iext : M \rightarrow 2^I,$$

where M is the set of all messages and I is the set of all intentions.

3.2 Intentions from Speech Acts

We now describe an implementation of the intention extraction function $Iext$, which combines a dialog-act classifier for online text messages and a mapping function that deduces specific intentions based on the extracted dialog-acts and constraints felicitous communication proposed by Singh [10]. Figure 6 shows an algorithm of the intention extraction function, $Iext$, which now will be discussed in detail below.

Speech act theory is an attempt to build a logical theory that would explain dialogue phenomena in terms of the participants' mental states [3]. The theory models how a speaker's mental state leads to communicative actions on what assumptions the speaker makes about the hearers.

To clarify terms used in speech act theory, let us consider an example of a speech act: "Please shut the door." This sentence can be rewritten as "I request you to shut the door." Here, the performative verb is "request" and the proposition is "shut the door". We can classify the speech act of the sentence as *directive* illocutionary force. In this paper, we adapt the classification made by Singh [10].

Using the formal speech act theory developed by Cohen and Levesque [2][3][4], we can deduce speakers' intentions from their speech acts. For example, upon receiving a sentence containing a request speech-act and a proposition $prop$, we can deduce the following mental states of the speaker using the definition of the request speech-act in [2]:

$$(Bel \text{ speaker } \neg p \wedge \neg q) \quad (1)$$

$$(Int \text{ speaker } p) \quad (2)$$

$$(Des \text{ speaker } \diamond q) \quad (3)$$

where q is $\exists e(Done \text{ hearer } e; prop?)$, and p is a mutual belief between *speaker* and *hearer* about certain conditions for the speech act to be appropriate. The intuition is that when certain conditions are met (e.g. *hearer* is cooperative), *hearer* will do an action e to bring about $prop$.

We now propose a simple method that can acquire a sender's intentions for an message handling agent (MA) by adapting normative constraints for felicitous communication proposed by Singh [10]. First, we introduce some notions to define an illocutionary force extractor function, $Fext$. Let $A = F \times Prop$ be the set of all speech acts, where F is the set of all illocutionary forces and $Prop$ is the set of all propositions. If $a \in A$ is a speech act, it is defined as $a = \langle i, p \rangle$, where i is an illocutionary force and p is a proposition. Then, the illocutionary force extractor function, $Fext$, is defined as:

$$Fext : S \rightarrow A, \quad (4)$$

where S is the set of all sentences. This function is implemented using a dialog-act classifier developed in the previous sections.

In Section 1.2, we discussed briefly the semantics of speech acts that Singh [10] proposed. His semantics of speech acts are for a speaker and a hearer, but we need to find an interpretation for an MA who reads a message received for its user from a sender. The following is an interpretation of the types of illocutionary forces for an MA.

If we have the type of the speech act identified for a sentence, we can reason what the sender's intention is for saying the sentence. That is, if the illocutionary force of a sentence is directive (e.g. I request you p), the intention of the sender is to make the user intend p . If it is assertive (e.g. I inform you p), the intention is to make the user believe p . If it is commissive (e.g. I promise you p), the intention is to make the user believe that the sender intends p , so it will be true in some future time. If it is permissive, the intention is to make the user believe that the user is not obliged to the sender not to intend. If it is prohibitive, the intention is to make the user believe that the user is obliged to the sender not to intend p . From this, we can formulate an algorithm for the intention extraction function, $Iext$, shown in (Figure 6). Formulas (1), (2), (3), and (4) can be used to deduce the sender's mental states from the intentions extracted.

1	Set $I := \{ \}$;
2	Set $x := \text{Sender}(\text{message})$;
3	For every sentence s in message;
4	$\langle i, p \rangle \leftarrow Fext(s)$;
5	If i is directive, add $(Int\ x\ (Int\ user\ p))$ to I ;
6	If i is assertive, add $(Int\ x\ (Bel\ user\ p))$ to I ;
7	If i is declarative, add $(Int\ x\ (Bel\ user\ p))$ to I ;
8	If i is commissive, add $(Int\ x\ (Bel\ user\ (Int\ x\ p)))$ to I ;
9	If i is permissive, add $(Int\ x\ E(\Diamond(Int\ user\ p)))$ to I ;
10	If i is prohibitive, add $(Int\ x\ A(\Box\neg(Int\ user\ p)))$ to I ;
12	Return I ;

Fig. 6. Intention extraction function: $Iext$

4 Discussion and Future Work

This is the first report of a novel approach to extracting intentions from non-spoken online text messages that combines classifiers and semantic processing, in particular a speech-act

mapping model for online text messages. Considerable further research is required. The approach of extracting intentions using machine learning in processing online-messages has the potential to improve query processing in Web-search domains. This approach of extracting intentions using dialog-act classifiers for online-messages and constraints on felicitous communication can be further expanded by using alternative feature representations of text data sets, such as concept terms or semantic terms. There is massive potential of incorporating these sophisticated information extraction technologies within automated message processing and in query processing more generally.

References

1. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. *Artificial Intelligence*, AI 42(2–3), 213–261 (1990)
2. Cohen, P.R., Levesque, H.J.: Performatives in a rationally based speech act theory. In: Meeting of the Association for Computational Linguistics, pp. 79–88 (1990)
3. Cohen, P.R., Levesque, H.J.: Rational interaction as the basis for communication. In: *Intentions in Communication*, pp. 221–255. MIT Press, Cambridge (1990)
4. Cohen, P.R., Levesque, H.J.: Communicative actions for artificial agents. In: Proc. ICMAS 1995, San Francisco, CA, USA, pp. 65–72. The MIT Press, Cambridge (1995)
5. Garner, P., Browning, S., Moore, R., Russell, M.: A theory of word frequencies and its application to dialogue move recognition. In: Proc. ICSLP 1996, Philadelphia, PA, vol. 3, pp. 1880–1883 (October 1996)
6. Gwizdka, J.: Reinventing the inbox: supporting the management of pending tasks in email. In: CHI 2002 Extended Abstracts on Human Factors in Computer Systems, pp. 550–551. ACM Press, New York (2002)
7. Kraut, R., Mukhopadhyay, T., Szczypula, J., Kiesler, S., Scherlis, W.: Communication and information: alternative uses of the internet in households. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 368–375. ACM Press/Addison-Wesley Publishing Co (1998)
8. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR 1991), pp. 473–484. Morgan Kaufmann publishers Inc, San Mateo (1991)
9. Reithinger, N., Klesen, M.: Dialogue act classification using language models. In: Proc. Eurospeech 1997, Rhodes, Greece, pp. 2235–2238 (September 1997)
10. Singh, M.P.: A semantics for speech acts. In: Readings in Agents, pp. 458–470. Morgan Kaufmann, San Francisco (1997); Reprinted from Annals of Mathematics and Artificial Intelligence (1993)
11. Song, I., Piggott, P.: Modelling message handling system. In: Gedeon, T.(T.) D., Fung, L.C.C. (eds.) AI 2003. LNCS (LNAI), vol. 2903, pp. 53–64. Springer, Heidelberg (2003)
12. Toshiaki, F., Koll, D., Waibel, A.: Probabilistic dialogue act extraction for concept based multilingual translation systems. In: Proc. ICSLP 1998, Sydney, Australia, vol. 6, pp. 2771–2774 (1998)
13. Wang, Y.-Y., Waibel, A.: Statistical analysis of dialogue structure. In: Proc. Eurospeech 1997, Rhodes, Greece, pp. 2703–2706 (September 1997)
14. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley Sons, Chichester (2001)

m-SNE: Multiview Stochastic Neighbor Embedding

Bo Xie¹, Yang Mu², and Dacheng Tao³

¹ School of Computer Engineering, Nanyang Technological University

² School of Computer Engineering, Nanyang Technological University Mingli Song
College of Computer Science, Zhejiang University

³ Centre for Quantum computation & Intelligent Systems Faculty of Engineering and
Information Technology University of Technology, Sydney

Abstract. In many real world applications, different features (or multiview data) can be obtained and how to duly utilize them in dimension reduction is a challenge. Simply concatenating them into a long vector is not appropriate because each view has its specific statistical property and physical interpretation. In this paper, we propose a multiview stochastic neighbor embedding (m-SNE) that systematically integrates heterogeneous features into a unified representation for subsequent processing based on a probabilistic framework. Compared with conventional strategies, our approach can automatically learn a combination coefficient for each view adapted to its contribution to the data embedding. Also, our algorithm for learning the combination coefficient converges at a rate of $O(1/k^2)$, which is the optimal rate for smooth problems. Experiments on synthetic and real datasets suggest the effectiveness and robustness of m-SNE for data visualization and image retrieval.

Keywords: multiview learning, dimension reduction, stochastic neighbor embedding, image retrieval.

1 Introduction

In many computer vision and information retrieval applications, data often lie in a high dimensional space. Direct manipulation in this space is difficult because of the so called “curse of dimensionality.” Many algorithms have been proposed to find a low dimensional embedding of the original high dimensional data [1][2][3][4][5][6]. For example, principal component analysis (PCA) [7] finds an orthogonal subspace to encode the data variance. Popular nonlinear embedding methods, such as ISOMAP [8], Laplacian eigenmaps [9], local linear embedding (LLE) [10][11] and stochastic neighbor embedding (SNE) [12], preserve pairwise distances in the low dimensional space by considering different types of geometric properties.

However, these approaches are only applicable to single view data while in many real world scenarios, different features are present and complementary to each other. One way of incorporating multiview data is simply concatenating them into a long vector and applying the conventional dimension reduction

techniques. This strategy has three major problems: 1) different statistical properties are not duly considered; 2) the complementary information of different features is not well explored; and 3) the performance of concatenation will easily deteriorate, if one or more views are corrupted by noise.

We propose to unify different features under a probabilistic framework. Under each view, we can construct a probability distribution from pairwise distances as that used in SNE. Unifying heterogeneous features in such a probabilistic strategy is meaningful because data with different properties, e.g. mean and variance, can be appropriately combined in a probability space. Moreover, our algorithm can automatically learn a weighting for each view, therefore discriminative features are promoted and random noise is suppressed. At last, our algorithm for learning the weighting converges at the optimal rate of $O(1/k^2)$, where k is the number of iterations, by adopting Nesterov's accelerated first order method [13].

2 Stochastic Neighbor Embedding and Its Variants

The main idea of SNE is to construct probability distributions from pairwise distances wherein larger distances correspond to smaller probabilities and vice versa. Then, low dimensional embedding is obtained by minimizing the Kullback Leibler (KL) divergence of the two probability distributions.

The t-distributed stochastic neighbor embedding (t-SNE) [14] accelerates SNE by replacing the original Gaussian distribution with a t-distribution, thus avoiding costly exponential calculation. Here, we base our algorithm on t-SNE for speed benefits.

Formally, suppose we have high dimensional data points $\{x_i \in \mathbb{R}^d\}_{i=1}^n$. And the normalized pairwise distances are represented in a symmetric matrix $P \in \mathbb{R}_+^{n \times n}$, where $p_{ii} = 0$ and $\sum_{i,j} p_{ij} = 1$. Similarly, in the low dimensional embedding, we define the probability distribution Q , with each element

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}, \quad (1)$$

where $y_i \in \mathbb{R}^r$ is the low dimensional data corresponding to x_i .

To find the optimal low dimensional embedding, the KL divergence between the two distributions over all data points

$$f = KL(P|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (2)$$

is minimized.

The objective function is not convex and gradient descent can be used to find a local solution. The gradient with respect to a low dimensional data point is

$$\frac{\partial f}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) (1 + \|y_i - y_j\|^2)^{-1}. \quad (3)$$

Besides the basic gradient descent, a number of strategies are employed to improve optimization [14].

3 Multiview Stochastic Neighbor Embedding

In this section, we generalize stochastic neighbor embedding to handle multiview data. The probabilistic nature of SNE offers a clean formulation for combining different features. In optimization, we alternately solve SNE and learn combination coefficients, i.e. weightings for different views. Due to space limitation, the detailed proof of following results is given in our technical report [15].

3.1 Multiview Dimension Reduction

We assume the final probability distribution on the high dimensional space is a convex combination of all the different views, i.e.

$$p_{ij} = \sum_{t=1}^v \alpha^t p_{ij}^t, \quad (4)$$

where α^t is the combination coefficient for view t and p_{ij}^t is the probability distribution under view t . The coefficient vector $\alpha = [\alpha^1, \dots, \alpha^v]^T$ lies on a simplex in \mathbb{R}^v , denoted as $\alpha \in \Delta^v$. Obviously, p_{ij} is a probability distribution since $\sum_{i \neq j} p_{ij} = \sum_t \alpha^t \sum_{i \neq j} p_{ij}^t = \sum_t \alpha^t = 1$.

The optimization is then over both y_i and α . We adopt alternating optimization to solve this problem. In every round, we first fix α and use t-SNE to find low dimensional embedding. After that, we fix y_i and optimize over α .

One caveat with this approach is that all the coefficients will concentrate on a single view that performs the best while the contributions of other views vanish. We tackle this problem by adding an l_2 norm regularization term to balance the coefficients over all views. The new objective function for learning optimal combination coefficients is

$$g(\alpha) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} + \lambda \|\alpha\|^2, \quad (5)$$

where λ is a trade-off coefficient.

Fortunately, this problem is convex and can be solved with convergence rate of $O(1/k^2)$, where k is the number of iterations.

4 Nesterov's Accelerated First Order Method

Recently, Nesterov [13][16] proposed an accelerated first order method for convex optimization problems that converges at a rate of $O(1/k^2)$, where k is the number of iterations. We adopt this method to optimize α with fixed y_i .

Nesterov's method requires that the gradient of objective function is Lipschitz-continuous, i.e. $\|\nabla g(\alpha_1) - \nabla g(\alpha_2)\| \leq L_g \|\alpha_1 - \alpha_2\|$. This condition is equivalent to

$$g(\alpha_1) \leq g(\alpha_2) + \langle \nabla g(\alpha_2), \alpha_1 - \alpha_2 \rangle + \frac{L_g}{2} \|\alpha_1 - \alpha_2\|^2, \quad (6)$$

where $\alpha_1 \in \Delta^v$ and $\alpha_2 \in \Delta^v$ are two arbitrary data points and L_g is the Lipschitz constant of $g(\alpha)$.

Probabilities between different high dimensional samples are calculated as in [4] and are strictly positive. Thus, the Hessian of $g(\alpha)$ is bounded and the Lipschitz constant for $\nabla g(\alpha)$ exists [5].

4.1 Accelerated First Order Method

In every iteration round, we construct an estimate function

$$\psi_k(\alpha) = \frac{1}{2}\|\alpha - \alpha_0\|^2 + \sum_{i=0}^k b_i[g(\alpha_i) + \langle \nabla g(\alpha_i), \alpha - \alpha_i \rangle], \quad k \geq 0, \quad (7)$$

where $\alpha_0 \in \Delta^v$ is the initial guess point and b_i is some scaling coefficient.

Also, we construct a second order approximation to the original function

$$m_L(y; \alpha) = g(y) + \langle \nabla g(y), \alpha - y \rangle + \frac{L}{2}\|\alpha - y\|^2, \quad (8)$$

$$T_L(y) = \underset{\alpha \in \Delta^v}{\operatorname{argmin}} m_L(y; \alpha), \quad (9)$$

where y is any point and L is a positive constant.

Intuitively, the first function captures the history information about the objective function and the second function is related to the gradient information of the current iteration. The two optimization problems have quadratic objective functions with linear constraints and can be conveniently solved using standard optimization toolkit.

We design the algorithm to maintain the following two relations in every iteration round.

$$\mathcal{R}_k^1 : B_k g(\alpha_k) \leq \psi_k^* \equiv \min_{\alpha} \psi_k(\alpha); \quad (10)$$

$$\mathcal{R}_k^2 : \psi_k(\alpha) \leq B_k g(\alpha) + \frac{1}{2}\|\alpha - \alpha_0\|^2, \quad (11)$$

where $k \geq 0$ and $B_0 = 0$ and $B_k = B_{k-1} + b_k$.

These two relations lead to the following convergence rate of the minimizing sequence:

$$g(\alpha_k) - g(\alpha^*) \leq \frac{\|\alpha^* - \alpha_0\|^2}{2B_k}, \quad (12)$$

where $k \geq 1$ and α^* is the optimal solution to the problem.

In our algorithm, the stopping criterion is

$$\langle \nabla g(T_L(y)), y - T_L(y) \rangle \leq \frac{1}{L}\|\nabla g(T_L(y))\|^2. \quad (17)$$

As in [6], this criterion is satisfied for any $L \leq L_g$. Details are given in [5].

This varying step size approach avoids costly computation of the Lipschitz constant and can adjust the step size according to local variation.

Algorithm 1. Accelerated First Order Method for Combination Coefficient

Initialize $B_0 = 0, \alpha_0, L_0 > 0, \psi_0(\alpha) = \frac{1}{2}\|\alpha - \alpha_0\|^2$

Iterate:

1. Set $L = L_k$.
2. Repeat
 - (a) Find b by solving the equation

$$Lb^2 = 2(B_k + b) \quad (13)$$

- (b) Set $z_k = \operatorname{argmin}_\alpha \psi_k(\alpha)$ and

$$y = \frac{B_k \alpha_k + bz_k}{B_k + b} \quad (14)$$

- (c) If $\langle \nabla g(T_L(y)), y - T_L(y) \rangle \leq \frac{1}{L}\|\nabla g(T_L(y))\|^2$

Then $L := \gamma L$

Else Break

3. Set $y_k := y, M_k := L, b_{k+1} := b, L_{k+1} := L$ and update

$$\alpha_{k+1} = T_{M_k}(y_k) \quad (15)$$

$$\psi_{k+1}(\alpha) = \psi_k(\alpha) + b_{k+1}[g(\alpha_{k+1}) + \langle \nabla g(\alpha_{k+1}), \alpha - \alpha_{k+1} \rangle] \quad (16)$$

4.2 Convergence Analysis

We first show that the sequences $\{\alpha_k\}$, $\{y_k\}$, $\{z_k\}$ and $\{b_k\}$ generated by Algorithm 1 satisfy the two relations in every iteration round. We then show the coefficients $\{B_k\}$ is $O(k^2)$. Combining these results with Eq. 12 we show that the convergence rate of our algorithm is $O(1/k^2)$.

Lemma 1. *The sequences $\{\alpha_k\}$, $\{B_k\}$ and $\{\psi_k\}$ generated by Algorithm 1 satisfy relations \mathcal{R}_k^1 and \mathcal{R}_k^2 for all $k \geq 0$.*

The proof employs similar strategies in [16]. For details, please refer to [15].

Next, we show in the following lemma that the scaling coefficients $\{B_k\}$ grow at a quadratic order. The proof of this lemma is in [16].

Lemma 2. *The scaling coefficients grow as follows*

$$B_k \geq \frac{k^2}{2\gamma L_g}. \quad (18)$$

Finally, by combining Eq. 12 and Eq. 18, we arrive at the following theorem.

Theorem 1. *Let $\nabla g(\alpha)$ be Lipschitz continuous with constant L_g and $0 \leq L_0 \leq L_g$. And the rate of convergence of Algorithm 1 is given as follows*

$$g(\alpha_k) - g(\alpha^*) \leq \frac{\gamma L_g \|\alpha^* - \alpha_0\|^2}{k^2}, \quad k \geq 0. \quad (19)$$

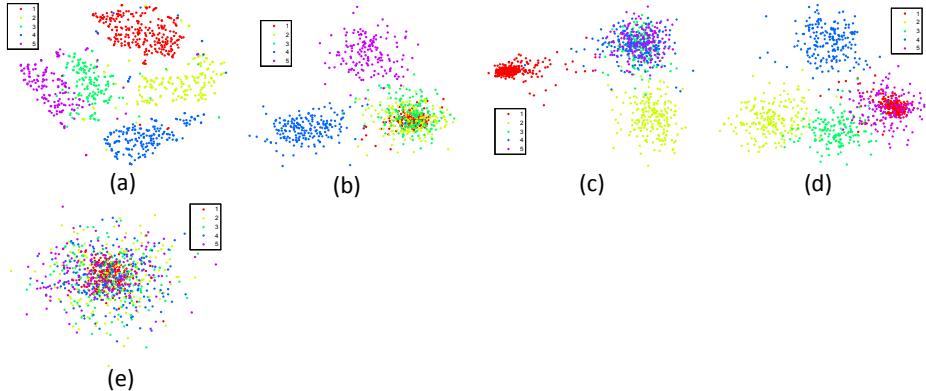


Fig. 1. Toy dataset. a) USPS digits 1-5; b) View 1, digits 1, 2 and 3 mixed; c) View 2, digits, 3, 4 and 5 mixed; d) View 3, digits 1 and 5 mixed; e) View 4, digits randomly mixed.

5 Experiment

In this section, we first demonstrate the effectiveness of the proposed algorithm on a toy dataset. Afterwards, we present experimental results on real datasets in image retrieval, object categorization and scene recognition.

5.1 Toy Dataset

We have designed a toy dataset by using a subset of the USPS digits [17] (digits 1 through 5). We generated four different views from these data by mixing some classes via Fischer Linear Discriminative Analysis (FLDA) [18]. In detail, in the first view, digit classes 1, 2 and 3 were considered as the same; for the second view, classes 3, 4 and 5; and for the third view, classes 1 and 5. Then, FLDA was performed with the new labels to map the original 256-D data into a 2-D space. The fourth view was generated similarly but class labels were random. Thus, this view is considered as noise. In Fig. 1 we plot the original data (by t-SNE) and the four generated views.

We then applied our m-SNE algorithm to combine the four views and embedded the data into a 2-D space for visualization. For comparison, t-SNE by concatenating multiple views (CAT) and distributed approach to spectral clustering (DSE) [19] were also conducted. We tested DSE with two kinds of low dimensional embedding: 1) t-SNE (DSE/t-SNE) and 2) Laplacian eigenmap (DSE/Lap). Experimental results are shown in Fig. 2

From the results, we can see that m-SNE correctly integrated the four views and produced a good low dimensional embedding. The coefficient α also depicts different importance of these views: the third view has the largest weighting since it is least distorted and the fourth view, which is random noise, is depressed in weighting.

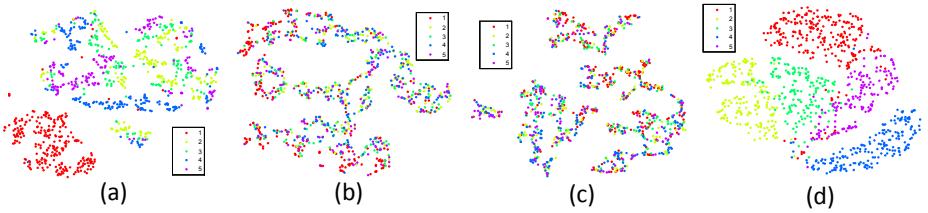


Fig. 2. Results by different algorithms on the toy dataset. a) CAT; b) DSE/t-SNE; c) DSE/Lap; d) m-SNE ($\alpha = [0.291, 0.201, 0.393, 0.115]$).

5.2 Image Retrieval

We conducted image retrieval experiment on the NUS WIDE dataset [20]. The dataset contains 269,648 images with a total of 81 concepts. Six low level features are provided for these images, including 500-D bag of words based on SIFT descriptions, 64-D color histogram, 225-D block-wise color moments, 144-D color correlogram, 73-D edge direction histogram and 128-D wavelet texture.

We randomly sampled 10 query images per concept, with each query retrieving about a hundred images for each view by using indexing. Thus, each query has six to seven hundred retrieval candidates. Then, we performed m-SNE on each query image and its retrieval candidates to exploit the six different features. Finally, we ranked those candidate images by their Euclidean distances to the query image in the learned low dimensional embedding.

For comparison, we also conducted experiments by using Euclidean distances from a single view, Euclidean distances from concatenated features and DSE. Due to limitation of space, precision results of 8 concepts for best single view

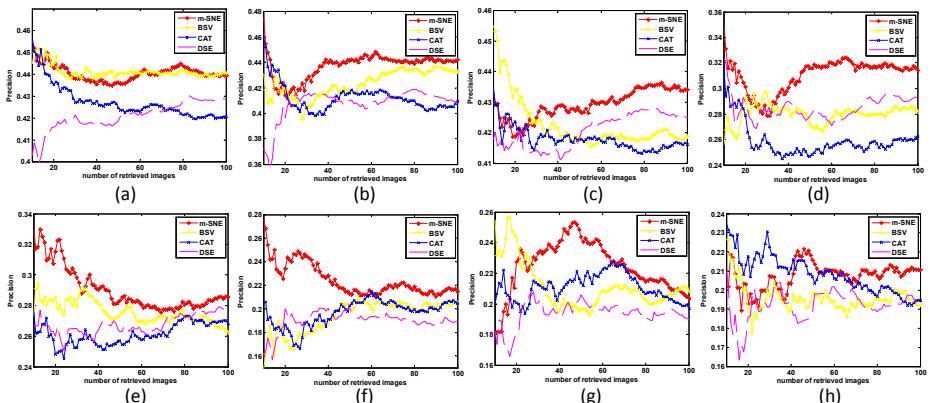


Fig. 3. Retrieval results on NUS WIDE dataset. Precision results for m-SNE, best single view (BSV), concatenated features (CAT) and DSE are shown for 12 concepts. (a) sky; (b) reflection; (c) water; (d) boats; (e) plants; (f) surf; (g) fox; (h) whales.

(BSV), concatenated features (CAT), DSE and m-SNE are plotted in Fig. 3. We can see that m-SNE performs better than or comparable to others.

6 Conclusion

In this paper, we have proposed multiview stochastic neighbor embedding for learning low dimensional data from multiview high dimensional data. Compared with traditional methods, the algorithm operates on a probabilistic framework that meaningfully integrates different views. Our approach can automatically learn a combination coefficient of different views according to their contributions to the final embedding. Thus, this combination coefficient can exploit complementary information in different views and suppress noise at the same time. In optimizing over combination coefficients, we employ Nesterov's gradient accelerating scheme and achieve converge rate of $O(1/k^2)$.

Acknowledgement

This work was supported by the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1006), Zhejiang University.

References

1. Xia, T., Tao, D., Mei, T., Zhang, Y.: Multiview spectral embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* (2010)
2. Zhou, T., Tao, D., Wu, X.: Manifold elastic net: a unified framework for sparse dimension reduction. *Data Mining and Knowledge Discovery*, 1–32 (2010)
3. Si, S., Tao, D., Chan, K.P.: Evolutionary cross-domain discriminative hessian eigenmaps. *IEEE Transactions on Image Processing* 19(4), 1075–1086 (2010)
4. Si, S., Tao, D., Geng, B.: Bregman divergence based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering* 22(7), 929–942 (2010)
5. 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)
6. Zhang, T., Tao, D., Li, X., Yang, J.: Patch alignment for dimensionality reduction. *IEEE Transactions on Knowledge and Data Engineering* 21(9), 1299–1313 (2009)
7. Jolliffe, I.T.: *Principal Component Analysis*. Springer, New York (2002)
8. Tenenbaum, J.B., Silva, V., Langford, J.C.: A global geometric framework for non-linear dimensionality reduction. *Science* 290(5500), 2319–2323 (2000)
9. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *Advances in Neural Information Processing Systems*, vol. 14, pp. 585–591. MIT Press, Cambridge (2001)
10. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500), 2323–2326 (2000)
11. Saxena, A., Gupta, A., Mukerjee, A.: Non-linear dimensionality reduction by locally linear isomaps. In: *Neural Information Processing 2004*, pp. 1038–1043 (2004)

12. Hinton, G., Roweis, S.: Stochastic neighbor embedding. *Advances in Neural Information Processing Systems* 15, 833–840
13. Nesterov, Y.: Smooth minimization of non-smooth functions. *Math. Program.* 103(1), 127–152 (2005)
14. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *Journal of Machine Learning Research* 9, 2579–2605 (2008)
15. Xie, B., Mu, Y., Tao, D.: Multiview stochastic neighbor embedding. Technical report, Nanyang Technological University (2010)
16. Nesterov, Y.: Gradient methods for minimizing composite objective function. Technical report (2007)
17. Hull, J.J.: A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.* 16(5), 550–554 (1994)
18. Fisher, R.A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7(2), 179–188 (1936)
19. Long, B., Yu, P.S., Zhang, Z.M.: A general model for multiple view unsupervised learning. In: *Proceedings of the SIAM International Conference on Data Mining*, pp. 822–833. SIAM, Atlanta (2008)
20. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.T.: Nus-wide: A real-world web image database from national university of singapore. In: *Proc. of ACM Conf. on Image and Video Retrieval (CIVR 2009)*, Santorini, Greece, July 8–10 (2009)

Learning Parametric Dynamic Movement Primitives from Multiple Demonstrations

Takamitsu Matsubara^{1,2}, Sang-Ho Hyon^{3,2}, and Jun Morimoto²

¹ Nara Institute of Science and Technology, Japan

² Department of Brain Robot Interface, ATR-CNS, Japan

³ Ritsumeikan University, Japan

Abstract. This paper proposes a novel approach to learn highly scalable Control Policies (CPs) of basis movement skills from *multiple* demonstrations. In contrast to conventional studies with a *single* demonstration, i.e., Dynamic Movement Primitives (DMPs) [1], our approach efficiently encodes multiple demonstrations by shaping a parametric-attractor landscape in a set of differential equations. This approach allows the learned CPs to synthesize novel movements with novel motion styles by specifying the linear coefficients of the bases as parameter vectors without losing useful properties of DMPs, such as stability and robustness against perturbations. For both discrete and rhythmic movement skills, we present a unified learning procedure for learning a parametric-attractor landscape from multiple demonstrations. The feasibility and highly extended scalability of DMPs are demonstrated on an actual dual-arm robot.

Keywords: Motor Learning, Movement Primitives, Motion Styles.

1 Introduction

Learning from demonstration has shown to be a suitable approach for learning Control Policies (CPs) for humanoid robots [2]. As in [2][3], using a dynamical system to learn the demonstrated movement is becoming popular. One reason is that a trajectory generated by a CP can be easily modulated by changing the parameters of the dynamical system. However, previous studies have focused on learning only from a *single* demonstration. Therefore, the scalability and generalization performance of CPs have been limited. For example, a human demonstration can largely vary depending on the environmental conditions for a particular task. Therefore, some demonstrations on specific situations are not useful to solve the task in other situations.

An illustrative example is a Point To Point (PTP) reaching task with an obstacle (Fig. 1(a)). While this task requires a discrete movement skill (moving from the start to the goal position), humans demonstrate largely different joint trajectories based on the obstacle's height. For this case, simple scaling of the nominal trajectory (no-obstacle's case in Fig. 1(a)), i.e., modification of the goal or spatial and temporal scaling is obviously insufficient to reach the goal position with obstacle avoidance. This issue also arises if we attempt to encode a

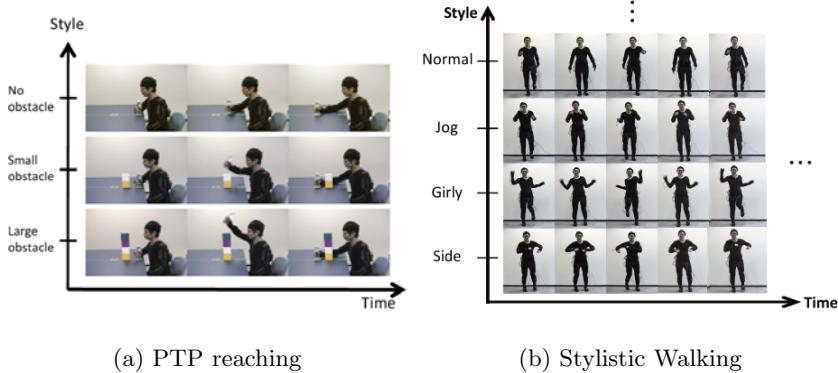


Fig. 1. Sequential snapshots of multiple human demonstrations. (a)The subject performs the same motor skill (point-to-point reaching) in many trials with different obstacles. (b)The subject performs the same content of motion (walking behavior) with different styles.

number of demonstrations with a diversity of motion styles as walking behavior in Fig. 1(b).

These problems motivate us to explore a novel approach to learn the CPs of basis movement skills from *multiple* demonstrations. Computer graphics and animation studies of human motion have proposed several techniques for the synthesis of human-like motions [4][5][6][7]. We adopt such style variable modeling approaches to extract canonical attractor dynamics in a framework of Dynamic Movement Primitives (DMPs) [1][8] by shaping parametric-attractor landscapes. That is, we find the styles in the shapes of the desired attractor landscapes in a set of differential equations unlike previously proposed style variable models that directly model joint trajectories, as in [4][6][7]. This study is based on our preliminary work that focused only on generating discrete movements without thorough evaluations [9]. In this paper, we present a unified learning procedure for both discrete and rhythmic movement skills. The feasibility and highly extended scalability of DMPs are thoroughly demonstrated on an actual dual-arm robot.

2 Learning CPs from “Single” Demonstration

2.1 Dynamic Movement Primitives for Discrete Movement

We first explain the definition of DMPs on discrete movement along with [1][8]. Assume that we have a point attractive system as a basic CP of one-DoF motor system as

$$\tau \dot{z} = \alpha_z (\beta_z (g - y) - z), \quad \tau \dot{y} = z + xf \quad (1)$$

where g denotes goal position, α_z , β_z are time constants, τ is a time scaling factor. y and \dot{y} correspond to the desired position and velocity generated by the

control policy. For appropriate settings of parameters α_z, β_z with the constraint $f = 0$, these equations have a global stability with a unique point attractor g , i.e., y converges to g after a transient from any initial conditions.

In the DMPs, the above dynamics is applied for a learning from the demonstration scenario by introducing an additional dynamical system of state x as

$$\tau \dot{x} = -\alpha_x x \quad (2)$$

and the setting of nonlinear function f with state x and parameter \mathbf{w} as $f(x; \mathbf{w})$, where α_x is a time constant. We call the system in Eq.(2) a *canonical system* as the most basic dynamic system available to create a point attractor. Corresponding to this, the nonlinear dynamical system in Eq.(1) is called *output system*, and the system including both is called the Dynamic Movement Primitives (DMPs). If the initial condition of x is 1 and α_x is properly set for the system to be stable, $x(t) \in [0, 1]$ is considered as a phase variable for $f(x)$ because $f(x)$ is a function of the space of phase variable x , and x also acts as a gating term for $f(x)$. With the above all settings and assumption of the boundness of $f(x; \mathbf{w})$ for all x , y asymptotically converges to the unique point g because the nonlinear function term f vanishes with the convergence of phase x to 0 through time evolution of the canonical system.

Learning parameter \mathbf{w} for shaping the attractor landscape suitable for imitation of a given trajectory $\{y_{\text{demo}}(t_c)\}$, $t_c = c\Delta t$, $c = 1, \dots, C$ with its duration $T = C\Delta t$ can be accomplished by a supervised learning algorithm. The target trajectory is given as $f_{\text{target}}(t_c) = \tau \dot{y}_{\text{demo}}(t_c) - z_{\text{demo}}(t_c)$ in Eq.(1), where $z_{\text{demo}}(t_c)$ is obtained by integrating Eq.(1) with $y_{\text{demo}}(t_c)$ instead of $y(t_c)$. Its input is corresponding phase value $x(t_c)$. We use a vector representation of each trajectory as $\mathbf{y}_{\text{demo}} = [y_{\text{demo}}(t_1), \dots, y_{\text{demo}}(t_C)]^T$ for short. It is also applied for $\mathbf{f}_{\text{target}} = [f_{\text{target}}(t_1), \dots, f_{\text{target}}(t_C)]^T$ and $\mathbf{x} = [x(t_1), \dots, x(t_C)]^T$.

The learned DMP has an attractor landscape to generate similar trajectories to the demonstration by time evolutions. Temporal scaling and modification of goal position can be easily accomplished by manipulating g and τ . Note that for a multi-DoFs motor system, the output system Eq.(1) must be set for every DoF independently. The canonical system can be shared across all DoFs if they are coordinated.

2.2 Dynamic Movement Primitives for Rhythmic Movement

For rhythmic movements, the limit cycle dynamics is modeled by replacing the canonical system of x in Eq.(2) with the following system which has a stable limit cycle in polar coordinates (ϕ, r) :

$$\tau \dot{\phi} = 1, \quad \tau \dot{r} = \mu(r - r_0). \quad (3)$$

Similar to the discrete system, the output system of y and \dot{y} is defined with the phase variable $\tilde{\phi} = \text{mod}(\phi, 2\pi)$ and amplitude signal r as follows:

$$\tau \dot{z} = \alpha_z (\beta_z (y_m - y) - z), \quad \tau \dot{y} = z + rf(x; \mathbf{w}) \quad (4)$$

where $x = \tilde{\phi}$, y_m denotes an anchor point for the oscillatory trajectory, and r_0 is the target amplitude. α_z , β_z are time constants, and τ is a temporal scaling factor. For appropriate settings of parameters α_z, β_z , it converges to a limit cycle attractor. Like the discrete system, supervised learning can be applied for shaping attractor landscape from a demonstrated trajectory by adjusting \mathbf{w} .

3 Learning CPs from “Multiple” Demonstrations

3.1 Parametric Dynamic Movement Primitives (PDMPs)

To encode multiple demonstrations efficiently with preserving useful properties of the DMPs, we propose to model the nonlinear function $f(x; \mathbf{w})$ in Eqs.(1) and (4) by a multi-factor function. We introduce an additional control variable s as $\tilde{f}(x, s; \mathbf{w})$ so that \tilde{f} with specific value of s represents the attractor landscape encoding specific demonstration as:

$$\tilde{f}(x, \mathbf{s}; \mathbf{W}) = \sum_{j=1}^J s_j b_j(x; \mathbf{w}_j) = \mathbf{s}^T \mathbf{b}(x; \mathbf{W}) \quad (5)$$

where $\mathbf{s} = [s_1, \dots, s_J]^T$ is *style parameter*. $\mathbf{b}(x; \mathbf{W}) = [b_1(x; \mathbf{w}_1), \dots, b_J(x; \mathbf{w}_J)]^T$ can be considered as basis functions (typically modeled by Normalized Gaussian Network (NGnet)) which span a subspace including a variety of attractor landscapes and $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_J]^T$ is parameter matrix. With this multi-factor modeling, we can efficiently encode a number of demonstrations with a relatively low dimensional subspace. We call this proposed framework as Parametric Dynamic Movement Primitives (PDMPs).

3.2 Algorithm for Learning Parametric Attractor Landscapes

The learning algorithm of PDMPs from multiple demonstrations has the following four steps:

1. Alignment of demonstrations as a prepossessing for subsequent steps. Assume that we have M sets of trajectories as multiple demonstrations $\{y_{\text{demo}}^m(t_c^m)\}$, $m = 1, \dots, M$, $t_c^m = c\Delta t$, $c = 1, \dots, C^m$, where the duration of each demonstration is given as $T^m = C^m \Delta t$. After selecting a nominal trajectory indexed by $n \in \{1, \dots, M\}$, other trajectories are time-scaled by the ratio $\frac{T^n}{T^m}$ so that all trajectories are represented as the same size of vectors as $\mathbf{y}_{\text{demo}}^m \in \mathbb{R}^{C^n \times 1}$ for all m , that is, $\mathbf{y}_{\text{demo}}^{1:M} = \{\mathbf{y}_{\text{demo}}^1, \dots, \mathbf{y}_{\text{demo}}^M\}$.
2. Calculation of target $\mathbf{f}_{\text{target}}^m$ from demonstration $\mathbf{y}_{\text{demo}}^m$ through numerical integrations with attractor dynamics separately for all m along with the same process of DMPs. By applying this process for all demonstrations, we obtain $\mathbf{f}_{\text{target}}^{1:M} = \{\mathbf{f}_{\text{target}}^1, \dots, \mathbf{f}_{\text{target}}^M\}$. Note that for discrete movements, all demonstration $\mathbf{y}_{\text{demo}}^m$ are translated to start at $y_{\text{demo}}^m(t_c^m) = 0$, while y_m are set by the average of $\mathbf{y}_{\text{demo}}^m$ for all m for rhythmic movements.

3. Extraction of *basis targets* $\mathbf{f}_{\text{basis}}^{1:J} = \{\mathbf{f}_{\text{basis}}^1, \dots, \mathbf{f}_{\text{basis}}^J\}$ from all targets $\mathbf{f}_{\text{target}}^{1:M}$ so that target $\mathbf{f}_{\text{target}}^m$ is approximately represented as $\mathbf{f}_{\text{target}}^m \approx \sum_{j=1}^J s^j \mathbf{f}_{\text{basis}}^j$, where typically $J \ll M$. The basis targets and corresponding style parameters can be extracted by a matrix factorization with Singular Value Decomposition (SVD). Let $\mathbf{F}_{\text{target}}^{\text{all}} = [\mathbf{f}_{\text{target}}^1 \cdots \mathbf{f}_{\text{target}}^M]^T$ be an $M \times C$ matrix. Then, SVD for this matrix leads to the following factorial representation as

$$\mathbf{F}_{\text{target}}^{\text{all}} = \mathbf{U} \Sigma \mathbf{V}^T \approx \mathbf{S} \mathbf{F}_{\text{basis}}. \quad (6)$$

- We define style parameter matrix $\mathbf{S} = [\mathbf{s}^1 \cdots \mathbf{s}^M]^T \in \mathbb{R}^{M \times J}$ to be the first $J (\leq M)$ rows of \mathbf{U} , and basis target matrix $\mathbf{F}_{\text{basis}} = [\mathbf{f}_{\text{basis}}^1 \cdots \mathbf{f}_{\text{basis}}^J]^T \in \mathbb{R}^{J \times C}$ to be the first J columns of $\Sigma \mathbf{V}^T$. The dimension J can be determined with the singular value spectrum (e.g., select J so that $\sum_{j=1}^J \sigma_j / \sum_{m=1}^M \sigma_m > 0.9$).
4. Learning \mathbf{W} (or basis functions $\mathbf{b}(x; \mathbf{W})$) is achieved by supervised learning with $\mathbf{f}_{\text{basis}}^j$ and corresponding phase vector \mathbf{x} separately for each $j \in \{1, \dots, J\}$ as

$$\mathbf{w}_j^* \leftarrow \arg \min_{\mathbf{w}_j} \sum_{c=1}^{C^n} \|f_{\text{basis}}^j(t_c) - b_j(x_c; \mathbf{w}_j)\|. \quad (7)$$

The optimal parameter \mathbf{w}_j^* with a proper supervised learning algorithm leads to a parametric-attractor landscape $\tilde{f}(x, \mathbf{s}; \mathbf{W}^*)$ parametrized by style parameter \mathbf{s} .

The learning $\tilde{f}(x, \mathbf{s}; \mathbf{W})$ is based on the extraction of bases $\mathbf{f}_{\text{basis}}^{1:J}$ which span J dimensional subspace encoding all attractor landscapes for all demonstrations. Typically $J \ll M$ if target trajectories are correlated, i.e., even with the increase of the number of demonstrations M , the dimension of style parameter J could be relatively small. This property leads to a compact representation of the PDMPs from multiple demonstrations. While the supervised learning problem in step 4. is often solved by standard least-square techniques such as LWL, in this paper, we utilize the Gaussian process regression [10] for the algorithmic simplicity and generalization performance.

3.3 Properties of CPs Based on PDMPs

Since the PDMPs are based on the same attractor dynamics to DMPs except attractor landscapes \tilde{f} , it preserves several properties. Thus, scaling of the goal g for the discrete CP and of the amplitude r_0 for the rhythmic CP does not affect the shape of the attractor landscape. The period (duration) is also manipulated by τ . This property is compatible with the linear time-scaling procedure as a pre-processing of basis extraction in the learning algorithm, i.e., original trajectories of demonstrations may be reconstructed with proper setting of τ . Even if these parameters are modified in on-line (during movement), CPs generate a smooth trajectory converging to attractor with a transient. This property is significant for on-line control of robots as explored in [11, 12]. The PDMPs additionally

supply highly extended scalability resulted from the multiple demonstrations, such as with respect to motion styles and variations. These factors can also be modified in on-line independent with other factors. Some of usefulness will be illustrated through our experimental results in Section 4.

4 Experimental Evaluations

4.1 PTP Reaching Movement with Obstacle Avoidance

In the first experiment we applied our technique to learn a motor skill of PTP reaching movements with a human-like obstacle avoidance strategy as a discrete PDMP from multiple demonstrations, and it was implemented on and demonstrated by our dual-arm robot as shown in Fig. 2. As the multiple demonstrations for the task, the subject performed the PTP reaching movement on the table with different heights of the obstacles with avoiding them from over as shown in Fig. II(a). In the all cases, the start and goal positions were fixed. The differences in height of the obstacle largely affected in resulted joint trajectories while it was preserving the content as the PTP reaching skill. Fifteen motion sequences were observed by a gyroscopic motion capture system. The joint trajectories of human's right arm were translated in the robot's joint space (four DoFs without wrist) by an inverse kinematics method and they were used as demonstrations.

Since the variation of motion style of the multiple demonstrations were corresponding to the variation of height of obstacles in this case, the learned PDMP obtained an additional scalability, i.e., an human-like obstacle avoidance strategy. By manipulating style parameter $s \in \mathbb{R}^3$, the PDMP can represent a variety of CPs suitable for avoiding several heights of obstacles. Of course, other scalability such as goal position g and duration τ can be also controlled by each parameters, respectively.

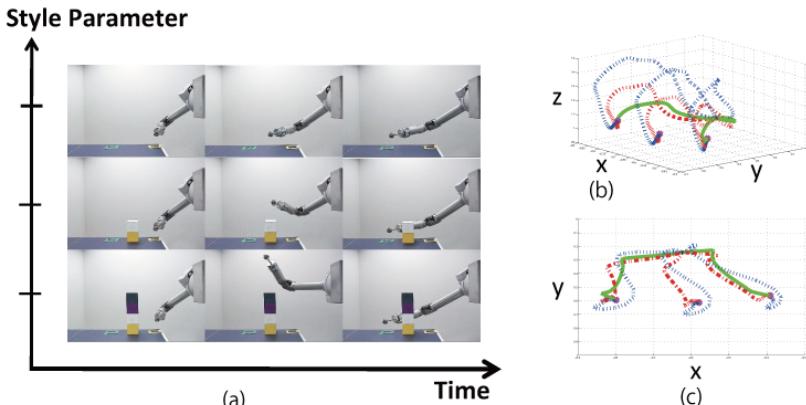


Fig. 2. PTP reaching with obstacle avoidance strategy on the dual-arm robot (using right arm). (a) shows sequential snapshots of the behavior. (b) and (c) plots hand trajectories with several styles and goals.

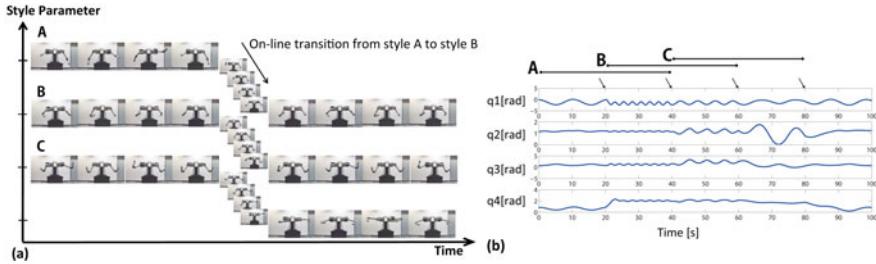


Fig. 3. Stylistic arm swing movements as a part of walking behaviors by the dual-arm robot. Several scaling parameters including style parameter were changed during movement in on-line manner at each 20s interval. (a) Sequential snapshots of the behavior. (b) Time series of all joint trajectories. $q_1 \sim q_4$ indicate 1st ~ 4th joint angles of the right arm, respectively.

The learned PDMP was successfully implemented on the robot. The demonstrations of the reaching with several obstacles were presented in Fig. 2. The style parameter was manually selected from \mathbf{S} by choosing one column for each, i.e., $\mathbf{s} = [-0.06, -0.12, -0.38]^T$ (top), $[0.12, -0.28, 0.00]^T$ (middle) and $[0.27, -0.23, 0.50]^T$ (bottom). In the experiment, several obstacles were successfully avoided as presented in Fig. 2(a). Figure 2(b) and (c) showed the generated trajectories with several styles and goals. It is important to note that all trajectories are properly converging to the goal state regardless of style parameters.

4.2 Stylistic Arm Movements of Walking Behavior

In the second experiment we applied our technique to learn a motor skill of walking behavior with several motion styles as a rhythmic PDMP from multiple demonstrations. As shown in Fig. 1(b), the subject performed walking behavior with different motion styles. Twelve motion sequences including a diversity of walking styles (such as normal walking, jogging, girly style, striding and so on) were observed by the motion capture system.

Since the variation of motion style of the multiple demonstrations were corresponding to the variation of walking styles in this case, the learned PDMP had an additional scalability, i.e., the style synthesis of walking motion. By manipulating style parameter $\mathbf{s} \in \mathbb{R}^4$, the PDMP can represent CPs of walking behavior with a variety of motion styles.

The learned PDMP was successfully implemented on the robot. The demonstrations of the walking behavior were presented in Fig. 3. The style parameter \mathbf{s} , frequency τ and amplitude r_0 were changed during movement in on-line manner at each 20s interval, where the style parameters were selected from \mathbf{S} as the discrete movement case. y_m was also properly set with \mathbf{s} . In the experiment, walking styles, frequency and amplitude were smoothly converged to a specific motion style with a transient as in Fig. 3. The generated walking represented the

motion styles of human's demonstration very well. Note again that with DMP learned from normal walking demonstration, even if we modify its frequency and amplitude parameters consistently, it is still impossible to generate jogging-style walking behavior. On the other hand, our technique allows such style synthesis by manipulating small number of parameters.

5 Conclusions

This paper presents a novel approach to learn highly scalable CPs of basis movement skills from multiple demonstrations. The feasibility and highly extended scalability of the learned CPs were demonstrated on an actual dual-arm robot for PTP reaching with obstacle avoidance and stylistic walking skills. Future work includes automatic adjustment of style parameters for environmental conditions.

References

- Ijspeert, A.J., Nakanishi, J., Schaal, S.: Learning attractor landscapes for learning motor primitives. *Advances in Neural Information Processing Systems* 15, 1523–1530 (2002)
- Schaal, S.: Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences* 3(6), 233–242 (1999)
- Inamura, T., Toshima, I., Nakamura, Y.: Acquisition and embodiment of motion elements in closed mimesis loop. In: *IEEE International Conference on Robotics and Automation*, pp. 1539–1544 (2002)
- Brand, M., Hertzmann, A.: Style machines. In: *Proceedings of the 2000 SIGGRAPH*, pp. 183–192 (2000)
- Hsu, E., Pulli, K., Popovic, J.: Style translation for human motion. *ACM Transactions on Graphics* 24(3), 1082–1089 (2005)
- Wang, J.M., Fleet, D.J., Hertzmann, A.: Multifactor gaussian process models for style-content separation. In: *Proceedings of the 24th Annual International Conference on Machine Learning*, pp. 975–982 (2007)
- Taylor, G.W., Hinton, G.E.: Factored conditional restricted boltzmann machines for modeling motion style. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1025–1032 (2009)
- Schaal, S., Mohajerian, P., Ijspeert, A.: Dynamics systems vs. optimal control -a unifying view. *Progress in Brain Research* 165, 425–445 (2007)
- Matsubara, T., Hyon, S.-H., Morimoto, J.: Learning stylistic dynamic movement primitives from multiple demonstrations. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* (accepted 2010)
- Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. MIT Press, Cambridge (2006)
- Park, D.-H., Hoffmann, H., Pastor, P., Schaal, S.: Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In: *IEEE-RAS International Conference on Humanoid Robotics* (2008)
- Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: *IEEE International Conference on Robotics and Automation*, pp. 763–768 (2009)

An Algorithm on Multi-View Adaboost

Zhijie Xu and Shiliang Sun

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, P.R. China
zjxu09@gmail.com, slsun@cs.ecnu.edu.cn

Abstract. Adaboost, one of the most famous boosting algorithms, has been used in various fields of machine learning. With its success, many people focus on the improvement of this algorithm in different ways. In this paper, we propose a new algorithm to improve the performance of adaboost by the theory of multi-view learning, which is called Embedded Multi-view Adaboost (EMV-Adaboost). Different from some approaches used by other researchers, we not only blend multi-view learning into adaboost thoroughly, but also output the final hypothesis in a new form of the combination of multi-learners. These theories are combined into a whole in this paper. Furthermore, we analyze the effectiveness and feasibility of EMV-Adaboost. Experimental results with our algorithm validate its effectiveness.

Keywords: Adaboost, Multi-view learning, Multi-learner learning, Machine learning, Supervised learning.

1 Introduction

In the research of machine learning, it is hard to construct a very good classifier using limited number of samples in most of the time. However, the combination of a number of weaker learners can change the situation and promote the performance of classification a lot. Recently, there has been great interest in ensemble methods for learning classifiers, in particular about boosting algorithms [12]. This kind of method for classification tasks constructs a set of base classifiers from the training data and performs classification by taking votes on the prediction of each base classifier [3]. In all of the boosting algorithms, Adaboost [4] is the most famous one. It has caught the eyes of a number of researchers and been used to deal with different kinds of problems. Moreover, due to the need of reality, the improvement of this algorithm still continues. Obviously, with the development of machine learning, the study of adaboost can become more diverse.

In real-world of machine learning, boosting has been widely applied in various areas such as text filtering, ranking, learning in natural language processing, image retrieval, medical diagnosis, customer monitoring and segmentation [5]. But sometimes, they can not reach the desired outcome. As a matter of course, some new approaches such as Real Adaboost [6] and σ Boost [7] appear. In all of

them, the combination of adaboost and multi-view is one typical approach. At the present, this approach has alerted its usage in some fields, especially in the problem of image processing. Nevertheless, some defects about them can not be neglected. Deng Peng [8], who combined adaboost and multi-view to improves the performance of face recognition, missed potential relationship between different views. Bo Wu [9], just proposed a linear combination of real adaboost and the technique of multi-view face detection. In most of the papers, they have all used these two approaches independently, and not focused on the mergence of them deeply.

In this paper, our key idea is to design an innovative algorithm, Embedded Multi-view Adaboost (EMV-Adaboost) based on the methods presented in [1]. Naturally, we embed multi-view learning [10] into adaboost algorithm and adapt the combination of multi-learner to output the hypothesis. In addition, in order to balance the relationship between different views and illustrate their contribution to the algorithm, the definitions of some parameters are refreshed with the appearance of the new one. Following this, the direction and analysis of our proposed algorithm can prove the feasibility and efficiency of EMV-Adaboost compared to traditional adaboost with the help of Probabilistically Approximately Correct (PAC) theory. At the same time, the experimental results support our theory as well and indicate that this algorithm is worthwhile to research much more continuously.

The remainder of this paper is organized as follows. In Section 2, we introduce the adaboost algorithm. Following this, in Section 3, we describe our EMV-Adaboost algorithm clearly and deeply. Then, details about the experiments are provided in Section 4. The next section shows the analysis about our algorithm, EMV-Adaboost compared to basis ababoost followed by the conclusion in Section 6.

2 Adaboost

Adaboost is a general supervised learning technique for incrementally building linear combinations of weak models to generate a strong predicative model [11]. In this algorithm, we get the input data $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ where x_i belongs to a domain X and y_i belongs to the class label set $Y = \{0, 1\}$. Then, we supply a weight set $W = \{w_1, w_2, \dots, w_n\}$ to every sample, and initialize them to $1/N$ for every element at first. Next, we start the iteration for T times with the distribution P of samples given in the help of W . In every iteration, the algorithm come to one weaker classifier $C(i)$. Moreover, after that, W and P are updated connecting to the error of the weaker learner. Finally, the set of weaker classifiers $C = \{C(1), C(2), \dots, C(n)\}$ are combined by weighted majority vote into the final classifier, which performs better than any of the weaker classifiers. (In our study, $T = 30$.)

Detailed algorithm can be seen below :

Algorithm Adaboost

Input: sequence of n labeled examples $X = (x_1, y_1), \dots, (x_n, y_n)$
distribution D over the n examples
integer T specifying number of iterations

Initialize the weight vector: $w_1^i = D(i)$ for $i = 1, \dots, n$.

For $t = 1, 2, \dots, T$

1. Set $P^t = \frac{w^t}{\sum_{i=1}^n w_i^n}$
2. Call WeakLearn with distribution P^t , get back a hypothesis $h_t : X \rightarrow \{0, 1\}$
3. Calculate the error of $h_t : \varepsilon_t = \sum_{i=1}^n p_i^t |h_t(x_i) - y_i|$
4. Set $\beta_t = \frac{\varepsilon_t}{(1-\varepsilon_t)}$
5. Set the new weights vector to be: $w_i^{t+1} = w_i^t \beta_t^{1-|h_t(x_i)-y_i|}$

End of For

Output the hypothesis:

$$h_f(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T (\log 1/\beta_t) h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \log 1/\beta_t \\ 0 & \text{otherwise} \end{cases}$$

3 Our Proposed Method

On the basis of the algorithms mentioned before, in this section, we develop our algorithm by embedding multi-view into adaboost algorithm.

3.1 Embedded Multi-View Adaboost(EMV-Adaboost)

For the purpose of using the benefits of adaboost and multi-view learning exhaustively, we design a new algorithm combining them together. Easy to see that, previous works always neglect the relationship between them and only use them as certain tools. As a result, now we change adaboost algorithm a lot and learn it from two different views with the help of the parameters defined by us. At the final step of the algorithm, we construct a new way to combine multi-learners to come to the final classifier. The experiments in the following part show that this algorithm is feasible and effective. Detailed algorithm is given below:

3.2 Explanation about EMV-Adaboost

According to the algorithms, it needs to pay attention to the following points. Firstly, we come to two weaker learners every iteration by two views: $V1$ and $V2$, which are formed at the beginning. In addition, in the step 3 of algorithm EMV-Adaboost, we suppose that one sample will contribute to the error rate

Algorithm EMV-Adaboost

Input: sequence of n labeled examples $X = (x_1, y_1), \dots, (x_n, y_n)$
 distribution D over the n examples
 integer T specifying number of iterations

Divide all of the features into two views: $V1$ and $V2$

Initialize the weight vector: $w_1^i = D(i)$ for $i = 1, \dots, n$.

For $t = 1, 2, \dots, T$

1. Set $P^t = \frac{W^t}{\sum_{i=1}^n w_i^n}$
2. Call WeakLearn with distribution P^t , get back two weaker learners from two views:
 $h_t^{V1}, h_t^{V2} : X \rightarrow \{0, 1\}$
3. Calculate the error of h_t : $\varepsilon_t = \sum_{i=1}^n p_i^t \{\text{Max}\{|h_t^{V1}(x_i) - y_i|, |h_t^{V2}(x_i) - y_i|\}\}$
4. Calculate the percentage of the samples predicted same by two weaker learners :
 $agree_t = \frac{\sum_{i=1}^n |h_t^{V1}(x_i) - h_t^{V2}(x_i)|}{n}$
5. Set $\epsilon_t = \varepsilon_t agree_t$
6. Set $\beta_t = \frac{\epsilon_t}{(1-\epsilon_t)}$
7. Set the new weights vector to be: $w_i^{t+1} = w_i^t \beta_t^{1-\text{Max}\{|h_t^{V1}(x_i) - y_i|, |h_t^{V2}(x_i) - y_i|\}}$

End of For

Output the hypothesis:

$$h_f(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T \sum_{i=1}^2 h_t^{V^i}(x) \geq T \\ 0 & \text{otherwise} \end{cases}$$

as long as it is predicted incorrectly in either of the weaker learners in every iteration. Detailed formula has been written as:

$$\varepsilon_t = \sum_{i=1}^n p_i^t \{\text{Max}\{|h_t^{V1}(x_i) - y_i|, |h_t^{V2}(x_i) - y_i|\}\}. \quad (1)$$

What is more, we define a new parameter $agree_t = \frac{\sum_{i=1}^n |h_t^{V1}(x_i) - h_t^{V2}(x_i)|}{n}$ to express the percentage of the samples predicted same by them. At the same time, we define another new parameter to combine the ε_t and $agree_t$ together so as to reflect the value about each view and adjust the parameter β_t . Detailed formulas have been written as:

$$\epsilon_t = \varepsilon_t agree_t. \quad (2)$$

$$\beta_t = \frac{\epsilon_t}{(1 - \epsilon_t)}. \quad (3)$$

Finally, in the hypothesis of the Output step, we use the approach of majority vote by all of weaker learners together to construct final classifier (In our study, the number of weaker learners is $2 \times T = 60$). It can be seen clearly that if at least half of weaker learners predict the sample as label 1, the sum of the outcome made by weaker learners will be larger than T . Detailed formula has been written as:

$$h_f(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T \sum_{i=1}^2 h_t^{Vi}(x) \geq T \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

In our paper, we make use of embedded multi-view learning to improve the adaboost algorithm. Due to this factor, on one hand, how to embody the otherness and diversification between two views is significant. On the other hand, it is necessary as well for us to indicate and enlarge the contribution of every weaker learner.

According to these two reasons above, we define one important parameter $agree_t$ and redefine some other parameters. Because of the attendance of $agree_t$, formulas (2) and (3), the difference between two views can be indicated more effectively, which can help to improve their contribution to the algorithm. Furthermore, these changes promote the diversity of different weaker learners and enlarge their own characteristics. This action can make a great impact on the final hypothesis because every weaker learner now can alert their characteristic to the output more sufficiently. With the help of multi-view learning, our algorithm can reach a good performance.

4 Experiments and Results

In this section, we implement the experiments of two-classes classification problem focusing on several *two-classes* data sets from UCI repository by basic threshold classifier. Table 1 demonstrates the summary of data sets.

Table 1. Summary of data sets

Dataset	StdText	Wpbc	Sonar
Number of Sample	1051	569	208
Number of features	244	30	60
Ratios of train set to test set	3:1	3:1	5:1

Because there exists many different ways to distribute two views, we run the experiments on every data set for ten times. As a result, we regard the mean of them as the final scores. In order to get more accurate outcome, we calculate the standard deviation(Std) simultaneously. Table 2 gives the classification results.

Table 2. Experimental results

Error rate	StdText	Wpbc	Sonar
Adaboost	0.038	0.0352	0.2857
Mean±Std of EMV-Adaboost	0.0285±0.0048	0.02676±0.0045	0.2486±0.0271

Table 2 indicates clearly that our proposed algorithm, Embedded Multi-View Adaboost (EMV-Adaboost), outperforms the standard Adaboost algorithm in every data set.

5 Analysis of EMV-Adaboost

Now, we analyze the bound of error $\varepsilon = \Pr_{i \sim D}[h_f(x_i) \neq y_i]$ of the final hypothesis h_f output by EMV-Adaboost.

Theorem 1. Suppose the weak learning algorithm WeakLearn, when called by EMV-Adaboost, generates hypotheses with error $\varepsilon_1, \dots, \varepsilon_T$ (as defined in Step 3.) Then, under the hypothesis that there at least one weaker learner predicts incorrectly in every iteration for the sample predicted incorrectly, the error $\varepsilon = \Pr_{i \sim D}[h_f(x_i) \neq y_i]$ of the final hypothesis h_f output by EMV-Adaboost is bounded above by

$$\varepsilon \leq \prod_{t=1}^T (1 - (1 - \varepsilon_t)(1 - \beta_t)). \quad (5)$$

Proof. According to the Theorem given in [112], we can come to the Eq.(7) through the evaluation of Eq.(6) in step 7 in EMV-Adaboost algorithm

$$w_i^{t+1} = w_i^t \beta_t^{1 - \text{Max}\{|h_t^{V1}(x_i) - y_i|, |h_t^{V2}(x_i) - y_i|\}}. \quad (6)$$

$$w_i^{t+1} \leq \prod_{t=1}^T (1 - (1 - \varepsilon_t)(1 - \beta_t)). \quad (7)$$

The final weight of any instance i is

$$w_i^{t+1} = D(i) \prod_{t=1}^T \beta_t^{1 - \text{Max}\{|h_t^{V1}(x_i) - y_i|, |h_t^{V2}(x_i) - y_i|\}}. \quad (8)$$

Then, combining Eqs.(8) and (9), we can lower bound the sum of the final weights by the sum of the final weights of the examples on which h_f is incorrect, which can be seen in Eq.(10)

$$\sum_{i=1}^N w_i^{t+1} \geq \sum_{i: h_f(x_i) \neq y_i} w_i^{t+1}. \quad (9)$$

$$\sum_{i=1}^N w_i^{t+1} \geq \sum_{i: h_f(x_i) \neq y_i} [D(i) \prod_{t=1}^T \beta_t^{1 - \text{Max}\{|h_t^{V1}(x_i) - y_i|, |h_t^{V2}(x_i) - y_i|\}}]. \quad (10)$$

Now, in order to make the Eq.(10) always be true, we want to get the maximum value of $\sum_{i:h_f(x_i) \neq y_i} [D(i) \prod_{t=1}^T \beta_t^{1-\text{Max}\{|h_t^{V1}(x_i)-y_i|, |h_t^{V2}(x_i)-y_i|\}}]$. This means that it is better to let us suppose $\beta_t^{1-\text{Max}\{|h_t^{V1}(x_i)-y_i|, |h_t^{V2}(x_i)-y_i|\}} (\beta_t \leq 1)$ come to its maximum, 1, in every iteration. Consequently, for all the sample predicted incorrectly, we hope that there at least one weaker learner comes to the wrong label in every iteration, because we assume that one sample will contribute to the error rate as long as it is predicted incorrectly in either of the weaker learners (Eq.1).

What is more, it needs to notice that the total number of weaker learners is $2 \times T$. With the h_f supplied by us, there are at least T weaker learners get the wrong hypothesis about the samples predicted incorrectly. Now, we can suppose that in every iteration, at least one weaker learner predicts incorrectly. As a result, we can get the formula $1 - \text{Max}\{|h_t^{V1}(x_i)-y_i|, |h_t^{V2}(x_i)-y_i|\} = 0$, which can deduce to the formula $\beta_t^{1-\text{Max}\{|h_t^{V1}(x_i)-y_i|, |h_t^{V2}(x_i)-y_i|\}} = 1$.

On the basis of the analysis above, the evolution of Eq.(10) can be seen clearly in Eq.(11).

$$\sum_i^N w_i^{T+1} \geq \sum_{i:h_f(x_i) \neq y_i} D(i) = \varepsilon. \quad (11)$$

Combining Eqs.(7) and (11), we get the theorem about the ε in our algorithm EMV-Adaboost as following:

$$\varepsilon \leq \prod_{t=1}^T (1 - (1 - \varepsilon_t)(1 - \beta_t)). \quad (12)$$

□

Compared to the following bound of ε got in [1] (Eq.(13)), it can be seen clearly that our hypothesis on error ε obviously works much better.

$$\varepsilon \leq \prod_{t=1}^T \frac{(1 - (1 - \varepsilon_t)(1 - \beta_t))}{\sqrt{\beta_t}}. \quad (13)$$

6 Conclusion and Future Work

In this paper, we propose the algorithms, Embedded Multi-View Adaboost. It embeds multi-view learning into adaboost and makes fully use of the contribution of two views through bringing out their own characteristics. In the help of two-class classification problem experiments, this algorithm proves to be quite useful and effective than traditional Adaboost. More importantly, appropriate theory analysis let us believe the reliability and feasibility of it. Overall speaking, compared to our proposed algorithm, it maybe a new challenge to research it from more than two views.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Projects 60703005 and 61075005.

References

1. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Science* 55, 119–139 (1997)
2. Grove, A.J., Schuurmans, D.: Boosting in the limit: Maximizing the margin of learned ensembles. In: *Proceedings of the National Conference on Artificial Intelligence*, pp. 692–699 (1998)
3. Vimal, B.V., Amit, G., Amit, T.: Boost a weak learner to a strong learner using ensemble system approach. In: *IEEE International Advance Computing Conference*, pp. 1432–1436 (2009)
4. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: *13th IEEE International Machine Learning Conference* (1996)
5. Schapire, R.E.: The Boosting Approach to Machine Learning: An Overview. *Lecture Notes in Statistics*, pp. 149–172. Springer, New York (2003)
6. Wang, Z., Fang, C., Ding, X.: Asymmetric Real Adaboost. In: *19th International Conference on Pattern Recognition* (2008)
7. Lima, N.H.C., Neto, A.D.D., Melo, J.D.: Creating an Ensemble of Diverse Support Vector Machines Using Adaboost. In: *Proceedings of The 2009 International Joint Conference on Neural Networks*, pp. 2342–2346 (2009)
8. Deng, P., Pei, M.: Multi-View Face Detection Based on AdaBoost and Skin Color. *Intelligent Networks and Intelligent System*, 457–460 (2008)
9. Wu, B., Ai, H., Huang, C., Lao, S.: Fast rotation invariant multi-view face detection based on real adaboost. In: *6th International Conference on Automatic Face and Gesture Recognition*, pp. 79–84 (2004)
10. Zhang, Q., Sun, S.: Multiple-view multiple-learner active learning. *Pattern Recognition* 43(9), 3113–3119 (2010)
11. Haffari, G., Wang, Y., Wang, S., Mori, G., Jiao, F.: Boosting with incomplete information. In: *25th International Conference on Machine Learning*, pp. 368–375 (2008)
12. Schapire, R.E.: A brief introduction to boosting. In: *International Joint Conference on Artificial Intelligence*, vol. 16, pp. 1401–1406 (1999)

An Analysis of Speaker Recognition Using Bagging CAN2 and Pole Distribution of Speech Signals

Shuichi Kurogi, Shota Mineishi, and Seitaro Sato

Kyusyu Institute of technology, Tobata, Kitakyushu, Fukuoka 804-8550, Japan
kuro@cntl.kyutech.ac.jp, mineishi@kurolab_CNTL.kyutech.ac.jp
http://kurolab_CNTL.kyutech.ac.jp/

Abstract. A method of speaker recognition which uses feature vectors of pole distribution derived from piecewise linear predictive coefficients obtained by bagging CAN2 (competitive associative net 2) is presented and analyzed. The CAN2 is a neural net for learning efficient piecewise linear approximation of nonlinear function, and the bagging CAN2 (bootstrap aggregating version of CAN2) is used to obtain statistically stable multiple linear predictive coefficients. From the coefficients, the present method obtains a number of poles which are supposed to reflect the shape of the speaker's vocal tract. Then, the pole distribution is used as a feature vector for speaker recognition. The effectiveness is analyzed and validated using real speech data.

Keywords: speaker recognition, pole distribution, bagging CAN2.

1 Introduction

An artificial neural net called CAN2 (competitive associative net 2) has been introduced for learning efficient piecewise linear approximation of nonlinear function [1][2][3][4] by means of utilizing competitive and associative schemes [5][6]. The effectiveness has been shown in several applications, and in the application to learning and analyzing speech signals, we show that the speech time-series is reproduced and recognized with high precision by the bagging CAN2, or a bagging (bootstrap aggregating) version of CAN2 [2]. Recently, we show that the poles of piecewise linear predictive coefficients obtained by the bagging CAN2 are effective for speaker recognition [4]. Here, note that among the previous research studies of speaker recognition, 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) [7][8][9][10]. Namely, these methods extract multi-dimensional 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. Thus, a single feature vector of the LPC or the MFCC corresponds to the average of multiple piecewise linear predictive coefficients of the bagging CAN2. Namely, the bagging CAN2 has stored more precise information on the speech signal so that it can reproduce speech signal with high precision.

In this paper, we show a formulation of the method of speaker recognition using bagging CAN2 and pole distribution of speech signals in [2] (see [4] for original and detailed explanation), and then an analysis and the effectiveness with the experimental results using real speech data in [3].

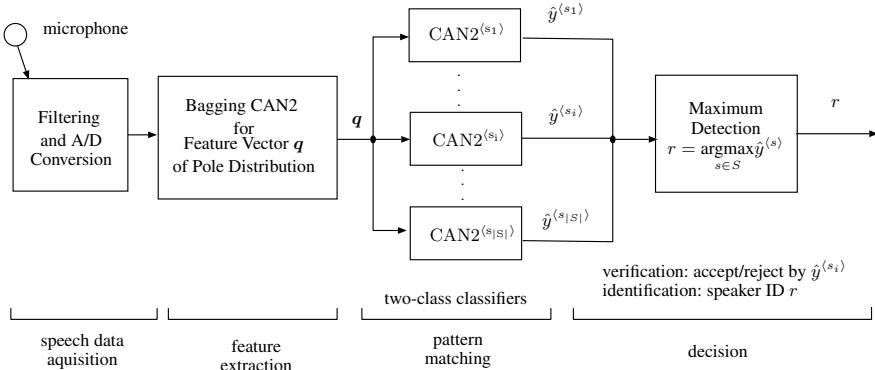


Fig. 1. Speaker recognition system using the CAN2s

2 Speaker Recognition Using CAN2 and Pole Distribution

2.1 Overview of Speaker Recognition

Fig. 1 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 is 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.

2.2 Single and Bagging CAN2

First, we briefly show single and bagging CAN2 (see [3] for details). Suppose a k -dimensional input vector $\mathbf{x}_t \triangleq (x_{t1}, x_{t2}, \dots, x_{tk})^T$ and a scalar output value y_t at a discrete time $t = 1, 2, \dots$ have the relationship given by

$$y_t \triangleq r_t + \epsilon_t = f(\mathbf{x}_t) + \epsilon_t \quad (1)$$

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. 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

$$\hat{y}_t \triangleq \hat{f}(\mathbf{x}_t) \triangleq \tilde{y}_{c_t} \triangleq \mathbf{M}_{c_t} \tilde{\mathbf{x}}_t, \quad (2)$$

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 $c_t \triangleq \operatorname{argmin}_{j \in I^N} \|\mathbf{x}_t - \mathbf{w}_j\|$. The above function approximation partitions the input space $V \in \mathbb{R}^k$ into the Voronoi (or Dirichlet) regions $V_i \triangleq \{\mathbf{x} \mid i = \operatorname{argmin}_{j \in I^N} \|\mathbf{x} - \mathbf{w}_j\|\}$ 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 [1] for details), which we also use in this application.

The bagging (bootstrap aggregation) [11] 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 for generating statistically stable linear coefficients as follows: For a given training dataset $D^n \triangleq \{(\mathbf{x}_t, y_t) \mid t \in I^n = \{1, 2, \dots, n\}\}$, let $D^{n\alpha^\sharp, j}$ be the j th bootstrap sample set (multiset, or bag) involving αn elements, where the elements in $D^{n\alpha^\sharp, j}$ are resampled with replacement from D^n , where $j \in I^b \triangleq \{1, 2, \dots, b\}$ and α is a positive constant. We use $b = 20$ and $\alpha = 0.7$ in the experiments shown below (see [3] for the effect of α). The prediction of the target value $r_t = f(\mathbf{x}_t)$ is done by the mean given by

$$\hat{y}_t^{\text{bg}} \triangleq \frac{1}{b} \sum_{j \in I^b} \hat{y}_t^{[j]} = \frac{1}{b} \sum_{j \in I^b} \mathbf{M}_{c_t^{[j]}} \tilde{\mathbf{x}}_t \quad (3)$$

where $\hat{y}_t^{[j]} = \mathbf{M}_{c_t^{[j]}} \tilde{\mathbf{x}}_t$ is the prediction by the j th CAN2 which has been trained with $D^{n\alpha^\sharp, j}$, and $c_t^{[j]}$ is the index of the selected unit in the j th CAN2.

2.3 Model of Speech Signal Production

The most standard model of the speech production is the all-pole linear prediction model described as follows (see [7]); a speech output signal y_t at a discrete time t is modeled by a linear combination of its past values $\mathbf{x}_t = (y_{t-1}, y_{t-2}, \dots, y_{t-k})$ as $y_t = \mathbf{a} \mathbf{x}_t + u_t$ where $\mathbf{a} = (a_1, a_2, \dots, a_k)$ represents the predictor coefficients, u_t is the input to the vocal system, and k is called prediction order. On the other hand, the speech signal in the present research is modeled by a more general expression as shown in Eq.(1), where we use the same $\mathbf{x}_t = (y_{t-1}, y_{t-2}, \dots, y_{t-k})$ as for the above standard model. Now, let us rewrite the prediction $\hat{y}_t^{[j]} = \mathbf{M}_{c_t^{[j]}} \tilde{\mathbf{x}}_t$ involved in Eq.(3) as

$$y_t = \mathbf{M}_i^{[j]} (1, y_{t-1}, y_{t-2}, \dots, y_{t-k})^T \quad (4)$$

where $\hat{y}_t^{[j]}$ and $\mathbf{M}_{c_t^{[j]}}$ are replaced by y_t and $\mathbf{M}_i^{[j]}$ for simplicity.

2.4 Pole Distribution

In order to obtain stable feature vectors from $\mathbf{M}_i^{[j]}$ in Eq.(4), we apply the z -transform to Eq.(4), then we have

$$Y(z) = \frac{M_{i0}^{[j]}}{1 - \sum_{m=1}^k M_{im}^{[j]} z^{-m}} = \frac{M_{i0}^{[j]}}{\prod_{m=1}^k (1 - p_{im}^{[j]}/z)} \quad (5)$$

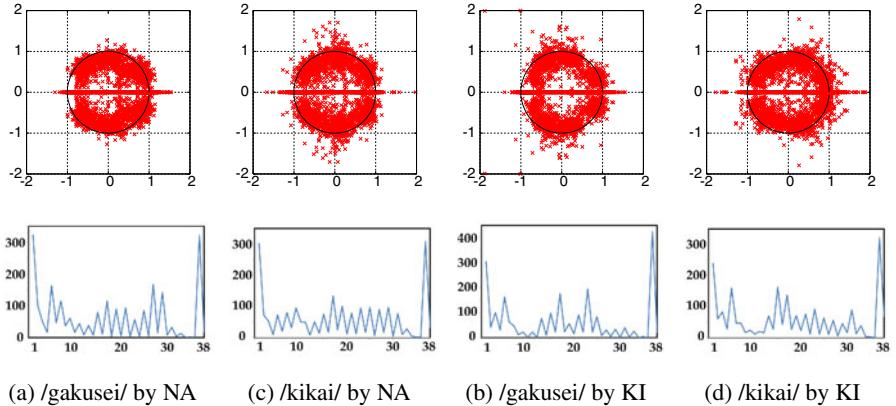


Fig. 2. Examples of pole distributions (upper) and the feature vectors (lower) for the Japanese words /gakusei/ and /kikai/ by speakers NA and KI for $(r_{\max}, n_r, n_\theta) = (2, 2, 18)$

where $p_{im}^{[j]}$ are the poles of $Y(z) = Z(y_t)$, or the z -transform of y_t . Then, a feature vector of pole distribution is obtained as follows; first, let $p_{im}^{[j]} = r_{im}^{[j]} \exp(-j\theta_{im}^{[j]})$ be the polar form of the pole, where $r_{im}^{[j]}$ is the magnitude, $\theta_{im}^{[j]}$ is the argument, and $j^2 = -1$. Let us evenly divide the magnitude range $[0, r_{\max}]$ and the argument range $[0, \pi]$ into n_r and n_θ regions, respectively. Then, by counting the number of poles in each region by raster scan from smaller to larger magnitude and 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 l th region for the argument $\theta_{im}^{[j]}$ as $[(l-1)\pi/n_\theta, l\pi/n_\theta]$ for $l = 1, 2, \dots, n_\theta$, and $[\pi, \pi]$ for $l = n_\theta + 1$. Moreover, we neglect the poles with negative imaginary parts, because the 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. 2.

In this figure, we can see that there are oscillatory poles, or the poles out of the unit circle. Although the vocal tract usually is supposed to be passive, those oscillatory poles are necessary for reproducing the high-quality vowel signal [2].

2.5 Speaker Verification and Identification by Single CAN2s

For learning and matching the feature vectors, we utilize a single CAN2 for each speaker as follows. Let $Q^{(s)}$ be the set of feature vectors \mathbf{q} of the pole distribution obtained from a speech signal of a speaker s , and $\text{CAN2}^{(s)}$ be the CAN2 as a two-class classifier for s . Furthermore, let us modify the relationship shown in Eq.(1) as $y^{(s)} = f^{(s)}(\mathbf{q}) + \epsilon_q$, where ϵ_q indicates the effect of the stochastic variation of \mathbf{q} , and the target function is binarized as

$$f^{(s)}(\mathbf{q}) = \begin{cases} 1, & \text{if } \mathbf{q} \in Q^{(s)}, \\ -1, & \text{otherwise.} \end{cases} \quad (6)$$

For evaluating the error in the learning method [1], we binarize the output of the CAN2 given by Eq.(2) as

$$v^{(s)} = \begin{cases} 1, & \text{if } \hat{y}^{(s)} = \mathbf{M}_{c^{(s)}} \tilde{\mathbf{q}} > 0, \\ -1, & \text{otherwise.} \end{cases} \quad (7)$$

where $\mathbf{M}_{c^{(s)}}$ 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 the input-output pair $(\mathbf{q}, f^{(s)}(\mathbf{q}))$ of the training data for $\mathbf{q} \in Q^{(s)}$, the speaker verification for the speaker s is done by Eq. 7.

The speaker identification number r is obtained by the maximum detection given by $r = \underset{s \in S}{\operatorname{argmax}} \hat{y}^{(s)}$, where $\hat{y}^{(s)} = \mathbf{M}_{c^{(s)}} \tilde{\mathbf{q}}$ is the output of CAN2^(s) before binarization.

3 Experimental Results and Analysis

3.1 Experimental Setting

We have used 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 have examined five texts of Japanese words: $W = \{/daigaku/, /fukuokaken/, /gakusei/, /kikai/, /kyukodai/\}$, where each utterance duration of the words is about 1s. For each speaker and each text, we have ten samples of speech data, $L = \{1, 2, \dots, 10\}$. Namely, we have speech data $x = x_{s,w,l}$ for $s \in S$, $w \in W$ and $l \in L$. In the following, let $s = s_i$ ($\in S$) denote the i th speaker, and $s = \bar{s}_i$ denote that s is not s_i .

In order to evaluate the performance of the present method, we use the leave one-set out cross-validation (LOOCV). Precisely, for text-dependent tasks, we evaluate the performance with test dataset $X(S, w, l) = \{x_{s,w,l} \mid s \in S\}$ and training dataset $X(S, w, \bar{L}) = \{x_{s,w,i} \mid s \in S, i \in \bar{L} \setminus \{l\}\}$ for each $w \in W$ and $l \in L$. We obtain the mean error ratio E_v for verification and E_i for identification over $l \in L$ for each $w \in W$. On the other hand, for text-independent tasks, we use test dataset $X(S, w, l)$ and training dataset $X(S, W_{\bar{w}}, \bar{L}) = \{x_{s,u,i} \mid s \in S, u \in W \setminus \{w\}, i \in \bar{L} \setminus \{l\}\}$ for each $w \in W$ and $l \in L$.

To obtain the poles of speech data, we use the bagging CAN2 with $N = 24$ units, $b = 20$ bags, and the prediction order $k = 8$. For feature vectors of pole distribution, we use $(r_{\max}, n_r, n_\theta) = (2, 2, 18)$. For learning and matching feature vectors \mathbf{q} , we use CAN2^(s) with $N = 20$ units for each $s \in S$. It is important to know the effect of these parameter values, which we omit in this paper owing to the space limit.

3.2 Experimental Results

We show the error ratios E_v and E_i in Table 1, where the results are slightly different from those in [4] because we modify some parameter values to obtain better and stable performance. Here, in order to evaluate the performance of the speaker identification shown in 2.5, we calculate $E_{i,\text{rand}} = (|S| - 1)E_v / (1 - E_v + (|S| - 1)E_v)$ which denotes the error ratio of a random identification where the speaker ID is chosen randomly from s of CAN2^(s) ($s \in S$) with $v^{(s)} = 1$ (indicating acceptance of verification). From the table we can see that the identification method shown in 2.5 works well because $E_i < E_{i,\text{rand}}$ for almost all cases.

Table 1. Error ratios E_v , E_i and $E_{i,\text{rand}}$. The unit of the ratios is %.

text	text-dependent			text-independent		
	E_v	E_i	$(E_{i,\text{rand}})$	E_v	E_i	$(E_{i,\text{rand}})$
/daigaku/	0.80	2.00	(3.13)	10.80	20.00	(32.63)
/fukuokaken/	2.80	8.00	(10.33)	20.80	48.00	(51.23)
/gakusei/	4.00	10.00	(14.29)	16.40	42.00	(43.97)
/kikai/	4.00	10.00	(14.29)	16.40	56.00	(43.97)
/kyukodai/	0.40	0.00	(1.58)	16.40	46.00	(43.97)
mean	2.40	6.00	(8.96)	16.16	42.40	(43.53)

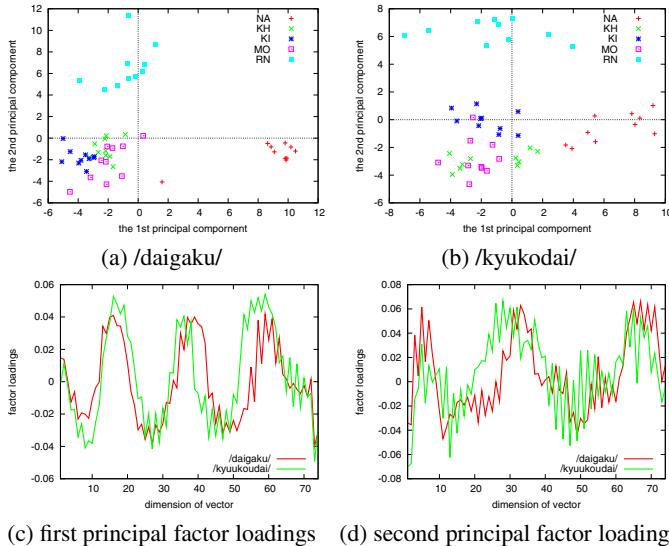


Fig. 3. Result of PCA of the feature vectors q generated from $X(S, w, L)$ for $w=$ /daigaku/ and /kyukodai/. (a) and (b) show the first and the second principal components for /daigaku/ and /kyukodai/, respectively, and (c) and (d) show the first and the second principal factor loadings, respectively.

It is not so easy to compare the error ratios in Table 1 with those by the previous works shown in [7], because they depend on many factors, such as utterance duration, and most of the previous works use more than 1s duration texts. However, $E_v = 16.16\%$ for the text-independent speaker verification does not seem to be so worse than $E_v = 11\%$ to 16% obtained by D.Reynalds work (1996) for 3s duration texts [7]. This indicates that the present method may successfully extract text-independent features of speakers via the feature vectors of pole distribution.

3.3 Principal Component Analysis of the Feature Vectors of Pole Distribution

We apply the PCA (principal component analysis) to the feature vectors q of pole distribution generated from the dataset $X(S, w, L)$ for $w=$ /daigaku/ and /kyukodai/, and we show the results in Fig. 3. From (a) and (b), we can see that the speakers NA and

RN, respectively, are characterized by the first and the second principal components, for both words /daigaku/ and /kyukodai/. Moreover, from (c) and (d), we can see that the first and the second principal factor loadings, respectively, are very similar for both words. These indicate that the principal components and the factor loadings are dependent on the speakers much more than the words. Furthermore, since we have used the test words not involved in the training words for the text-independent tasks, the same level of error ratio is expected for other words not used in the above experiments of text-independent speaker recognition.

3.4 Bayesian Inference for Multistep Verification and Identification

Considering speaker verification, the probability of the output $v^{\langle s_i \rangle}$ of the classifier CAN2 $^{\langle s_i \rangle}$ for a text spoken by a speaker s is given by

$$p(v^{\langle s_i \rangle} | s) = \begin{cases} 1 - E_v & \text{if } (s = s_i) \wedge (v^{\langle s_i \rangle} = 1) \text{ or } (s = \bar{s}_i) \wedge (v^{\langle s_i \rangle} = -1) \\ E_v & \text{if } (s = s_i) \wedge (v^{\langle s_i \rangle} = -1) \text{ or } (s = \bar{s}_i) \wedge (v^{\langle s_i \rangle} = 1), \end{cases} \quad (8)$$

where E_v denotes the mean verification error ratio for all s_i and s in S , or

$$E_v \triangleq \frac{1}{|S|} \sum_{s_i \in S} \left(p(v^{\langle s_i \rangle} = -1 | s_i) + \sum_{s \in S \setminus \{s_i\}} p(v^{\langle s \rangle} = 1 | s_i) \right). \quad (9)$$

Here, note that $p(v^{\langle s_i \rangle} = 1 | s) + p(v^{\langle s_i \rangle} = -1 | s) = 1$ for every s and s_i in S . From the Bayes' rule, the posterior probability of a speaker s ($= s_i$ or \bar{s}_i) for a verified output sequence $v_{1:n}^{\langle s_i \rangle} = v_1^{\langle s_i \rangle}, \dots, v_n^{\langle s_i \rangle}$ from CAN2 $^{\langle s_i \rangle}$ is given by

$$p(s | v_{1:n}^{\langle s_i \rangle}) = \frac{p(s)p(v_{1:n}^{\langle s_i \rangle} | s)}{p(s_i)p(v_{1:n}^{\langle s_i \rangle} | s_i) + p(\bar{s}_i)p(v_{1:n}^{\langle s_i \rangle} | \bar{s}_i)}, \quad (10)$$

where we assume the prior probability as $p(s_i) = 1/|S|$ and $p(\bar{s}_i) = 1 - 1/|S|$. Then, the posterior verification and identification error ratios are derived as

$$E_{v,\text{post}} \triangleq \frac{p(\bar{s}_* | v_{1:n}^{\langle s_* \rangle}) + \sum_{s \in S \setminus \{s_*\}} p(s | v_{1:n}^{\langle s \rangle})}{|S|}, \quad (11)$$

$$E_{i,\text{post}} \triangleq \frac{\sum_{s \in S \setminus \{s_*\}} p(s | v_{1:n}^{\langle s \rangle})}{p(s_* | v_{1:n}^{\langle s_* \rangle}) + \sum_{s \in S \setminus \{s_*\}} p(s | v_{1:n}^{\langle s \rangle})}, \quad (12)$$

where s_* indicates unknown correct speaker, and the above $E_{i,\text{post}}$ is derived for the random identification using $v^{\langle s \rangle}$ as shown in 3.2. Here, note that it is expected that we may have smaller identification error when we use the identification using $\hat{y}^{\langle s \rangle}$ as shown in 2.5 but there are several ways to use them and we could not have derived the probability, which is for our future studies.

Now, let n_A and n_R ($= n - n_A$), respectively, denote the accepted and rejected numbers in the verified sequence $v_{1:n}^{\langle s \rangle}$. Then, it is estimated that $n_A = n(1 - E_v)$ and $p(v_{1:n}^{\langle s \rangle} | s_*) = (1 - E_v)^{n_A} (E_v)^{n_R}$ for $s = s_*$, while $n_A = nE_v$ and $p(v_{1:n}^{\langle s \rangle} | s_*) =$

Table 2. Posterior error ratios $E_{v,\text{post}}$ and $E_{i,\text{post}}$ of multistep recognition estimated for text-dependent ($E_v = 2.40\%$) and text-independent ($E_v = 16.16\%$) verification shown in Table II

n	text-dependent				text-independent					
	0	1	2	3	0	1	2	3	4	5
$E_{v,\text{post}}$	32.00	4.20	0.14	0.00	32.00	22.70	12.04	4.95	1.77	0.60
$E_{i,\text{post}}$	80.00	10.52	0.34	0.01	80.00	56.76	30.11	12.38	4.43	1.50

$(E_v)^{n_A}(1 - E_v)^{n_R}$ for $s \neq s_*$. Thus, from $E_v = 2.40\%$ and $E_v = 16.16\%$ in Table II, we have the posterior error ratios as shown in Table 2. From the table, we can see that the error rate 5% is expected to be achieved at $n = 1$ and 2 for text-dependent verification and identification, respectively, and $n = 3$ and 4 for text-independent ones, respectively. These numbers of steps for accurate recognition indicate some reasonable interactive speaker recognition systems.

4 Conclusion

We have shown a method of speaker recognition which uses feature vectors of pole distribution derived via the bagging CAN2. By means of analysing the experimental results using real speech data, we show the effectiveness of the present method. Especially, the multistep recognition using Bayes' rule indicates some reasonable interactive speaker recognition systems which we would like to develop in our future research studies.

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., Ueno, T., Sawa, M.: A batch learning method for competitive associative net and its application to function approximation. In: Proc. SCI 2004, pp. 24–28 (2004)
2. 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 (2008)
3. Kurogi, S.: Improving generalization performance via out-of-bag estimate using variable size of bags. J. Japanese Neural Network Society 16(2), 81–92 (2009)
4. Kurogi, S., Sato, S., Ichimaru, K.: Speaker Recognition Using Pole Distribution of Speech Signals Obtained by Bagging CAN2. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009. LNCS, vol. 5863, pp. 622–629. Springer, Heidelberg (2009)
5. Ahalt, A.C., Krishnamurthy, A.K., Chen, P., Melton, D.E.: Competitive learning algorithms for vector quantization. Neural Networks 3, 277–290 (1990)
6. Kohonen, T.: Associative Memory. Springer, Heidelberg (1977)
7. Campbell, J.P.: Speaker Recognition: A Tutorial. Proc. the IEEE 85(9), 1437–1462 (1997)
8. 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)
9. Hasan, M.R., Jamil, M., Rabban, M.G., Rahman, M.S.: Speaker identification using Mel frequency cepstral coefficients. In: Proc. ICEC 2004, pp. 565–568 (2004)
10. Bocklet, T., Shriberg, E.: Speaker recognition using syllable-based constraints for cepstral frame selection. In: Proc. ICASSP (2009)
11. Breiman, L.: Bagging predictors. Machine Learning 26, 123–140 (1996)

Sparse and Low-Rank Estimation of Time-Varying Markov Networks with Alternating Direction Method of Multipliers

Jun-ichiro Hirayama¹, Aapo Hyvärinen², and Shin Ishii¹

¹ Graduate School of Informatics, Kyoto University
Gokasho, Uji, Kyoto 611-0011, Japan

<http://hawaii.sys.i.kyoto-u.ac.jp/~junich-h/>

² Department of Mathematics and Statistics/Department of Computer Science and
HIIT, University of Helsinki, Helsinki, Finland

Abstract. Several authors have recently proposed sparse estimation techniques for *time-varying* Markov networks, in which both graph structures and model parameters may change with time. In this study, we extend a previous approach with a low-rank assumption on the matrix of parameter sequence, utilizing a recent technique of nuclear norm regularization. This can potentially improve the estimation performance particularly in such cases that the local smoothness assumed in previous studies do not really hold. Then, we derive a simple algorithm based on the alternating direction method of multipliers (ADMM) which can effectively utilize the separable structure of our convex minimization problem. With an artificially-generated dataset, its superior performance in structure learning is demonstrated.

1 Introduction

Markov networks (MNs), or Markov random fields (MRFs), are basic statistical models for representing dependency networks of multiple random variables, and have many applications in various fields related to machine learning. A MN describes the structure of conditional (in)dependences by an undirected graph, and defines the full probability distribution with parameterized feature functions associated with their nodes and edges (or more generally cliques) [16]. Two fundamental examples of MNs are the Gaussian Graphical Model and the Ising model, the latter of which we focus on in this study.

Sparse estimation techniques, based typically on ℓ_1 -norm regularization or its variants, are probably the most practical approaches to training MNs from data [18][4][11]. Here, we consider to estimate both the graph structure and model parameters. Since a graph structure is often completely specified by which parameters or parameter groups (under a fully-connected graph) are zero, one can estimate both the structure and parameters jointly by ensuring irrelevant parameter values to shrink exactly to zero owing to sparse estimation.

Recently, several authors have proposed sparse estimation techniques for *time-varying* MNs [19][13][21]. These studies assumed both the graph structure and

model parameters may change with time smoothly or disruptly, which is one of potential non-stationarities of real dependency networks. These studies basically showed that, if the changes are relatively slow (in smooth cases) or not very frequent (in disrupt cases), time-varying MNs can be estimated by utilizing certain mechanisms of smoothing or change-point detection in addition to sparseness.

When the dimensionality of parameter space is large, however, it may be difficult to obtain as many samples in each (approximately) stationary domain as required for effective estimation. Then, it could be unrealistic to estimate large MNs based only on the local smoothness or stationarity assumption, particularly when the changes are relatively fast or frequent.

To overcome this issue, in this study, we extend a previous approach [13] (based on both sparsity and local smoothness) to incorporate a further assumption that the parameter matrix, whose column is a parameter vector at a single time step, have a relatively low rank. This is done by heuristically combining a recent technique of nuclear norm (or trace norm) [9] regularization with sparse estimation. Then, we derive a simple algorithm based on the alternating direction method of multipliers (ADMM) [24], which can effectively utilize the separable structure of our (convex) minimization problem.

2 Pseudolikelihood Estimation of Sparse Markov Networks

Let $\mathbf{y} = (y_1, y_2, \dots, y_D)^\top \in \{-1, 1\}^D$ be a D -dimensional binary random vector that takes each of the 2^D spin configurations. Then, a general form of Ising model is given by

$$p(\mathbf{y}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp \left(\sum_{i < j} \theta_{ij} y_i y_j + \sum_i \theta_{ii} y_i \right), \quad (1)$$

where $Z(\boldsymbol{\theta}) = \sum_{\mathbf{y}} \exp \left(\sum_{i < j} \theta_{ij} y_i y_j + \sum_i \theta_{ii} y_i \right)$ is the partition function. The first summation in the exponent is over all pairs (i, j) that satisfy $i < j$, and we put all the $C = D(D+1)/2$ parameters in a vector $\boldsymbol{\theta} \in \mathbb{R}^C$. The corresponding undirected graph to this model has nodes $i = 1, 2, \dots, D$, and any pair of nodes (i, j) is connected if and only if θ_{ij} is non-zero.

Since exact computation of the likelihood is infeasible unless D is sufficiently small, which is due to the intractability of partition function $Z(\boldsymbol{\theta})$, we often employ other estimation techniques than maximum likelihood (ML) or its penalized variants. One popular approach is the maximum pseudolikelihood (MPL) [34, 12], particularly with ℓ_1 -regularization for determining graph structures [11]. This can be formulated as a minimization problem as follows:

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad -\frac{1}{N} \sum_{n=1}^N \sum_{i=1}^D \log p(y_i^n | \mathbf{y}_{\setminus i}^n; \boldsymbol{\theta}) + \|\boldsymbol{\lambda} \circ \boldsymbol{\theta}\|_1, \quad (2)$$

where $\|\cdot\|_1$ denotes the ℓ_1 -norm (*i.e.*, summation of the absolute value over all the elements), $\boldsymbol{\lambda} \in [0, \infty)^C$ the vector of regularization coefficients, and \circ

element-wise multiplication. In practice, we typically use $\lambda_{ii} = 0$ and $\lambda_{ij} = \lambda$ ($i \neq j$) with a common λ .

3 Sparse and Low-Rank Estimation of Time-Varying MNs

Now, let us consider time-varying MNs with parameters $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \dots, \boldsymbol{\theta}^N \in \mathbb{R}^C$ each for single time step. Each observation \mathbf{y}^n is assumed to be generated instantaneously from $p(\mathbf{y}^n | \boldsymbol{\theta} = \boldsymbol{\theta}^n)$. We also put all these parameter vectors into a matrix $\boldsymbol{\Theta} = (\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^N) \in \mathbb{R}^{C \times N}$. Clearly, we need some assumptions about $\boldsymbol{\Theta}$ in addition to the sparsity, in order to estimate it effectively.

One basic approach is to introduce temporal smoothness. In this study, we take a kernel smoothing approach studied in [2][13], in which estimation losses (e.g., the negative logarithm of single-sample pseudolikelihood in our case) are temporally smoothed by locally taking a weighted average with a window function $\varphi(\cdot)$. That is, we minimize the following smoothed average loss,

$$f(\boldsymbol{\Theta}) = -\frac{1}{N} \sum_{n=1}^N \sum_{m=1}^N \varphi(|m-n|) \sum_{i=1}^D \log p(y_i^m | \mathbf{y}_{\setminus i}^m; \boldsymbol{\theta}^n), \quad (3)$$

plus regularization terms as shown below. If only the ℓ_1 -regularization is used (i.e., $\eta = 0$ below), this is a time-varying extension of sparse pseudolikelihood estimation [11], which is basically the same as a method in [13].

3.1 Regularization with Both ℓ_1 and Nuclear Norms

In this study, we further assume $\boldsymbol{\Theta}$ has a relatively low rank, i.e., $K = \text{rank}(\boldsymbol{\Theta}) \leq \min\{C, N\}$, in order to effectively reduce the degree of freedom in the parameter matrix. This implies every $\boldsymbol{\theta}^n$ can be embedded in a low-dimensional subspace of \mathbb{R}^C , which can be given a basis $\{\mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^K\}$ and thus we may write $\boldsymbol{\theta}^n = s_1^n \mathbf{a}^1 + s_2^n \mathbf{a}^2 + \dots + s_K^n \mathbf{a}^K$ with real-valued coefficients s_k^n 's. Then, it is implicitly assumed here that all the parameter vectors are generated from a relatively small number of underlying “patterns” or “causes.” Such an assumption would be useful in practice even when temporal smoothness does not really hold.

To incorporate the low-rank assumption of $\boldsymbol{\Theta}$, we employ the nuclear norm regularization [9][5] instead of explicitly constraining the rank of $\boldsymbol{\Theta}$ which leads to a non-convex problem. Our minimization problem is then given by

$$\underset{\boldsymbol{\Theta}}{\text{minimize}} \quad f(\boldsymbol{\Theta}) + \|\boldsymbol{\Lambda} \circ \boldsymbol{\Theta}\|_1 + \eta \|\boldsymbol{\Theta}\|_*, \quad (4)$$

where $\|\cdot\|_1$ is again the vector ℓ_1 -norm and $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}, \boldsymbol{\lambda}, \dots, \boldsymbol{\lambda})$. The last term is newly introduced here, where $\|\cdot\|_*$ is the nuclear norm (summation of all the singular values) and $\eta \geq 0$ is its coefficient. Since all the three terms are convex, the problem (4) itself is also convex. Although the joint regularization with both ℓ_1 and nuclear norms is heuristic, but it turns out to be promising for making $\boldsymbol{\Theta}$ both sparse and low-rank in our later experiment.

3.2 Alternating Direction Method of Multipliers (ADMM)

To solve the minimization problem above, we employ the alternating direction method of multipliers (ADMM) [2,6], which is also known as the Split Bregman Iteration (SBI) method [10,7] and has recently been actively studied in several application areas [20,9].

Consider the following problem with two convex functions ϕ and γ :

$$\begin{aligned} & \text{minimize } \phi(\mathbf{x}) + \gamma(\mathbf{z}) \\ & \text{subject to } \mathbf{J}\mathbf{x} = \mathbf{z} \end{aligned}, \quad (5)$$

with respect to real-valued vectors \mathbf{x} and \mathbf{z} , where \mathbf{J} is a matrix of appropriate size for representing linear constraints. To solve this, the ADMM algorithm iterates the following three steps from initial conditions $\mathbf{x}^{(0)}, \mathbf{z}^{(0)}, \mathbf{r}^{(0)}$ until a certain convergence criterion is satisfied:

$$\mathbf{x}^{(t)} = \operatorname{argmin}_{\mathbf{x}} \left\{ \phi(\mathbf{x}) + \langle \mathbf{r}^{(t-1)}, \mathbf{J}\mathbf{x} \rangle + \frac{\alpha}{2} \|\mathbf{J}\mathbf{x} - \mathbf{z}^{(t-1)}\|_2^2 \right\}, \quad (6a)$$

$$\mathbf{z}^{(t)} = \operatorname{argmin}_{\mathbf{z}} \left\{ \gamma(\mathbf{z}) - \langle \mathbf{r}^{(t-1)}, \mathbf{z} \rangle + \frac{\alpha}{2} \|\mathbf{J}\mathbf{x}^{(t)} - \mathbf{z}\|_2^2 \right\}, \quad (6b)$$

$$\mathbf{r}^{(t)} = \mathbf{r}^{(t-1)} + \alpha(\mathbf{J}\mathbf{x}^{(t)} - \mathbf{z}^{(t)}), \quad (6c)$$

where $\langle \cdot, \cdot \rangle$ and $\|\cdot\|_2$ are the standard inner product and the ℓ_2 -norm, and $\alpha > 0$ is any positive constant. The superscript (t) denotes the iteration number. Each single iteration step of ADMM can be seen as a single cycle in an alternating minimization of the Augmented Lagrangian [2,15]

$$\mathcal{L}_\alpha(\mathbf{x}, \mathbf{z}, \mathbf{r}) = \phi(\mathbf{x}) + \gamma(\mathbf{z}) + \langle \mathbf{r}, \mathbf{J}\mathbf{x} - \mathbf{z} \rangle + \frac{\alpha}{2} \|\mathbf{J}\mathbf{x} - \mathbf{z}\|_2^2, \quad (7)$$

with respect to the primal vectors \mathbf{x} and \mathbf{z} , followed by the dual update of \mathbf{r} . The constant α can then be chosen rather freely, as in the standard augmented Lagrangian method [15]. This is in contrast to the penalty method [15] where the strength of the penalty has to be large, which can cause numerical instability. It has been shown that the above procedure converges to the optimal solution, under certain conditions about the accuracy of the two minimization steps [6].

3.3 Derivation of ADMM for Our Problem

We reformulate our problem (4) to apply the ADMM in a similar manner to that in [2,8] as follows. As an important consequence, this allows us to deal with the two different regularization terms as well as the loss term separately for each other, which leads to quite a simple iterative algorithm. First, we introduce auxiliary variables $\mathbf{Z}_1, \mathbf{Z}_2$ and \mathbf{Z}_3 each in $\mathbb{R}^{C \times N}$ and define

$$\gamma(\mathbf{Z}) = f(\mathbf{Z}_1) + \|\Lambda \circ \mathbf{Z}_2\|_1 + \eta \|\mathbf{Z}_3\|_*, \quad (8)$$

with $\mathbf{Z} = (\mathbf{Z}_1^\top, \mathbf{Z}_2^\top, \mathbf{Z}_3^\top)^\top$. Then, we can rewrite the problem (4) equivalently as

$$\begin{aligned} & \underset{\Theta, \mathbf{Z}}{\text{minimize}} \quad \phi(\Theta) + \gamma(\mathbf{Z}) \\ & \text{subject to } \mathbf{J}\Theta = \mathbf{Z} \end{aligned}, \quad (9)$$

where $\phi(\Theta) = 0$, $\mathbf{J} = (\mathbf{I}_C, \mathbf{I}_C, \mathbf{I}_C)^\top$ and \mathbf{I}_C is a $C \times C$ unit matrix. With the dual variables $\mathbf{R} = (\mathbf{R}_1^\top, \mathbf{R}_2^\top, \mathbf{R}_3^\top)^\top$, therefore, the ADMM procedure can be straightforwardly applied to this problem. The first step, Eq. (6a), is now a quadratic minimization and thus has a closed-form solution:

$$\Theta^{(t)} = \frac{1}{3} \left(\mathbf{Z}_1^{(t-1)} + \mathbf{Z}_2^{(t-1)} + \mathbf{Z}_3^{(t-1)} \right) - \frac{1}{3\alpha} \left(\mathbf{R}_1^{(t-1)} + \mathbf{R}_2^{(t-1)} + \mathbf{R}_3^{(t-1)} \right). \quad (10)$$

The second step, Eq. (6b), can be separately written as

$$\mathbf{Z}_1^{(t)} = \underset{\mathbf{Z}_1}{\text{argmin}} \left\{ f(\mathbf{Z}_1) + \frac{\alpha}{2} \|\mathbf{Z}_1 - \mathbf{G}_1^{(t)}\|_F^2 \right\}, \quad (11a)$$

$$\mathbf{Z}_2^{(t)} = \underset{\mathbf{Z}_2}{\text{argmin}} \left\{ \|\Lambda \circ \mathbf{Z}_2\|_1 + \frac{\alpha}{2} \|\mathbf{Z}_2 - \mathbf{G}_2^{(t)}\|_F^2 \right\}, \quad (11b)$$

$$\mathbf{Z}_3^{(t)} = \underset{\mathbf{Z}_3}{\text{argmin}} \left\{ \eta \|\mathbf{Z}_3\|_* + \frac{\alpha}{2} \|\mathbf{Z}_3 - \mathbf{G}_3^{(t)}\|_F^2 \right\}, \quad (11c)$$

where $\mathbf{G}_q^{(t)} = \Theta^{(t)} + \alpha^{-1} \mathbf{R}_q^{(t-1)}$ ($q = 1, 2, 3$) and $\|\cdot\|_F$ is the Frobenius norm. Then, $\mathbf{Z}_1^{(t)}$ can be numerically obtained by using any standard unconstrained optimization technique; $\mathbf{Z}_2^{(t)}$ and $\mathbf{Z}_3^{(t)}$ can be obtained as follows:

$$\mathbf{Z}_2^{(t)} = \text{Soft} \left(\Theta^{(t)} + \alpha^{-1} \mathbf{R}_2^{(t-1)}, \Lambda/\alpha \right), \quad (12a)$$

$$\mathbf{Z}_3^{(t)} = \text{Svt} \left(\Theta^{(t)} + \alpha^{-1} \mathbf{R}_3^{(t-1)}, \eta/\alpha \right). \quad (12b)$$

Here, $\text{Soft}(\cdot, \cdot)$ is an element-wise application of the soft-thresholding operator $\text{soft}(a, b) = \text{sign}(a) \max(|a| - b, 0)$ and $\text{Svt}(\cdot, \cdot)$ denotes the “singular value thresholding” [5, 17] operator given by

$$\text{Svt}(\mathbf{A}, b) = \mathbf{U} \text{diag}(\text{Soft}(\boldsymbol{\sigma}, b)) \mathbf{V}^\top, \quad (13)$$

where $\mathbf{A} = \mathbf{U} \text{diag}(\boldsymbol{\sigma}) \mathbf{V}^\top$ is the singular value decomposition of \mathbf{A} and $\boldsymbol{\sigma}$ is the vector of singular values. Finally, the dual update step, Eq. (6c), is given separately for \mathbf{R}_q ($q = 1, 2, 3$) by $\mathbf{R}_q^{(t)} = \mathbf{R}_q^{(t-1)} + \alpha(\Theta^{(t)} - \mathbf{R}_q^{(t-1)})$.

4 An Experiment with Artificial Dataset

We conducted an experiment with an artificial dataset to demonstrate the effect of introducing low-rank regularization. In this experiment, we used a rectangular window function for temporal smoothing:

$$\varphi(|m - n|) = \begin{cases} 1/w & |m - n| \leq \tau \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

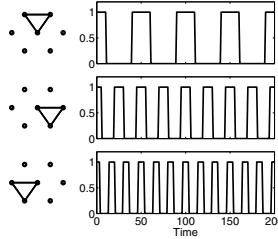


Fig. 1. Three graphs corresponding to the basis elements (left) and the time-series of their coefficients (right) used for generating a sparse parameter time-series that are embedded in a three-dimensional subspace

where $w = 2\tau + 1$. The minimization in the first step of ADMM, Eq. (IIa), was numerically solved by a quasi-newton method [4]. As an initial setting, both the primal and dual variables were all set at zero matrices, and the algorithm was stopped when $\max\{\delta_z^{(t)}, \delta_r^{(t)}\} \leq \epsilon = 10^{-5}$ was achieved, where $\delta_z^{(t)}$ and $\delta_r^{(t)}$ are the maximum absolute values of all the elements in $\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}$ and $\mathbf{R}^{(t)} - \mathbf{R}^{(t-1)}$, respectively. The constant α was set at 10^{-3} , but to accelerate the final convergence, we multiply α by a factor 1.5 at every iteration after reaching $\delta_r^{(t)} \leq 0.8\epsilon$, similar to [20]. Then, our final estimate of Θ was obtained by explicitly putting zero into all such elements of Θ that corresponding values in \mathbf{Z}_2 are actually zero; the graph structure was then determined according to the non-zero elements.

The dataset was sampled from the Ising model (II) under a time-varying parameter sequence. The dimensionality of observations was $D = 7$ and the length of time-series $N = 200$. The parameter space was then \mathbb{R}^{28} , but every θ^n was constrained to be in a three-dimensional subspace, according to

$$\theta^n = s_1^n \mathbf{a}^1 + s_2^n \mathbf{a}^2 + s_3^n \mathbf{a}^3, \quad (15)$$

and thus $\text{rank}(\Theta) = 3$. Here, we chose the three basis elements, \mathbf{a}^1 , \mathbf{a}^2 and \mathbf{a}^3 , so that they correspond respectively to the three graphs in the left column of Fig. II, and their non-zero elements (*i.e.*, edge weights) were uniformly set at 0.5. The right column of Fig. II also shows time-series of their coefficients, s_1^n , s_2^n and s_3^n , which only took 0 or 1 for simplicity. Finally, each observation \mathbf{y}^n at single time step was generated randomly from $p(\mathbf{y}^n | \theta^n)$ by directly calculating joint probability tables over the 2^7 possible values. We examined several values for the time-window width at $w = 5, 9, 13$ and 17 ($\tau = 2, 4, 6$ and 8). The regularization coefficient for the ℓ_1 -norm was set by $\lambda_{ii} = 0$ and $\lambda_{ij} = \lambda$ ($i \neq j$) with various values of λ .

Figure 2 shows the Receiver Operating Characteristic (ROC) curves for some η at several choices of w , which was drawn by regarding the structure recovery

¹ We used a Matlab implementation of limited-memory BFGS by Mark Schmidt, available from <http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html>

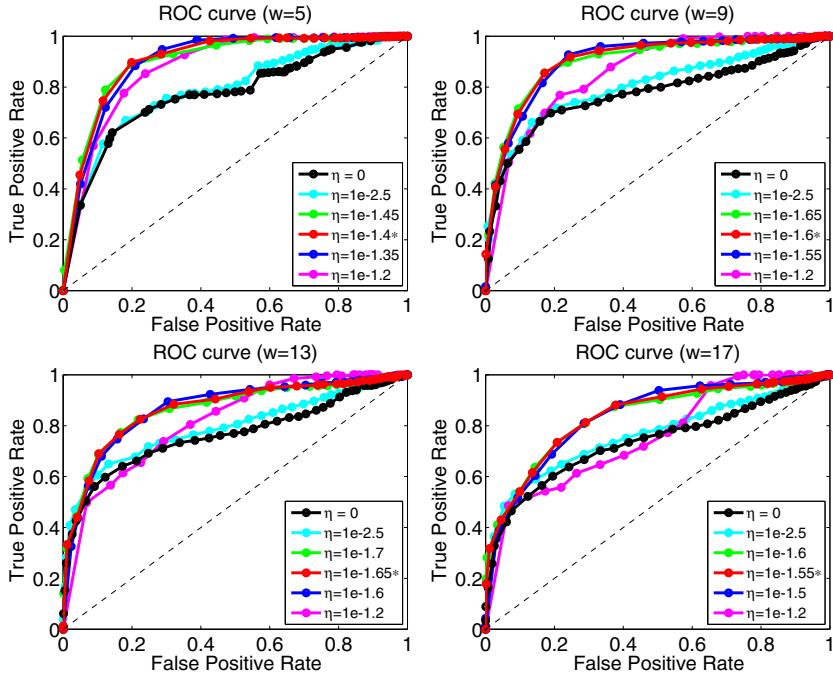


Fig. 2. ROC curves for several choices of window-width (w) as well as for some η (weight of nuclear norm penalty). The '*' in the legend indicates the best η value in terms of AUC shown in Fig. 3.

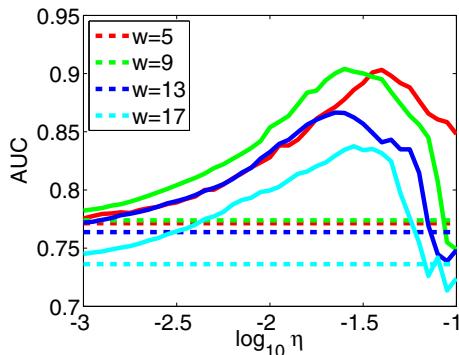


Fig. 3. Area Under the ROC Curve versus strength of low-rank regularization ($\log \eta$). The horizontal dashed lines indicate the AUC values when $\eta = 0$ for each w .

problem as a binary classification. In other words, from the final estimate of Θ obtained as above, we have a binary classifier which says whether a single weight θ_{ij}^n belongs to the class of non-zero weights or to that of weights equal to zero

for each $i \neq j$ and n . The performance of this detection can be quantified by an ROC curve. This figure illustrates that our method with some choices of non-zero η clearly outperformed that with $\eta = 0$. Note that when $\eta = 0$, our method almost reduces to the previous method [13] based only on the sparsity and local smoothness. Figure 3 plots the Area Under the ROC Curve (AUC) versus $\log_{10} \eta$. This also shows that the performance of structure recovery was improved by introducing low-rank regularization within an appropriate range of η .

5 Summary

We have proposed a new “sparse and low-rank” estimation framework of time-varying MNs, particularly using an Ising model as a concrete example of MNs. A simple algorithm was derived by applying ADMM to our convex minimization problem based both on the ℓ_1 and nuclear norm regularizations. An experiment with artificially-generated dataset showed the low-rank regularization can potentially improve the estimation performance over those by the existing approaches based only on sparsity and local smoothness. We are currently conducting a real-data experiment to fully confirm the effectiveness of our method.

References

1. Banerjee, O., et al.: Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *JMLR* 9, 485–516 (2008)
2. Bertsekas, D.P., Tsitsiklis, J.N.: *Parallel and distributed computation: Numerical methods*. Prentice-Hall, Inc., Englewood Cliffs (1989)
3. Besag, J.: Statistical analysis of non-lattice data. *J R Stat Soc Series D* 24(3), 179–195 (1975)
4. Besag, J.: Efficiency of pseudo-likelihood estimation for simple Gaussian fields. *Biometrika* 64, 616–618 (1977)
5. Cai, J.-F., et al.: A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20(4), 1956–1982 (2010)
6. Eckstein, J., Bertsekas, D.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming* 5, 293–318 (1992)
7. Esser, E.: Applications of Lagrangian-based alternating direction methods and connections to split Bregman. *CAM Reports* 09-31, UCLA (2009)
8. Figueiredo, M., Bioucas-Dias, J.: Restoration of Poissonian images using alternating direction optimization. *IEEE Transactions on Image Processing* (preprint) (2010), <http://arxiv.org/abs/1001.2244>
9. Fazel, M., Hindi, H., Boyd, S.: Rank minimization and applications in system theory. In: *Proceedings American Control Conference*, pp. 3273–3278 (2004)
10. Goldstein, T., Osher, S.: The split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences* 2(2), 323–343 (2009)
11. Höfling, H., Tibshirani, R.: Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods. *JMLR* 10, 883–906 (2009)
12. Hyvärinen, A.: Consistency of pseudolikelihood estimation of fully visible Boltzmann machines. *Neural Comput.* 18(10), 2283–2292 (2006)

13. Kolar, M., et al.: Estimating time-varying networks. *Annals of Applied Statistics* 4(1), 94–123 (2010)
14. Lee, S.-I., et al.: Efficient structure learning of Markov networks using L1-regularization. In: NIPS 2006 (2006)
15. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, Heidelberg (1999)
16. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
17. Toh, K., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems, National Univ. of Singapore (Preprint 2009)
18. Wainwright, M.J., et al.: High-dimensional graphical model selection using ℓ_1 -regularized logistic regression. In: NIPS (2006)
19. Xuan, X., Murphy, K.: Modeling changing dependency structure in multivariate time series. In: ICML 2007, pp. 1055–1062 (2007)
20. Yuan, X.M.: Alternating direction methods for sparse covariance selection, (2009), http://www.optimization-online.org/DB_FILE/2009/09/2390.pdf (Preprint)
21. Zhou, S., et al.: Time varying undirected graphs. *Machine Learning* (2010)

Nearest Hit-Misses Component Analysis for Supervised Metric Learning

Wei Yang, Kuanquan Wang, and Wangmeng Zuo

Biocomputing Research Centre, School of Computer Science and Technology,
Harbin Institute of Technology, Harbin, 150001, China

Abstract. Metric learning is the task of learning a distance metric from training data that reasonably identifies the important relationships between the data. An appropriate distance metric is of considerable importance for building accurate classifiers. In this paper, we propose a novel supervised metric learning method, nearest hit-misses component analysis. In our method, the margin is first defined with respect to the nearest hits (nearest neighbors from the same class) and the nearest misses (nearest neighbors from the different class), and then the distance metric is trained by maximizing the margin while minimizing the distance between each sample and its nearest hits. We further introduce a regularization term to alleviate overfitting. Moreover, the proposed method can perform metric learning and dimensionality reduction simultaneously. Comparative experiments with the state-of-the-art metric learning methods on various real-world data sets demonstrate the effectiveness of the proposed method.

Keywords: metric learning, dimensionality reduction, k -nearest neighbor, supervised learning, distance metric.

1 Introduction

Distance metric is of considerable importance in machine learning and pattern recognition. The classification performance of many learning algorithms, such as k -nearest neighbor, radial basis function networks, and many kernel methods, often depends significantly on the adopted distance metric over the multidimensional input space. Moreover, appropriate distance metrics are also critical in various real-world applications, including face verification, image retrieval, image classification, visual identification, and text categorization. It is natural that different applications often require different distance metrics. Since distance metric is problem-specific and designing a good distance metric manually is extremely difficult, learning a proper distance metric from training data has caught a great deal of attention from machine learning community.

Many supervised metric learning algorithms have been developed in order to induce a good distance metric from labeled samples. For example, linear discriminant analysis (LDA) [12] computes the linear transformation by maximizing the ratio of the between-class variance to the within-class variance. Discriminant

adaptive nearest neighbor (DANN) [3] uses a local linear discriminant analysis to estimate the distance metric. Recently, Xing et al. [4] formulate distance metric learning as a convex optimization problem. Relevant component analysis (RCA) [5] estimates a full-rank Mahalanobis metric by employing the equality constraints to downscale global unwanted variability within the data. Hoi et al. [6] improve RCA by exploiting the inequality constraints and introducing kernel tricks. Local Fisher discriminant analysis (LFDA) [7], which combines the ideas of LDA and locality-preserving projection, induces a linear transformation by maximizing between-class separability and preserving within-class local structure. Information-theoretic metric learning (ITML) [8] formulates distance metric learning as a particular Bregman optimization problem. Comprehensive survey of distance metric learning methods can be referred to [9].

Nearest neighbor (NN) is one of the most popular pattern classification techniques. Several nearest neighbor-based distance metric learning algorithms, e.g., neighborhood component analysis (NCA) [10], large margin nearest neighbor (LMNN) [11], and large margin component analysis (LMCA) [12], have been successfully developed. NCA learns a linear transformation by directly maximizing the stochastic variant of the expected leave-one-out classification accuracy on the training set. By restricting the learned distance metric to be low rank, NCA can also be used for dimensionality reduction. LMNN formulates distance metric learning into a semi-definite programming problem by forcing that the k nearest neighbors of each training sample belong to the same class while examples from different classes are separated by a large margin. LMCA further extends LMNN by introducing low rank distance metric and kernel trick. However, there still exist some weaknesses for these k NN-based metric learning methods. For example, NCA has high computational cost and is computationally infeasible for high-dimensional datasets. For LMNN and LMCA, the target neighbors of each training sample are determined by the k nearest neighbors in original input space and do not change during the learning process. For some applications with irrelevant and redundant features, however, the nearest neighbor in original feature space may not be true in the learned metric space. In this case, the requirement of maintaining large margin is too strong to obtain good distance metric. Moreover, all of them lack a mechanism to deal with high-dimensional applications with small sample size, in which overfitting would happen and an appropriate regularizer should be required.

In this paper, we present a novel nearest neighbor-based metric learning method, nearest hit-misses component analysis (NHMCA). NHMCA learns a linear transformation by maximizing the margin while minimizing the distance between each sample and its k nearest hits. Moreover, a Frobenius norm regularizer is introduced to avoid the overfitting of the learned metric. NHMCA can also perform dimensionality reduction and thus can deal with the high-dimensional problem. Experimental results show that, our method performs better than the existing metric learning methods, e.g., NCA and LMNN.

The remainder of this paper is organized as follows: Section 2 describes the proposed distance metric learning method. Section 3 presents experimental results on various datasets. Finally, Section 4 concludes this paper.

2 Nearest Hit-Misses Component Analysis

We begin this section with several notational definitions. Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denotes a training set of N labeled samples, where $x_i \in R^d$ is a d -dimensional sample and $y_i \in \{1, \dots, C\}$ is the associated class labels. Our goal is to learn a Mahalanobis distance metric for optimal KNN classification. Specifically, the metric is parameterized by a linear transformation of the input space. If we denote the transformation by a matrix \mathbf{L} of size $r \times d$ ($r \leq d$), the squared Mahalanobis distance between samples \mathbf{x}_i and \mathbf{x}_j can be computed by:

$$\|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{L}^T \mathbf{L} (\mathbf{x}_i - \mathbf{x}_j) \quad (1)$$

Given an unclassified sample point \mathbf{x} , KNN first finds its k nearest neighbors in the training set and then assigns the label by the class that appears most frequently in the k nearest neighbors. Therefore, for robust KNN classification, each training sample \mathbf{x}_i should has the same label with its k nearest neighbors. For each sample \mathbf{x}_i , we call \mathbf{x}_j as a hit if \mathbf{x}_j ($j \neq i$) has the same class label with \mathbf{x}_i , and the nearest hit \mathbf{x}_j ($j \neq i$) is defined as the hit which has the minimum distance with the sample \mathbf{x}_i . Similarly, we call \mathbf{x}_j as a miss if the class label of \mathbf{x}_j ($j \neq i$) is different with \mathbf{x}_i , and the nearest miss \mathbf{x}_j ($j \neq i$) is defined as the miss which has the minimum distance with the sample \mathbf{x}_i . Let $\text{NH}_l(\mathbf{x}_i)$ and $\text{NM}_l(\mathbf{x}_i)$ be the l th nearest hit and miss of \mathbf{x}_i , respectively. The margin of sample \mathbf{x}_i with respect to its j th nearest hit and l th nearest miss is defined as:

$$\rho_{ijl} = \|\mathbf{L}(\mathbf{x}_i - \text{NM}_l(\mathbf{x}_i))\|_2^2 - \|\mathbf{L}(\mathbf{x}_i - \text{NH}_j(\mathbf{x}_i))\|_2^2 \quad (2)$$

where $1 \leq j, l \leq k$. Note that $\text{NH}_j(\mathbf{x}_i)$ and $\text{NM}_l(\mathbf{x}_i)$ are determined by the Mahalanobis distance and the transformation matrix \mathbf{L} affects the margin through the distance metric. Obviously, if the margins of all the samples in the training set are higher than zero, KNN classifier would obtain a robust and perfect classification accuracy. By maximizing the margins of all training samples, the transformation matrix can be obtained by minimizing the following loss function:

$$\sum_{i=1}^N \sum_{jl} \log(1 + \exp(-\rho_{ijl})) \quad (3)$$

Here, the utility function $f(\rho) = \log(1 + \exp(-\rho))$ is used to control the contribution of each margin term to the loss function. Note that in (3) maximizing the margins can also be attained by increasing the distances between each sample and its nearest hits and the distances to its nearest misses simultaneously, where the latter obtain the much larger increase. However, we expect that each training sample and its nearest hits form a compact clustering. Therefore, we

further introduce a term to constrain the distances between each sample and its nearest hits and obtain the following objection function:

$$(1 - \mu) \sum_{i=1}^N \sum_j \|\mathbf{L}(\mathbf{x}_i - \text{NH}_j(\mathbf{x}_i))\|_2^2 + \mu \sum_{i=1}^N \sum_{jl} \log(1 + \exp(-\rho_{ijl})) \quad (4)$$

where $\mu \in [0, 1]$ is a balance parameter trading off the effect between two terms.

For some high dimensional applications with small sample size, such as face recognition and microarray data analysis, the minimization of (4) probably overfits the training data which results in poor generalization performance. To alleviate such problems, an effective strategy is to introduce some restrictions or prior knowledge to regularize the model complexity. In this work, we choose the popular Frobenius norm regularizer. With this regularizer, the objective loss function can be rewritten as:

$$\xi(\mathbf{L}) = (1 - \mu) \sum_{i=1}^N \sum_j \|\mathbf{L}(\mathbf{x}_i - \text{NH}_j(\mathbf{x}_i))\|_2^2 + \mu \sum_{i=1}^N \sum_{jl} \log(1 + \exp(-\rho_{ijl})) + \lambda \|\mathbf{L}\|_F^2 \quad (5)$$

where $\|\mathbf{L}\|_F^2 = \text{tr}(\mathbf{L}^T \mathbf{L})$ is the Frobenius norm and λ is a nonnegative regularization parameter. We minimize the objective function $\xi(\mathbf{L})$ using the steepest gradient descent method. Differentiating $\xi(\mathbf{L})$ with respect to the transformation matrix \mathbf{L} gives the following gradient which we adopt for learning:

$$\begin{aligned} \frac{\partial \xi(\mathbf{L})}{\partial \mathbf{L}} &= 2\mathbf{L} \left[(1 - \mu) \sum_{i=1}^N \sum_j \mathbf{H}_{ij} \mathbf{H}_{ij}^T \right. \\ &\quad \left. + \mu \sum_{i=1}^N \sum_{jl} \frac{\exp(-\rho_{ijl})}{1 + \exp(-\rho_{ijl})} (\mathbf{H}_{ij} \mathbf{H}_{ij}^T - \mathbf{M}_{il} \mathbf{M}_{il}^T) \right] + 2\lambda \mathbf{L} \end{aligned} \quad (6)$$

where $\mathbf{H}_{ij} = (\mathbf{x}_i - \text{NH}_j(\mathbf{x}_i))$ and $\mathbf{M}_{il} = (\mathbf{x}_i - \text{NM}_l(\mathbf{x}_i))$.

The proposed method is named as nearest hit-misses component analysis (NHMCA). Generally speaking, NHMCA needs a matrix of size $r \times d$ to initialize the linear transformation matrix \mathbf{L} . In our work, if the metric learning without dimensionality reduction is performed, we use the identity matrix to initialize NHMCA; otherwise, the matrix consisting of the first r leading components of principal component analysis (PCA) is used. Obviously, NHMCA can perform both metric learning and feature extraction for dimensionality reduction. In particular, when the transformation matrix \mathbf{L} is restricted to be diagonal, NHMCA can also perform feature selection for eliminating irrelevant and redundant features.

3 Experiments

We conduct two sets of experiments to demonstrate the effectiveness of the proposed approach. In the first set of experiments, we focus on the full rank metric

learning and the classification performance of the proposed method was compared with that of state-of-the-art metric learning algorithms, including LMNN¹, NCA², ITML³, LFDA⁴ and RCA⁵, on six low-dimensional UCI datasets [13]. In the second set of experiments, three different high-dimensional datasets were used to evaluate the capability of our algorithm on dimensionality reduction. For the results reported in this paper, the numbers of nearest hits and nearest misses of our algorithm are all set to $k = 3$ and the balance parameter μ is automatically tuned via cross validation.

3.1 Experiments on Low Dimensional Datasets

In this section, we present a comparative study on the proposed method with the state-of-the-art metric learning methods. The experiments are done on six UCI datasets by using KNN classifier with $k = 3$. Table 1 gives the description information about the datasets. For the larger dataset Letter, the default train/test partition specified in the UCI repository was adopted. For the other five small datasets, the classification errors are obtained by averaging over 100 runs with random 70/30 splits of each dataset. Since this experiment focuses on the full rank metric learning, the regularization parameter λ of our algorithm is set to 0. The experimental results of the proposed method and other five metric learning algorithms are reported in Table 2. The best classification result for each dataset is shown in bold. From the table, it can be seen that except for Balance, in which NCA obtains the best classification result, for the remaining five datasets, NHMCA is the clear winner compared to LMNN, NCA, ITML, LFDA and RCA according to the classification error.

Table 1. Summary of the datasets used in the experiments

Datasets	Samples	Features	Classes
Balance	625	4	3
Ionosphere	351	34	2
Libras	360	90	15
Sonar	208	60	2
Spectf	267	44	2
Letter	20000	16	26

3.2 Experiments on High Dimensional Datasets

We compare the performance of the proposed method with three algorithms NCA, LMCA and LFDA in the low rank metric learning case, where LMCA

¹ Available at <http://www.weinbergerweb.net/Downloads/LMNN.html>

² Available at <http://www.cs.berkeley.edu/~fowlkes/software/nca/>

³ Available at <http://www.cs.utexas.edu/~pjain/itml/>

⁴ Available at <http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LFDA/>

⁵ Available at <http://www.cs.huji.ac.il/~tomboy/code/RCA.zip>

Table 2. Classification errors (%) of KNN ($k = 3$) using different learned metrics on the six UCI data sets. The number in the brackets represents the standard deviation. The minimum classification error of each row is highlighted in bold.

Datasets	NHMCA	LMNN	NCA	ITML	LFDA	RCA
Balance	5.39(1.56)	17.58(1.97)	5.30 (1.59)	10.33(3.11)	8.63(2.07)	17.67(2.02)
Ionosphere	10.95 (3.02)	11.76(2.80)	12.99(3.32)	11.93(3.07)	44.02(12.95)	33.84(4.54)
Libras	17.79 (3.78)	20.24(4.25)	22.27(4.07)	25.21(5.65)	38.43(4.87)	21.36(10.76)
Sonar	15.61 (4.74)	16.31(4.25)	15.72(4.48)	19.18(6.45)	21.44(5.05)	54.30(4.36)
Spectf	22.57 (4.97)	24.48(4.32)	24.56(4.00)	27.47(4.32)	25.75(3.52)	24.39(5.86)
Letter	2.35	3.60	2.57	4.45	3.58	4.62

is a variant of LMNN and can perform dimensionality reduction and metric learning simultaneously. In the following experiments, we use the transformation computed by PCA to initialize NCA, LMCA and NHMCA. Three public high-dimensional datasets ORL, Semeion and MADELON were used for testing. The detailed description of these datasets is given in the following:

- The ORL (Olivetti Research Laboratory)⁶ face database contains 400 images of 40 individuals, each individual has 10 images. As in [24], each image was manually aligned, cropped, and then resized to 32×32 pixels, with 256 grey levels per pixel. Training and test sets were constructed by randomly sampling 5 images of each person for training and the remaining 5 images for testing. The experimental results are obtained by averaging over 10 runs of random splits.
- Semeion is a dataset of handwritten digit from UCI repository, which contains 1593 handwritten digits from around 80 persons. We use 50% of the data for training and the remaining 50% for testing according to the suggestion of UCI repository.
- MADELON is a two-class artificial dataset, which was part of the NIPS 2003 feature selection challenge. The dataset consists of 2000 training examples and 600 testing examples with 500 features and is publicly available from UCI repository. In the 500 features, only 20 features have discriminant information and the remaining 480 features have no predictive power. The default train/test split was adopted.

The curves of classification errors of KNN based on the learned metrics derived from four metric learning algorithms under different projection dimensions on three datasets are plotted in Fig. II. For the results shown in the figure, the number of nearest neighbors of KNN is set to 1 for ORL and 3 for the other two datasets. From the figure, it can be observed that NHMCA performs consistently better than LFDA, NCA and LMCA on all datasets. Specially, on the MADELON dataset containing noise features, NHMCA obtains significantly better classification results compared other three methods, which may be due to the

⁶ <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

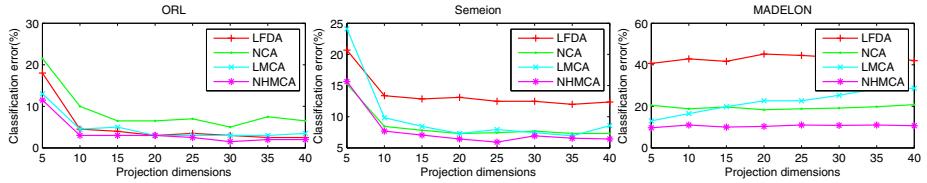


Fig. 1. Classification error vs. projection dimension on three high dimensional datasets

fact that the regularization technique is introduced to overcome overfitting. For a comparison among LFDA, NCA and LMCA, no method is conformably better than other methods across all datasets. In particular, it should be noted that the performance of LFDA is very sensitive to the used dataset. LFDA performs quite well on ORL while the worst on Semeion and MADELON. One possible explanation is that its affinity matrix used to instruct the metric learning is determined by the local scaling method, while the latter is significantly depended on the Euclidean distance in the original input space. When the initial Euclidean distance cannot reflect the importance relationships between the data very well, LFDA will fails to perform better.

Using the MADELON dataset, we further compare the computational efficiency of three nearest neighbor based metric learning algorithms, NHMCA, NCA and LMCA. For a fair comparison, three algorithms are all implemented in standard C++. The experiment is conducted on Windows XP system with Pentium(R) Dual-Core 2.5GHz CPU and 2G main memory. Fig. 2 presents the training time of three algorithms under different projection dimensions on MADELON. From Fig. 2 one can see that the computational complexity of three algorithms is approximately linear with respect to the projection dimensions, and NHMCA is much faster than NCA and LMCA.

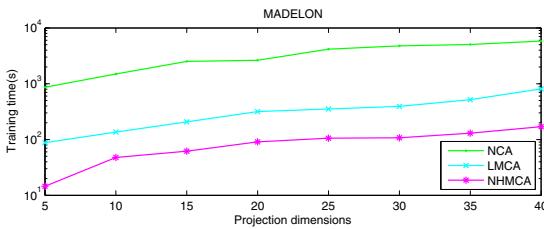


Fig. 2. Training time vs. projection dimension on MADELON dataset

4 Conclusion and Future Work

In this paper, we propose a novel nearest neighbor-based algorithm to efficiently learn a distance metric based on maximizing the margin while maintaining the compactness between each training sample and its nearest hits. Experimental results on various datasets show the superior performance of the proposed metric

learning method. In particular, the proposed method is very similar to LMNN, but performs better than the latter. In the future, we will investigate the nonlinear extension of the proposed metric learning method by introducing kernel trick. We also intend to extend the proposed method for feature selection applications.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Nos. 60872099 and 60902099.

References

1. Fisher, R.: The use of multiple measures in taxonomic problems. *Annals of Eugenics* 7, 179–188 (1936)
2. Xu, Y., Yang, J.Y., Jin, Z.: Theory analysis on fslda and ulda. *Pattern Recognition* 36, 3031–3033 (2003)
3. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18(6), 607–616 (1996)
4. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side-information. In: Xing, E., Ng, A., Jordan, M., Russell, S. (eds.) *Advances in neural information processing systems*, pp. 521–528 (2003)
5. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research* 6(1), 937–965 (2006)
6. Hoi, S., Liu, W., Lyu, M., Ma, W.: Learning distance metrics with contextual constraints for image retrieval. In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2 (2006)
7. Sugiyama, M.: Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. *Journal of Machine Learning Research* 8, 1027–1061 (2007)
8. Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. In: *Proceedings of the 24th International Conference on Machine learning*, ACM, New York (2007)
9. Yang, L., Jin, R.: Distance metric learning: A comprehensive survey. Michigan State University (2006)
10. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. In: *Advances in Neural Information Processing Systems*, vol. 17, pp. 513–520. MIT Press, Cambridge (2005)
11. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: *Advances in Neural Information Processing Systems*, vol. 18, pp. 1473–1480. MIT Press, Cambridge (2006)
12. Torresani, L., Lee, K.: Large margin component analysis. *Advances in neural information processing systems* 19 (2007)
13. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
14. Cai, D., He, X., Han, J., Zhang, H.: Orthogonal laplacianfaces for face recognition. *IEEE Trans. Image Processing* 15(11), 3608–3614 (2006)

Backward-Forward Least Angle Shrinkage for Sparse Quadratic Optimization

Tianyi Zhou and Dacheng Tao

Centre for Quantum computation & Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney
dacheng.tao@uts.edu.au

Abstract. In compressed sensing and statistical society, dozens of algorithms have been developed to solve ℓ_1 penalized least square regression, but *constrained sparse quadratic optimization* (SQO) is still an open problem. In this paper, we propose *backward-forward least angle shrinkage* (BF-LAS), which provides a scheme to solve general SQO including sparse eigenvalue minimization. BF-LAS starts from the dense solution, iteratively shrinks unimportant variables' magnitudes to zeros in the backward step for minimizing the ℓ_1 norm, decreases important variables' gradients in the forward step for optimizing the objective, and projects the solution on the feasible set defined by the constraints. The importance of a variable is measured by its correlation w.r.t the objective and is updated via least angle shrinkage (LAS). We show promising performance of BF-LAS on sparse dimension reduction.

1 Introduction

Various problems can be exactly/approximately solved by optimizing an quadratic function on given constraints [2][5][6]. Comparing against *quadratic optimization* (QO), sparse QO (SQO) brings computational efficiency, robustness and explicit interpretation to the solution. Thus it is recently preferred in machine learning and statistics, e.g., dimension reduction[10][8], classification and model selection. SQO imposes an ℓ_1 penalty on a quadratic function:

$$\text{SQO} : \min_{x \in Q} x^T Px + v^T x + \lambda \|x\|_1, \quad (1)$$

where Q is a convex set that defined by given constraints. The ℓ_1 penalty is a convex relaxation of the cardinality penalty, which encourages a sparse solution.

Lasso [2] tackles the ℓ_1 penalized least square regression and has been widely applied to feature selection and sparse signal recovery:

$$\text{lasso} : \min_x \|Ax - b\|_2^2 + \lambda \|x\|_1. \quad (2)$$

Forward greedy methods, e.g., orthogonal matching pursuit (OMP), are convectional solutions to (2) by sequentially adding one or more mostly important variables to the solution. The importance of a variable is usually measured by its correlation w.r.t the

given response. However, forward greedy algorithms cannot always reach the global optimum. *Least angle regression* (LARS) [3] is a path-following algorithm which obtains the global optimum of (2). LARS starts from $x = \mathbf{0}$, sequentially adds one variable to the support set of x , and updates the solution x to one point satisfying KKT conditions of (2) with a decreasing λ . The solution of each iteration round is a “warm start” for the subsequent iteration round. Another flaw of forward greedy algorithms is that they can never correct mistakes made in earlier steps. FoBa [9] has been developed to correct these mistakes by adding a backward reduction of variables after forward selections.

However, existing algorithms for sparse least square regression cannot be directly applied to the SQO with additional constraints. In this paper, we propose *backward-forward least angle shrinkage* (BF-LAS) to solve general SQO problems, one representative of which is the *sparse eigenvalue minimization* (SEM):

$$\text{SEM: } \min_{x: \|x\|_2=1} x^T Px + \lambda \|x\|_1. \quad (3)$$

The eigenvalue maximization problem can be transformed to (3) by letting $P := -P$. SEM has a lot of applications in sparse principal component analysis and sparse coding. In this paper, we will detail how to solve general SQO by BF-LAS, some parts will use SEM as an example. In particular, BF-LAS starts from the dense solution (the last eigenvector for SEM) and then seeks a sparse solution which produces the smallest shift from the optimum objective (the minimum eigenvalue for SEM) in the feasible set. The variables are separated into important ones and unimportant ones according to their correlations w.r.t the objective. BF-LAS iteratively removes the unimportant variables by shrinking their magnitudes, selects the important variables by decreasing their gradients, and projects the solution onto the feasible set. The removal of unimportant variables is called “backward step (removal)” and contributes to the decreasing of the ℓ_1 norm, while the selection of important variables is called “forward step (selection)” and contributes to the optimization of the objective. The backward step and forward step are accomplished by least angle shrinkage (LAS), which ensures the optimality of the solution and constructs the whole solution path.

BF-LAS applies backward-forward greedy search method to the path-following scheme and thus provides an efficient and effective solution to general SQO. First, BF-LAS improves the robustness of the solution in small λ situation, because the variables are removed or selected via path-following strategy rather than purely greedy search. Second, BF-LAS can tackle the SQO problems with constraints, because when the projection to the feasible set is applied, the iterative alternation of backward and forward steps guarantee the minimization of both the ℓ_1 penalty and the objective.

2 Preliminary

In this paper, we use lower-case letters, *e.g.*, x , to represent vectors and x_i is the i^{th} entry of x . We use upper-case letters, *e.g.*, X , to represent matrices, X_{ij} is the entry of the i^{th} row and the j^{th} column of X , $X_{\cdot i}$ is the i^{th} row of X and $X_{\cdot j}$ is the j^{th} column of X . Given an index set S , $(x_S)_i = x_{S_i}$, $(X_S)_{ij} = X_{S_i S_j}$. Given two index sets A and B , $(X_{AB})_{ij} = X_{A_i B_j}$. Let \overline{A} be the complement of set A . We use $\mathcal{P}_Q(x)$ to mark

the projection of x on the convex set Q . The solution and correlation vector at the k^{th} iteration step are $x(k)$ and $c(k)$, respectively.

Definition 1. (Correlation). *The correlation of the i^{th} variable in x is marked as c_i and is the gradient of the objective function w.r.t x_i . The correlation vector of x in (1) is:*

$$c = \frac{\partial f(x)}{\partial x} = 2Px + v. \quad (4)$$

So the correlation vector c for SEM (3) is $c = 2Px$. c describes the variables' correlations to the objective $f(x) = x^T Px + v^T x$. An absolute correlation is the absolute value of the correlation. Variables with large absolute correlations are **important** to the objective optimization, while the ones with small absolute correlations are **unimportant**. Therefore, BF-LAS shrinks the unimportant variables' magnitudes for ℓ_1 norm minimization, because their changes bring slight shift of the objective. BF-LAS shrinks the important variables' correlations, because their changes bring large decreasing of the objective.

Definition 2. (Active Set). *In forward/backward step of BF-LAS, important/unimportant variables are selected sequentially for shrinkage of correlation/magnitude. The active set \mathcal{A} is the set of the selected variables.*

Definition 3. (Variable Group). *In backward steps of BF-LAS, unimportant variables are further divided into two subgroups: the variables which magnitudes are shrunk with the increasing of correlations, i.e., $G1$, and the variables which magnitudes and correlations are simultaneously shrunk, i.e., $G2$. To define $G1$ and $G2$, the second order derivative of the objective $f(x) = x^T Px + v^T x$ is considered*

$$\frac{\partial^2 f(x)}{\partial x_i^2} = 2P_{ii} \text{ and } \text{sign}\left(\frac{c_i(k) - c_i(k-1)}{x_i(k) - x_i(k-1)}\right) = \text{sign}(P_{ii}). \quad (5)$$

For any $i \in G1$, we have:

$$\frac{|c_i(k)| - |c_i(k-1)|}{|x_i(k)| - |x_i(k-1)|} = \frac{\text{sign}(c_i)}{\text{sign}(x_i)} \cdot \text{sign}\left(\frac{c_i(k) - c_i(k-1)}{x_i(k) - x_i(k-1)}\right) = -1. \quad (6)$$

The $G2$ has a similar result by replacing -1 with 1 on the right hand side of (6). Both $G1$ and $G2$ can then be identified by their signs of coefficients and correlations:

$$G1 = \{i | \text{sign}(c_i) \cdot \text{sign}(x_i) \cdot \text{sign}(P_{ii}) = -1\}, \quad (7)$$

$$G2 = \{i | \text{sign}(c_i) \cdot \text{sign}(x_i) \cdot \text{sign}(P_{ii}) = 1\}. \quad (8)$$

The $G3$ is the set of important variables.

Lemma 1. *An $x = \mathcal{P}_Q(y)$ obtained by BF-LAS is the solution of SQO (1) iff:*

$$|c_{i:y_i \neq 0}| = \lambda, \quad (9)$$

$$|c_{i:y_i=0}| \leq \lambda. \quad (10)$$

If $g(x) = f(x) + \lambda\|x\|_1$ is the objective function of SQO (1), we have $\partial g(y) = 0$, wherein ∂ represents the subgradient. Thus y is the solution of the unconstrained minimization with objective $g(x)$. According to the projection theorem [1], $x = \mathcal{P}_Q(y)$ is the solution of (1). In BF-LAS, absolute correlations of variables in the active set are kept equal to each other before projection to satisfy Lemma 1.

3 Backward-Forward Least Angle Shrinkage

BF-LAS starts from the dense solution and iteratively executes backward and forward steps via least angle shrinkage. In backward steps, unimportant variables in $G1$ are sequentially added to the active set and keep their absolute correlations equally increasing (with decreasing magnitudes and increasing objective). Afterward, unimportant variables in $G2$ are sequentially added to the active and keeps their absolute correlations equally decreasing (with decreasing magnitudes and objective). In forward steps, variables in $G3$ are sequentially added to the active set and keep their absolute correlations equally decreasing (with decreasing objective). The solution is projected onto the feasible set in the end of each backward or forward step.

We first introduce the least angle shrinkage (use SEM as an example), and then present the backward-forward iterations. The algorithm of BF-LAS is shown in Algorithm 2.

3.1 Least Angle Shrinkage

Least angle shrinkage (LAS) is a slight modification of LARS [3] and is used in each backward/forward step of BF-LAS to build the solution path that satisfies Lemma 1. It initially obtains the direction of $x_{\mathcal{A}}$ along which the correlations $|c_{\mathcal{A}}|$ are equal to each other. Then it proceeds along the direction until either of the following two events happens: 1) another variable x_i outside \mathcal{A} reach the same correlation of \mathcal{A} , i.e., $|c_i| = |c_{j:j \in \mathcal{A}}|$; and 2) a variable x_i is shrunk to zero. When 1) happens, x_i will be added to \mathcal{A} ; when 2) happens, x_i will be removed from \mathcal{A} . The above procedure are iterated until convergence. We detail it in the following two stages by using SEM as an example.

Stage 1. The direction d is calculated from $P_{\mathcal{A}}$ and $c_{\mathcal{A}}$ according to Lemma 1. Let $\Delta c = c(k) - c(k-1)$ and $\Delta x = x(k) - x(k-1)$, the change of correlations of active variables is:

$$\Delta c_{\mathcal{A}} = 2P_{\mathcal{A}}\Delta x_{\mathcal{A}}.$$

In order to keep the correlations in $|\Delta c_{\mathcal{A}}|$ equal to each other, i.e., $|\Delta c_{\mathcal{A}}| = \lambda \cdot \mathbf{1}$, $\Delta x_{\mathcal{A}}$ should satisfy:

$$\Delta x_{\mathcal{A}} = (\lambda/2) (P_{\mathcal{A}}^{-1} \text{sign}(\Delta c_{\mathcal{A}})),$$

where $\text{sign}(\Delta c_{\mathcal{A}})$ is determined by the definitions of variable groups.

- Since the variables in $G1$ have decreasing magnitudes, $\text{sign}(\Delta x_{\mathcal{A}}) = -\text{sign}(x_{\mathcal{A}})$. According to (6), we have $\text{sign}(\Delta c_{\mathcal{A}}) = \text{sign}(c_{\mathcal{A}})$ for $G1$.
- Similar to the above derivations, we have $\text{sign}(\Delta c_{\mathcal{A}}) = -\text{sign}(c_{\mathcal{A}})$ for $G2$.
- Since the variables in $G3$ have decreasing absolute correlations, we have $\text{sign}(\Delta c_{\mathcal{A}}) = -\text{sign}(c_{\mathcal{A}})$ for $G3$.

The direction d is obtained by eliminating the scale $\lambda/2$ from $\Delta x_{\mathcal{A}}$, i.e.,

$$d_{\mathcal{A}} = (P_{\mathcal{A}}^{-1} \text{sign}(\Delta c_{\mathcal{A}})) \quad \text{and} \quad d_{\overline{\mathcal{A}}} = 0. \quad (11)$$

Algorithm 1. Least Angle Shrinkage (LAS)

```

Initialize:  $\mathcal{A}$ ,  $x$ ,  $k = 1$ .
repeat
  Step 1: Compute correlation vector  $c$  using (4)
  Step 2: Compute direction  $d$  using (11)
  Step 3: Compute step size  $\alpha$  using (14)
  Step 4: Update  $\mathcal{A}$  and  $x$  using (15) and (16),  $k := k + 1$ 
until  $|f(x(k)) - f(x(k-1))| \leq \epsilon$ 
Project  $x$  to the feasible set, i.e.,  $x = \mathcal{P}_Q(x(k))$ 
return  $x$ 

```

Stage 2. The step size α is calculated from the direction d and the correlation c . In BF-LAS, the solution proceeds along d until either event 1) or 2) is satisfied. Thus we compute all the possible step sizes that satisfy event 1) and event 2), and choose the smallest one as α . We use \hat{c} to represent the absolute correlation of variables in \mathcal{A} , i.e., $\hat{c}_{\mathcal{A}} = |c_{i:i \in \mathcal{A}}|$, and let $\hat{s} = \text{sign}(\Delta c_{i:i \in \mathcal{A}}) \cdot \text{sign}(c_{i:i \in \mathcal{A}})$.

For event 1), let the absolute correlation of a variable in $\overline{\mathcal{A}}$ reach the absolute correlation of variables in \mathcal{A} :

$$|c_{j:j \in \overline{\mathcal{A}}} + \Delta c_{j:j \in \overline{\mathcal{A}}}| = \hat{c} + \alpha \hat{s}, \quad \Delta c_{\overline{\mathcal{A}}} = P_{\overline{\mathcal{A}}\mathcal{A}} \Delta x_{\mathcal{A}} = \alpha (P_{\overline{\mathcal{A}}\mathcal{A}} d_{\mathcal{A}}). \quad (12)$$

For event 2), we have

$$x_{i:i \in \mathcal{A}} + \alpha d_{i:i \in \mathcal{A}} = 0. \quad (13)$$

Let $dc = 2P_{\mathcal{A}}d$ be the Δc when $\Delta x = d$. The α can be selected from α_1 that satisfies (12) and α_2 that satisfies (13):

$$\alpha_1 = \min_{j:j \in \overline{\mathcal{A}}}^+ \left\{ \frac{|\hat{c}| \mp c_j}{\pm dc_j - \hat{s}} \right\}, \quad \alpha_2 = \min_{i:i \in \mathcal{A}}^+ \{-x_i/d_i\}, \quad \alpha = \min \{\alpha_1, \alpha_2\}. \quad (14)$$

In the following lines, let \hat{j} and \hat{i} be the indices corresponding to α_1 and α_2 , respectively.

The active set \mathcal{A} and the solution x are updated according to the direction d and the step size α :

$$\text{if } \alpha = \alpha_1, \mathcal{A} = \{\hat{j}\} \cup \mathcal{A}; \text{ else } \mathcal{A} = \mathcal{A} - \{\hat{i}\}, \quad (15)$$

$$x = x + \alpha d. \quad (16)$$

At the end of LAS, the solution x is projected onto the feasible set Q defined by constraints. For SEM problem (3), the projection equals to the normalization of x , i.e., $x := x/\|x\|_2$. The LAS algorithm is shown in Algorithm 1.

3.2 Backward-Forward Iteration

BF-LAS starts from the dense solution $x(0)$, and iterates backward and forward steps via LAS several times until a q -sparse solution is obtained, i.e.,

$$\text{card}(x) \leq q. \quad (17)$$

Algorithm 2. Backward-forward least angle shrinkage (BF-LAS)

```

Input:  $P, v, x(0), q$  and  $\epsilon$ .
Initialize:  $t = 1, x = x(0)$ .
repeat
    //Step 1: Backward removal of  $G1$ .
    Initialize  $\mathcal{A}$  using (18) and run LAS in Algorithm 1.
    //Step 2: Backward removal of  $G2$ .
    Initialize  $\mathcal{A}$  using (19) and run LAS in Algorithm 1.
    //Step 3: Forward selection of  $G3$ .
    Initialize  $\mathcal{A}$  using (20) and run LAS in Algorithm 1.
     $t = t + 1$ .
until  $\text{card}(x) \leq q$ 
return  $x$ .

```

In each iteration round, BF-LAS separates the variables into $G1$, $G2$ and $G3$ according to Definition 3, and sequentially executes backward removal of $G1$, backward removal of $G2$ and forward selection of $G3$.

In the backward removal of $G1$, the active set \mathcal{A} in LAS is initiated as the variable in $G1$ with the smallest absolute correlation:

$$\mathcal{A} = \left\{ \arg \min_{i:i \in G1} \{|c_i|\} \right\}. \quad (18)$$

Then LAS is applied to obtain a sparse x . The absolute correlations of variables in the active set \mathcal{A} are kept equal to each other and gradually increasing, while their magnitudes are shrunk. This is a trade-off between the sparsity of the solution and the optimization of the objective. BF-LAS shrinks the magnitudes of the variables with the smallest correlations and thus minimize the shift of the objective.

After the backward removal of $G1$, BF-LAS switches to the backward removal of $G2$. The initialization of \mathcal{A} is similar to (18), i.e.,

$$\mathcal{A} = \left\{ \arg \min_{i:i \in G2} \{|c_i|\} \right\}. \quad (19)$$

Then LAS is applied to obtain a sparse x . The absolute correlations of variables in \mathcal{A} are kept decreasing with shrinking of their magnitudes. Hence the objective and the ℓ_1 penalty are simultaneously decreased.

Then BF-LAS switches to the forward selection of $G3$. We first initialize the active set \mathcal{A} as:

$$\mathcal{A} = \left\{ \arg \max_i \{|c_i|\} \right\}. \quad (20)$$

Then LAS is applied to obtain a sparse x . In the forward step, important variables with large absolute correlations are sequentially added to \mathcal{A} , their absolute correlations are kept equal and decreasing. Hence the objective is decreased.

The backward and forward steps are iterated for several times to arrive at a preferred sparse solution. BF-LAS is shown in Algorithm 2.

For SEM (3), The main time cost of running BF-LAS is the matrix inverse computation $P_{\mathcal{A}}^{-1}$ in (11) with complexity $\mathcal{O}(\bar{p}^3)$ if $P_{\mathcal{A}} \in \mathbb{R}^{\bar{p} \times \bar{p}}$. However, BF-LAS adds to

or removes from \mathcal{A} only one variable at each step, and thus there are only one column and one row are added to or removed from $P_{\mathcal{A}}$. Therefore, we can accelerate this matrix inverse computation by using block matrix inverse formula. The time complexity of BF-LAS iteration is thus reduced to $\mathcal{O}(\bar{p}^2 + 5\bar{p})$. Hence BF-LAS is much faster than the SDP relaxation for (3) including Sparse PCA, whose complexity is about $\mathcal{O}(p^3)$ by using fast SDP algorithms.

4 Experiments

In this section, we apply BF-LAS to sparse dimension which is an SEM problem. BF-LAS is compared against existing algorithm.

4.1 Sparse Dimension Reduction

Dimension reduction plays an important role in statistical data analysis, and eigenvalue decomposition can solve many dimension reduction algorithms, e.g., principal component analysis and discriminative locality alignment (DLA) [8]. DLA achieves top level performance, because it preserves the local geometry of intraclass samples, maximizes the margin defined by interclass samples, and has no small sample size problem.

Taking DLA as an example, we apply BF-LAS to obtain its sparse projection matrix for face recognition. We compare the sparse project matrix against the dense projection matrix obtained by eigenvalue decomposition on the FERET dataset [4]. The original dataset consists of 13,539 face images from 1,565 individuals. The images vary in size, gender, pose, illumination, facial expression and age. In this paper, we randomly select 100 individuals, each of which has 7 images, 5 for training and the other 2 for test.

Figure 1 shows the first 10 eigenvectors of DLA and BF-LAS based sparse DLA. BF-LAS selects 480 features from the original 1600 ones (every face is a 40×40 image). Important areas, e.g., eyes, eyebrows, cheekbone, nose, mouth and face contour, are correctly selected and can be well interpreted as the important features to distinct different persons. Figure 2 presents the recognition rate versus the feature dimension (that changes from 1 to 30) obtained by using BF-LAS and eigenvalue decomposition. BF-LAS is superior to DLA even when only 30% features are selected. This is because BF-LAS is able to select critical features and eliminate noisy ones for subsequent classification.



Fig. 1. First 10 eigenvectors of dimension reduction on human face dataset. Top row: sparse eigenvectors (480 features) of DLA obtained by BF-LAS. Bottom row: dense eigenvectors (1600 features) of DLA obtained by eigenvalue decomposition.

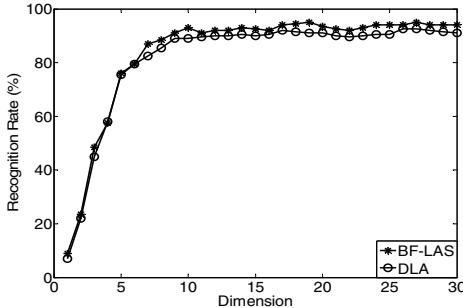


Fig. 2. Recognition rate vs. Dimension. BF-LAS based sparse DLA (only 30% features are selected) achieves better performance than dense DLA.

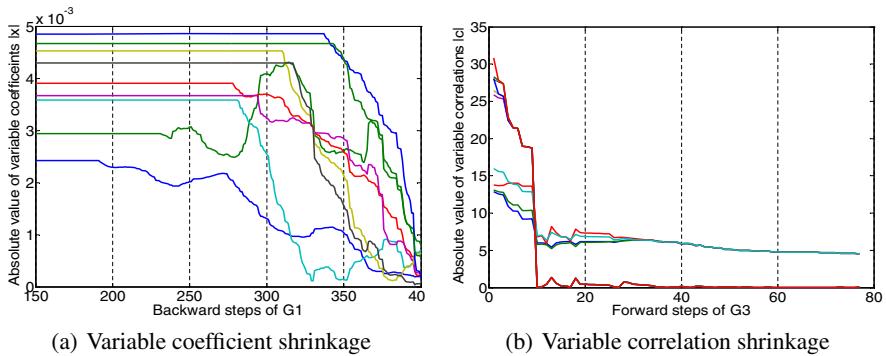


Fig. 3. Solution paths of G_1 (a) and correlation paths of G_3 (b) in backward steps in BFLAS

Figure 3(a) shows the change of absolute coefficients of 10 G_1 variables in backward steps of G_1 . The solution paths indicate that unimportant variables are selected sequentially and shrunk to zeros rapidly. This is consistent with our analyses. Figure 3(b) shows the change of absolute correlations of 10 G_3 variables in forward steps of G_3 . In BF-LAS, the absolute correlations of important variables should be shrunk to an identical value in forward steps. In Figure 3(b), however, variables are separated into two groups and their absolute correlations are shrunk to two different values. This is because variable coefficients of the group with larger absolute correlations are nonzero and thus their correlations should satisfy (9) in Lemma 1, while the coefficients of variables in the other group are shrunk to zeros and they have lower absolute correlations according to (10).

5 Conclusion

In this paper, we propose *backward-forward least angle shrinkage* (BF-LAS) to solve sparse quadratic optimization (SQO) problems with different constraints. BF-LAS is a backward-forward path-following algorithm. It starts from the dense solution, shrinks

unimportant variables towards zeros along the backward path, and decreases the gradients of the important variables along the forward path. Backward-forward steps are iterated by using least angle shrinkage. It induces sparsity to the solution by using variable magnitude shrinkage, and simultaneously minimizes the objective by using variable correlation shrinkage. Comparing with the existing ℓ_1 penalized least square regression algorithms, the proposed BF-LAS can handle general SQO problems including both sparse eigenvalue minimization and *lasso*. Experimental results of sparse dimension reduction and model selection suggest the effectiveness of BF-LAS.

Acknowledgement

This work was supported by the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1006), Zhejiang University.

References

- [1] Bertsekas, D.P.: Convex Analysis and Optimization. Athena Scientific (2003)
- [2] Bian, W., Tao, D.: Manifold regularization for sir with rate root-n convergence. In: NIPS, pp. 1–8 (2009)
- [3] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Annals of Statistics 32, 407–499 (2002)
- [4] Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The feret evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 22(10), 1090–1104 (2000)
- [5] Si, S., Tao, D.: Bregman divergence based regularization for transfer subspace learning. IEEE Transactions on Knowledge and Data Engineering 22(7), 929–942 (2010)
- [6] Tao, D., Li, X., Wu, X., Maybank, S.J.: Geometric mean for subspace selection. IEEE Transactions on Pattern Analysis and Machine Intelligences 31(2), 260–274 (2009)
- [7] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B 58, 267–288 (1994)
- [8] Zhang, T., Tao, D., Yang, J.: Discriminative locality alignment. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 725–738. Springer, Heidelberg (2008)
- [9] Zhang, T.: Adaptive forward-backward greedy algorithm for sparse learning with linear models. In: NIPS, pp. 1921–1928 (2008)
- [10] Zhou, T., Tao, D., Wu, X.: Manifold elastic net: a unified framework for sparse dimension reduction. In: Data Mining and Knowledge Discovery, pp. 1–32. Springer, Heidelberg (2010), doi:10.1007/s10618-010-0182-x

An Enhanced Semi-supervised Recommendation Model Based on Green's Function

Dingyan Wang and Irwin King

Department of Computer Science and Engineering,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
`{dywang,king}@cse.cuhk.edu.hk`

Abstract. Recommendation, in the field of machine learning, is known as a technique of identifying user preferences to new items with ratings from recommender systems. Recently, one novel recommendation model using Green's function treats recommendation as the process of label propagation. Although this model outperforms many standard recommendation methods, it suffers from information loss during graph construction because of data sparsity. In this paper, aiming at solving this problem and improving prediction accuracy, we propose an enhanced semi-supervised Green's function recommendation model. The main contributions are two-fold: 1) To reduce information loss, we propose a novel graph construction method with global and local consistent similarity; 2) We enhance the recommendation algorithm with the multi-class semi-supervised learning framework. Finally, experimental results on real world data demonstrate the effectiveness of our model.

Keywords: Green's function, semi-supervised learning, recommender system, item graph.

1 Introduction

In modern days, people are usually overwhelmed with various of choices on the web and waste much time searching the wanted one. Recommendation, as the technology to suggest personalized items to meet special needs and tastes of different persons [1,6], has been widely applied into many e-commercial and entertainment web sites, like Amazon and IMDb. The feedback shows that recommender systems not only improve consumer satisfaction but also increase the profit of e-commercial systems. On the other hand, recommendation with rating information from recommender systems, as an application of machine learning, has been studied widely in academic. Normally, recommendation can be regarded as a prediction task : given a partially observed user-item rating matrix $R_0 \in \mathbb{R}^{M \times N}$, whose rows represent M users, columns represent N items, non-zero elements represent observed ratings and zero elements represent those unknown ratings, the goal is to predict unknown ratings to complete the matrix, with each element r_{jk} ($1 \leq j \leq M, 1 \leq k \leq N$) in the range of rating $1, \dots, R$ ($R > 1, R \in \mathbb{Z}$). Various recommendation methods have been proposed, mainly

divided into two categories: memory-based and model-based methods [10]. In our paper, we focus on the memory-based methods, which assume that similar users rate the same item similarly or similar items get similar ratings from the same users. Standard memory-based methods can be further divided into user-based [5] [12] and item-based methods [2] [11].

Recently, one model applying Green's function learning framework [3] is proposed as an item-based recommendation method. Green's function for the Laplace operator represents the propagation of influence of point sources. This model takes a novel view by treating recommendation as the process of label information propagation from labeled data (i.e., items with ratings) to unlabeled data (i.e., items without ratings). Although this model is based on memory-based assumption, it utilizes the item graph and label propagation to make prediction. Moreover, it obtains a higher prediction accuracy compared to standard memory-based methods while its algorithm is more compact and simpler.

However, there are still some limitations with the Green's function recommendation model. One essential issue of the model is that the Green's function is sensitive to the item graph. The previous model constructs the item graph with the simple *cosine similarity* between items. Due to the sparse data in real recommender systems, one item with few ratings and the other with many ratings may be considered to be with low similarity using *cosine similarity*. In addition, *cosine similarity* computes the similarity in a local view since each similarity is inferred from the concurrent ratings of only two items. As a result, the item graph by *cosine similarity* suffers from information loss so that it can degrade the performance of Green's function method. Likewise, some other standard similarity computation methods, like *Pearson Correlation Coefficient* (PCC) [7] and *conditional probability* [11], also suffer the same local similarity problem. Another limitation is that the previous algorithm needs normalization and post-processing to get the prediction value and it is more preferable to the two-rating case.

In this paper, aiming at resolving the local similarity problem in item graph construction, we first propose an enhanced item Graph construction method. More specially, we derive latent features from ratings with one popular latent feature model PMF (probabilistic matrix factorization), and impose the consistency between the global similarity from latent features and the local similarity from standard similarity methods. At the same time, considering the process of Green's function recommendation similar to semi-supervised learning, we extend the previous algorithm with multi-classes semi-supervised learning, making it more suitable for recommendation with a larger rating range. Finally, we conduct a series of experiments on the famous real world dataset (i.e., MovieLens) to evaluate the performance of our model. Comparing to previous Green's function model and other memory-based methods, the experimental results demonstrate the effectiveness of our model.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of Green's function recommendation framework and item graph construction. Section 3 illustrates our novel recommendation model. The results

of empirical analysis are presented in Section 4, followed by the conclusion in Section 5.

2 Related Works

2.1 Green's Function Recommendation Model

Originally Green's function is involved with the Laplace operator in many physics problems, like the diffusion process. The physical explanation is that Green's function represents the propagation of influence of point sources. Considering the similarity between label propagation and the diffusion process, Green's function is applied into machine learning with label propagation. As a graph-based learning model, Green's function is applied into recommendation [3] to predict unknown ratings for items. This recommendation model is memory-based while it is different from other standard memory-based methods which only use simple averaging ratings of similar items or users.

In this graph-based recommendation model, an item graph must be constructed first. An item graph is an undirected graph $\mathcal{G} = (\nu, \varepsilon)$ with a weight $w \in W$ in each edge $e \in \varepsilon$ and each node $v \in \nu$ as an item, where $w_{jk} = w_{kj}$, $0 \leq w_{jk} < 1$ when $j \neq k$ and $w_{jk} = 1$ when $j = k$. An item graph is constructed by calculating similarity between items. This model to construct an item graph is to utilize standard similarity methods, such as *cosine similarity* and PCC.

Cosine similarity is simple to calculate but it is in favor of frequently rated items and ignore the different rating styles of different users. Pearson Correlation Coefficient (PCC) is proposed to consider different rating styles of users. PCC between two items is based on the common users rating the two items. However, PCC can overestimate similarities between items which happen to be rated by a few users identically in the sparse dataset [8]. Therefore, they both suffer from information loss in large sparse datasets from real recommender systems. What's more, both *cosine similarity* and PCC only take a local view of similarity, which is based on the common items or users. All these can reduce the prediction accuracy for recommendation.

When the item graph is constructed, Green's function is calculated in this way: given the item graph with edge weights W , the combinatorial Laplacian is defined as $L = D - W$, where D is the diagonal matrix with sums of each row of W , and then Green's function is defined as $G = L_{(+)}^{-1} = \frac{1}{(D-W)_+} = \sum_{i=2}^n \frac{v_i v_i^T}{\lambda_i}$ without the zero-value eigenvalue, where $Lv_i = \lambda_i v_i$, $v_p^T v_q = \delta_{pq}$, and $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ are the eigenvalues of the corresponding eigenvectors v_1, v_2, \dots, v_n . With Green's function from the item graph, recommendation is viewed as a linear influence propagation. The algorithm is defined as $\hat{R}^T = GR_0^T$, where R_0, \hat{R}^T are the original and predicted matrix respectively. The algorithm is direct and simple while the results from this formula need much postprocess to get the integer ratings. As a matter of fact, the label propagation process is similar to semi-supervised learning which makes label prediction of unlabeled data only with labeled data. This scenario of semi-supervised learning with only a small number of labeled data is in favor of data sparsity in recommender systems.

2.2 Latent Feature Models

Latent features are usually inferred from the whole rating information and hence they are in a global view to describe items or users. In spite being implicit, some latent features can best describe the characteristics of items or users. Among latent feature models, probabilistic matrix factorization (PMF) [9][4] is used widely recently. PMF assumes the probabilistic linear model into the observations and it works well on a large and sparse dataset. The objective of PMF is to maximize the log likelihood of the posterior distribution. The gradient algorithm is applied to calculate the results.

3 Our Enhanced Model

In this section, we detail our proposed enhanced Green's function recommendation model based on: 1) a novel item graph construction with global and local consistent similarity (GLCS); 2) extended multi-classes semi-supervised recommendation algorithm.

3.1 Item Graph Based on Global and Local Consistent Similarity (GLCS)

The item graph in previous model suffers from information loss, with *cosine similarity* only in the view of local similarity. To construct a more precise item graph, we should impose the consistency between local and global view of similarity. The latent features are able to describe items in a global view since they are referred from the whole rating information, which is different from *cosine similarity* only based on ratings of two items. We adopt the popular PMF to calculate the item latent features from the user-item rating matrix R_0 .

The idea of Global and Local Consistent Similarity (GLCS) is that the similarity between two items consists of two parts: one is the classical similarity and the other is similarity between two item latent feature vectors derived by PMF. The representation of GLCS is given in the following Eq.(I):

$$GLCS(j, k) = \mu sim(\mathbf{v}_j, \mathbf{v}_k) + (1 - \mu) sim(j, k), \quad (1)$$

where μ is a parameter to control the weight of consistency between the global and local view, $sim(j, k)$ is *cosine similarity* or PCC as the local view, and $sim(\mathbf{v}_j, \mathbf{v}_k)$ is the *cosine similarity* of two item latent feature vectors $\mathbf{v}_j, \mathbf{v}_k$ as the global view.

Let r_{ij} be the rating of user i for item j , and each rating r_{ij} values from 0 to 1 using the mapping function $f(x) = (x - 1)/(R - 1)$. $U \in \mathbb{R}^{k \times M}$ and $V \in \mathbb{R}^{k \times N}$ represent the latent user-specific and item-specific feature matrices, with column vectors \mathbf{u}_i and \mathbf{v}_j denoting user and item feature vectors respectively. We define the conditional distribution over the observed ratings as $p(R_0|U, V, \sigma_R^2) = \prod_{i=1}^M \prod_{j=1}^N \mathcal{N}[(r_{ij}|g(\mathbf{u}_i^T \mathbf{v}_j), \sigma^2)]^{I_{ij}}$, where I_{ij} is the indicator function equal to 1 if user i rated item j . Besides, we also place zero-mean spherical Gaussian priors

on user and item feature vectors, with $p(U|\sigma_U^2) = \prod_{i=1}^M \mathcal{N}(\mathbf{u}_i|0, \sigma_U^2 \mathbf{I})$, $p(V|\sigma_V^2) = \prod_{j=1}^N \mathcal{N}(\mathbf{v}_j|0, \sigma_V^2 \mathbf{I})$. The objective of PMF is to minimize the function

$$\mathcal{L}(R_0, U, V) = \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N I_{ij} (r_{ij} - g(\mathbf{u}_i^T \mathbf{v}_j))^2 + \frac{\lambda_U}{2} \sum_{i=1}^M \|\mathbf{u}_i\|_{Fro}^2 + \frac{\lambda_V}{2} \sum_{j=1}^N \|\mathbf{v}_j\|_{Fro}^2, \quad (2)$$

where $\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_V = \sigma^2/\sigma_V^2$ and $\|\cdot\|_{Fro}^2$ denotes the Frobenius norm. A local minimum of the objective function can be obtained with gradient descent algorithm in \mathbf{u}_i and \mathbf{v}_j .

3.2 Semi-supervised Recommendation Algorithm Based on GLCS

In our paper, we apply the algorithm of the multi-class semi-supervised learning with Green's function to recommendation. This model treats each rating in the rating sets $\{1, \dots, \mathbf{R}\}$ as a label and there are R ratings corresponding to \mathbf{R} labels. When we construct the item graph based on GLCS, we calculate its Green's function $G = \sum_{i=2}^n \frac{v_i v_i^T}{\lambda_i}$. With G , we apply the multi-classes semi-supervised learning model to our recommendation model. The algorithm is presented as:

$$y_{jk} = \begin{cases} 1, & k = \arg \max_k \sum_{i=1}^l G_{ji} y_{ik}, \quad 1 \leq j \leq n. \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Here we set the background to be that the ratings are discrete values. When we get the resulting R^T , we can do some data process to make sure that the elements are integers in the range $\{1, \dots, \mathbf{R}\}$. One frequently used method is to set a threshold for each rating value.

4 Experimental Analysis

Our experimental analysis is expected to address the following questions: 1) What is the performance of our Green's function recommendation with GLCS comparing with previous Green's function recommendation algorithms with *coseine similarity* or PCC? 2) How does our approach comparing to traditional memory-based recommendation methods? 3) How does the parameter μ in calculating GLCS affect the performance of our approach?

4.1 Dataset and Metrics

We conduct our experiments over the MovieLens¹ dataset. The data were collected through the MovieLens, a famous Web-based recommender system, during the 7-month period from September 19th, 1997 through April 22nd, 1998.

¹ <http://www.cs.umn.edu/Research/GroupsLens/>

This dataset contains 100,000 integer ratings (1-5 scales) from 943 users on 1682 movies. Each user on average rated at least 20 movies with a sparsity level $1 - \frac{100000}{943 \times 1682} = 93.70\%$. The data are split 80%/20% into training data and test data, with 80,000 ratings in training dataset and 20,000 ratings in test dataset.

We use three most widely used metrics to measure the prediction quality of recommendation approaches in our experiments: *Mean Absolute Error*(MAE), *Mean Zero-one Error*(MZOE) and *Rooted Mean Squared Error*(RMSE).

4.2 Experimental Results

In our paper, we compared our model (GGLCS) with 6 representative methods from memory-based methods in order to measure the prediction accuracy. We implement all the 6 methods on the same dataset. Four of these baseline methods are item-based (ICOS, IPCC) [2][1] and user-based (UCOS, UPCC) [5] method with *cosine similarity* and PCC respectively. GCOS and GPCC are the previous model using *cosine similarity* and PCC respectively. Besides, we also conduct some experiments to measure the impact of parameter weight μ with different latent feature dimensionality k . In our experiments, the rating background is set to be discrete-valued.

- **Impact of Weight Parameter μ .** There is a weight parameter μ to balance the similarity from global view with latent features and local view from *cosine similarity* or PCC. When $\mu = 0$, the item similarity is only the classical *cosine similarity* or PCC, and when $\mu = 1$, the item similarity is only derived from latent features. In other cases that μ is between (0, 1), we obtain the item similarity combining with global and local views.

Fig. 1 shows the impacts of μ on MAE, MZOE and RMSE with the dimensionality $k = 5$. We can observe from this figure that the value of μ affects the performances of our model significantly. As shown in all the three charts in Fig 1, when μ increases, the three prediction errors decrease first. But when μ passes 0.5, the prediction error begins to decrease with further increase of μ . From Fig. 2, we can observe the similar sensitivity of weight parameter μ to performance of our model when the $k = 10$. The optimal value of weight parameter is near 0.5 to get the best prediction accuracy.

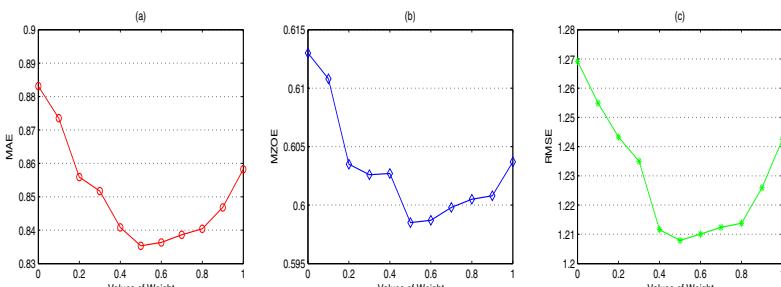


Fig. 1. Performance with Different Values of Weight Parameter μ at $k=5$

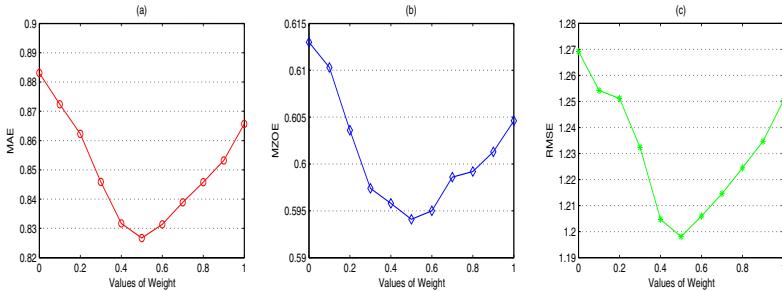


Fig. 2. Performance with Different Values of Weight Parameter μ at $k=10$

Table 1. Comparison with Different Recommendation Methods on MAE, MZOE and RMSE

Metrics \ Methods	UCOS	UPCC	ICOS	IPCC	GCOS	GPCC	GGLCS(k_5)	GGLCS(k_10)
MAE	0.962	0.943	0.890	0.889	0.883	0.861	0.835	0.827
MZOE	0.653	0.639	0.621	0.618	0.613	0.604	0.599	0.594
RMSE	1.281	1.278	1.275	1.271	1.269	1.249	1.208	1.198

This phenomenon coincides with the intuition that item graph construction with both global and local similarity can help to obtain better performances. We can conclude that when the weight μ between global and local similarities is equal and balanced, our enhanced model can have a better performance in the MovieLens dataset.

- **Performance Comparison.** In order to evaluate the efficiency and superiority of our model, we set the parameter $\mu = 0.5$ in our model which achieves the best consistency between global and local similarity and the best accuracy. Table I shows the results of performance of different recommendation methods. We observe that our model GGLCS has the lowest MAE, MZOE and RMSE among these errors in both $k = 5$ and $k = 10$. Compared to MAE of previous model with *cosine similarity*, the accuracy of our model improves about 6.4% when $k = 10$ and 5.4% when $k = 5$. Compared to ICOS, our model at $k = 10$ can improve the accuracy about 7.2%. The results demonstrate efficiency and superiority of our model over other memory-based methods. Another observation from this table is that user-based recommendation is worse than any item-based recommendation in MovieLens dataset and also much worse than any Green's function recommendation.

5 Conclusions

Previous Green's function recommendation model with *cosine similarity* suffers from information loss during the item graph construction in sparse data due to the local similarity problem. In this paper, we propose an enhanced method to construct the item graph by imposing global and local consistence on similarity

computation, where the global view is provided by latent features. What's more, we extend the Green's function recommendation algorithm with multi-classes semi-supervised learning framework. Finally, we conduct some experiments with a famous dataset to demonstrate that our approach outperforms the previous Green's function recommendation and other memory-based methods.

Acknowledgment

The work described in this paper is supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No.: CUHK 4128/08E).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17(6), 734–749 (2005)
2. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 22(1), 177 (2004)
3. Ding, C., Simon, H., Jin, R., Li, T.: A learning framework using Green's function and kernel regularization with application to recommender system. In: *Proceedings of the 13th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, p. 269. ACM, New York (2007)
4. Dueck, D., Frey, B.: Probabilistic sparse matrix factorization. University of Toronto technical report PSI-2004-23 (2004)
5. Jin, R., Chai, J., Si, L.: An automatic weighting scheme for collaborative filtering. In: *Proceedings of the 27th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 337–344. ACM, New York (2004)
6. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *IEEE Computer* 42(8), 30–37 (2009)
7. Ma, H., King, I., Lyu, M.: Effective missing data prediction for collaborative filtering. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 46. ACM, New York (2007)
8. McLaughlin, M., Herlocker, J.: A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In: *Proceedings of the 27th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 329–336. ACM, New York (2004)
9. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. *Advances in neural information processing systems* 20, 1257–1264 (2008)
10. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web*, p. 295. ACM, New York (2001)
11. Wang, F., Ma, S., Yang, L., Li, T.: Recommendation on item graphs. In: Perner, P. (ed.) *ICDM 2006. LNCS (LNAI)*, vol. 4065, pp. 1119–1123. Springer, Heidelberg (2006)
12. Wang, J., De Vries, A., Reinders, M.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 508. ACM, New York (2006)

Reinforcement Learning by KFM Probabilistic Associative Memory Based on Weights Distribution and Area Neuron Increase and Decrease

Takahiro Hada and Yuko Osana

Tokyo University of Technology,
1401-1 Katakura, Hachioji, Tokyo, Japan
osana@cs.teu.ac.jp

Abstract. In this paper, we propose a reinforcement learning method using Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution and Area Neuron and Increase and Decrease (KFMPAM-WD-NID). The proposed method is based on the actor-critic method, and the actor is realized by the KFMPAM-WD-NID. The KFMPAM-WD-NID is based on the self-organizing feature map, and it can realize successive learning and one-to-many associations. Moreover, the weights distribution in the Map Layer can be modified by the increase and decrease of neurons in each area. The proposed method makes use of these properties in order to realize the learning during the practice of task. We carried out a series of computer experiments, and confirmed the effectiveness of the proposed method in the pursuit problem.

1 Introduction

The reinforcement learning is a sub-area of machine learning concerned with how an agent ought to take actions in an environment so as to maximize some notion of long-term reward [1]. Reinforcement learning algorithms attempt to find a policy that maps states of the world to the actions the agent ought to take in those states.

The Temporal Difference (TD) learning is one of the reinforcement learning algorithm. The TD learning is a combination of Monte Carlo ideas and dynamic programming (DP) ideas. TD resembles a Monte Carlo method because it learns by sampling the environment according to some policy. TD is related to dynamic programming techniques because it approximates its current estimate based on previously learned estimates. The actor-critic method [2] is the method based on the TD learning, and consists of two parts; (1) actor which selects the action and (2) critic which evaluate the action and the state.

On the other hand, neural networks are drawing much attention as a method to realize flexible information processing. Neural networks consider neuron groups of the brain in the creature, and imitate these neurons technologically. Neural networks have some features, especially one of the important features is that the networks can learn to acquire the ability of information processing. The flexible

information processing ability of the neural network and the adaptive learning ability of the reinforcement learning are combined, some reinforcement learning method using neural networks are proposed [3]-[5].

In this paper, we propose the reinforcement learning method using Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution Area Neuron Increase and Decrease (KFMPAM-WD-NID) [6]. The proposed method is based on the actor-critic method, and the actor is realized by the KFMPAM-WD-NID. The KFMPAM-WD-NID is based on the self-organizing feature map [7], and it can realize successive learning and one-to-many associations. Moreover, the weights distribution in the Map Layer can be modified by the increase and decrease of neurons in each area. The proposed method makes use of these properties in order to realize the learning during the practice of task.

2 KFM Probabilistic Associative Memory Based on Weights Distribution and Area Neuron Increase and Decrease

Here, we explain the Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution and Area Neuron Increase and Decrease (KFMPAM-WD-NID) [6] which is used in the proposed method.

2.1 Structure

The KFMPAM-WD-NID has two layers; (1) Input/Output(I/O) Layer and (2) Map Layer, and the I/O Layer is divided into some parts.

2.2 Learning Process

In the learning algorithm of the KFMPAM-WD-NID, the patterns are learned as follows:

- (1) In the network with the Map Layer composed of $x_{max} \times y_{max}$ neurons, the connection weights are initialized randomly. Here, x_{max} is the initial number of neurons of a horizontal direction, and y_{max} is the initial number of neurons of a vertical direction. In the initial state, $x_{max} \times y_{max}$ neurons are arranged at the coordinates $(0, 0), (1, 0), \dots, (x_{max} - 1, 0), (1, 0), \dots, (x_{max} - 1, y_{max} - 1)$.
- (2) The Euclidean distance between the learning vector $\mathbf{X}^{(p)}$ and the connection weights vector \mathbf{W}_i , $d(\mathbf{X}^{(p)}, \mathbf{W}_i)$ is calculated. If $d(\mathbf{X}^{(p)}, \mathbf{W}_i) > \theta^t$ is satisfied for all neurons, the input pattern $\mathbf{X}^{(p)}$ is regarded as an unknown pattern. If the input pattern is regarded as a known pattern, go to (6).
- (3) The neuron which is the center of the learning area c is determined as follows:

$$c = \underset{\substack{i : D_{iz} + D_{zi} \leq d_{iz} \leq D_{iz} + 1 \\ (\text{for } \forall z \in F)}}{\operatorname{argmin}} d(\mathbf{X}^{(p)}, \mathbf{W}_i). \quad (1)$$

In this equation, the neuron whose Euclidean distance between its connection weights and the learning vector is minimum in the neurons which can take areas without overlaps to the areas corresponding to the patterns which are already trained. In Eq.(1), F is the set of the weight-fixed neurons, d_{iz} is the distance between the neuron i and the weight-fixed neuron z . And D_{ij} is the radius of the ellipse area whose center is the neuron i for the direction to the neuron j , and is given by

$$D_{ij} = \begin{cases} \sqrt{\frac{a_i^2 b_i^2}{b_i^2 + m_{ij}^2 a_i^2} (m_{ij}^2 + 1)}, & (d_{ij}^x \neq 0 \text{ and } d_{ij}^y \neq 0) \\ a_i, & (d_{ij}^y = 0) \\ b_i, & (d_{ij}^x = 0) \end{cases} \quad (2)$$

where a_i and b_i are the long and short radius of the ellipse, and m_{ij} is the slope of the line through the neurons i and j , and is given by

$$m_{ij} = d_{ij}^y / d_{ij}^x, \quad (d_{ij}^x \neq 0). \quad (3)$$

- (4) If $d(\mathbf{X}^{(p)}, \mathbf{W}_c) > \theta^t$ is satisfied, the connection weights of the neurons in the ellipse whose center is the neuron c are updated as follows:

$$\mathbf{W}_i(t+1) = \begin{cases} \mathbf{X}^{(p)}, & (\theta_1^{learn} \leq H(\overline{d_{ci}})) \\ \mathbf{W}_i(t) + H(\overline{d_{ci}})(\mathbf{X}^{(p)} - \mathbf{W}_i(t)), & (\theta_2^{learn} \leq H(\overline{d_{ci}}) < \theta_1^{learn} \text{ and } H(\overline{d_{i^*i}}) < \theta_1^{learn}) \\ \mathbf{W}_i(t), & (\text{otherwise}) \end{cases} \quad (4)$$

where θ_1^{learn} and θ_2^{learn} are the thresholds, and i^* is the nearest weight-fixed neuron from the neuron i . $H(\overline{d_{ci}})$ and $H(\overline{d_{i^*i}})$ are the semi-fixed functions and are given by

$$H(\overline{d_{ci}}) = \frac{1}{1 + \exp((\overline{d_{ci}} - D)/\varepsilon)} \quad (5)$$

where ε is the steepness parameter of the function $H(\overline{d_{ci}})$, and D ($1 < D$) is the size of the neighborhood area. $\overline{d_{ci}}$ is the normalized distance between the center neuron of the area c and the neuron i , and is given by

$$\overline{d_{ci}} = d_{ci} / D_{ci}. \quad (6)$$

- (5) The connection weights of the neuron c \mathbf{W}_c are fixed.
 (6) (2)~(5) are iterated when a new pattern set is given.

2.3 Neuron Increase and Decrease

In the KFMPAM-WD-NID, the weights distribution in the Map Layer can be modified by the increase and decrease of neurons in each area.

Neuron Increase. In the KFMPAM-WD-NID, the neuron is added at the position corresponding to the neurons that exist in the initial Map Layer. When the N_z neurons whose connection weights are same as that of the center neuron are in the area z , the neuron is added as follows. Here, N_z^{ini} is the number of the neurons when the area z is generated, N_z^{min} is the minimum number of the neurons in the area z , and N_z^{max} is the maximum number of the neurons in the area z .

(1) $N_z^{ini} \leq N_z < N_z^{max}$. If $N_z^{ini} \leq N_z < N_z^{max}$ is satisfied, the new neuron is added in the area z .

The reference neuron j^* corresponding to the adding neuron j is given by

$$j^* = ((N_z - N_z^{ini}) \bmod (N_z^{ini} - 1)) + 1 \quad (7)$$

where mod shows remainder operation, and j^* shows the sequential serial number for the neurons except for the center neuron which exist when the area is generated (See Fig. 1). The coordinates of the new neuron is given by

$$x_j = x_{j^*} - \text{sgn}(x_{j^*}) \cdot (S_j^{add}/C_z) \quad (8)$$

$$y_j = y_{j^*} - \text{sgn}(y_{j^*}) \cdot (S_j^{add}/C_z) \quad (9)$$

where x_{j^*} , y_{j^*} are the coordinates of the reference neuron j^* , and $\text{sgn}(\cdot)$ is the sign function. C_z is the coefficient for the distance between neurons in the area z , and is given by

$$C_z = \lceil (N_z^{max} - N_z^{ini})/N_z^{ini} - 1 \rceil + 1. \quad (10)$$

S_j^{add} is

$$S_j^{add} = \lfloor (N_z - N_z^{ini})/N_z^{ini} - 1 \rfloor. \quad (11)$$

where $\lceil \cdot \rceil$ is the ceiling function and $\lfloor \cdot \rfloor$ is the floor function.

(2) $N_z^{min} < N_z \leq N_z^{ini}$. If the number of the neurons in the area N_z is smaller than N_z^{ini} , the connection weights of the neuron in the area (the neuron which satisfy $d_{ci} \leq D_{ci}$ and whose connection weights are same as that of the center neuron) are initialized randomly. The connection weights are updated in the neuron whose sequential serial number is $N_z - 1$.

Neuron Decrease. In the KFMPAM-WD-NID, the neuron is deleted from the neuron which is added finally.

2.4 Recall Process

In the recall process of the KFMPAM-WD-NID, when the pattern \mathbf{X} is given to the I/O Layer, the output of the neuron i in the Map Layer, x_i^{map} is calculated by

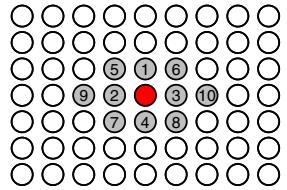


Fig. 1. Sequential Serial Number for Original Neurons in Area

$$x_i^{map} = \begin{cases} 1, & (i = r) \\ 0, & (\text{otherwise}) \end{cases} \quad (12)$$

where r is selected randomly from the neurons which satisfy

$$\frac{1}{N^{in}} \sum_{k \in C} g(X_k - W_{ik}) < \theta^{map} \quad (13)$$

where N^{in} is the number of neurons which receive the input in the I/O Layer. $g(\cdot)$ is given by

$$g(h) = \begin{cases} 1, & (|h| < \theta^d) \\ 0, & (\text{otherwise}) \end{cases} \quad (14)$$

where θ^d is the threshold, and θ^{map} is the threshold of the neuron in the Map Layer.

When the binary pattern \mathbf{X} is given to the I/O Layer, the output of the neuron k in the I/O Layer x_k^{io} is given by

$$x_k^{io} = \begin{cases} 1, & (W_{rk} \geq \theta_b^{in}) \\ 0, & (\text{otherwise}) \end{cases} \quad (15)$$

where θ_b^{in} is the threshold of the neurons in the I/O Layer.

When the analog pattern \mathbf{X} is given to the I/O Layer, the output of the neuron k in the I/O Layer x_k^{io} is given by

$$x_k^{io} = W_{rk}. \quad (16)$$

3 Reinforcement Learning Using KFMPAM-WD-NID

Here, we explain the proposed reinforcement learning method using KFMPAM-WD-NID[6].

3.1 Outline

In the proposed method, the actor in the actor-critic[2] is realized by the KFMPAM-WD-NID. In this research, the Input/Output Layer in the KFMPAM-WD-NID is divided into two parts corresponding to the state s and the action a , and the actions for the states are memorized.

In this method, the critic receives the states which are obtained from the environment, the state is estimated and the value function is updated. Moreover, the critic outputs the Temporal Difference (TD) error to the actor. The KFMPAM-WD-NID which behaves as the actor (we call this “actor network”) is trained based on the TD error, and selects the action from the state of environment.

3.2 Actor Network

In the proposed method, the actor in the Actor-Critic[2] is realized by the KFMPAM-WD-NID.

Dynamics. In the actor network, when the state \mathbf{s} is given to the Input/Output Layer, the corresponding action \mathbf{a} is recalled. In the proposed method, the other action is also selected randomly (random selection), and the more desirable action from the recalled action and the action selected in the random selection is chosen as the action finally.

When the pattern \mathbf{X} is given to the network, the output of the neuron i in the Map Layer at the time t $x_i^{map}(t)$ is given by Eq.(12).

In the recall process, the input which does not receive the pattern is set to -1 , and the winner neuron r is selected randomly from the neurons which satisfy the condition given by

$$\frac{1}{N^{in}} \sum_{\substack{k \in C \\ k : X_k(t) \neq -1}} g(X_k(t) - W_{ik}) \geq \theta^{map} \quad (17)$$

where N^{in} is the number of neurons which receive the input that not equal -1 , C is the set of neurons in the Input/Output Layer which receive the input that not equal -1 , θ^{map} is the threshold of the neuron in the Map Layer, and $g(\cdot)$ is the function which is given by Eq.(14). The input vector $\mathbf{X}(t)$ is given by

$$\mathbf{X}(t) = (\mathbf{s}(t) - 1 \cdots - 1)^T \quad (18)$$

For binary patterns, the output of the neuron k in the Input/Output Layer at the time t , $x_k^{io}(t)$ is given by

$$x_k^{io}(t) = \begin{cases} 1, & (0.5 \leq W_{rk}) \\ 0, & (0 \leq W_{rk} < 0.5) \\ -1, & (W_{rk} < 0). \end{cases} \quad (19)$$

For analog patterns, the output of the neuron k in the Input/Output Layer at the time t , $x_k^{io}(t)$ is given by

$$x_k^{io}(t) = \begin{cases} W_{rk}, & (0 \leq W_{rk}) \\ -1, & (W_{rk} < 0). \end{cases} \quad (20)$$

Learning. The actor network is trained based on the TD error from the critic.

The learning vector at the time t $\mathbf{X}^{(t)}$ is given by the state $\mathbf{s}(t)$ and the corresponding action $\mathbf{a}(t)$ as follows.

$$\mathbf{X}^{(t)} = (\mathbf{s}(t), \mathbf{a}(t))^T \quad (21)$$

(1) When Action is Selected by Actor Network

When the action $\mathbf{a}(t)$ is selected by the actor network and the pair of the state and the selected action are memorized in the actor network, the area size corresponding to the pair is updated. If the reward and the TD error are larger than 0 , the area is expanded (that is, neurons are added to the area). If the reward and the TD error are smaller than 0 , the area is reduced (that is, neurons are deleted from the area). When the action $\mathbf{a}(t)$ is selected by the actor network

and the pair of the state and the selected action are not memorized in the actor network and the TD error is larger than 0, the pair is memorized as a new pattern.

(1-1) When State and Action are Stored

(a) Reward and TD error are Larger than 0. If the reward and the TD error are larger than 0, neurons are added to the corresponding area z . The number of added neurons N_z^{add} is given by

$$N_z^{add} = \begin{cases} \lfloor \delta N_z^{add} \rfloor, & (N_z + \lfloor \delta N_z^{add} \rfloor \leq N_z^{max}) \\ N_z^{max} - N_z, & (\text{otherwise}) \end{cases} \quad (22)$$

where δ is the TD error, N_z^{add} is the coefficient, N_z is the number of neurons in the area z , N_z^{max} is the upper limit of the neurons in the area z .

(b) Reward and TD error are Smaller than 0. If the reward and the TD error are smaller than 0, neurons are deleted from the the corresponding area z . The number of deleted neurons N_z^{del} is given by

$$N_z^{del} = \begin{cases} \lfloor \delta N_z^{del} \rfloor, & (N_z - \lfloor \delta N_z^{del} \rfloor \geq N_z^{min}) \\ N_z - N_z^{min}, & (\text{otherwise}) \end{cases} \quad (23)$$

where N_z^{del} is the coefficient, and N_z^{min} is the lower limit of the number of neurons in the area z .

(1-2) When State and Action are not Stored and TD error is Larger than 0

When the pair of the state and the selected action are not memorized in the actor network and the TD error is larger than 0, the pair is trained as a new pattern.

(2) When Action is Selected Randomly

When the action $a(t)$ is selected randomly and the TD error is larger than 0, the pair of the state and the selected action is trained as a new pattern.

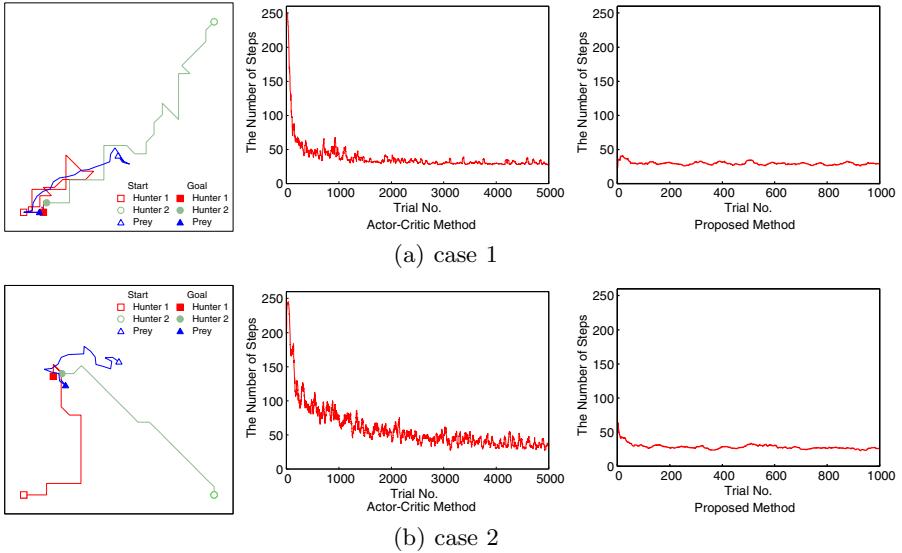
3.3 Reinforcement Learning Using KFMPAM-WD-NID

The flow of the proposed reinforcement learning method using KFMPAM-WD-NID is as follows:

- (1) The initial values of weights in the actor network are chosen randomly.
- (2) The agent observes the environment $s(t)$, and the actor $a(t)$ is selected by the actor network or the random selection.
- (3) The state $s(t)$ transits to the $s(t+1)$ by action $a(t)$.
- (4) The critic receives the reward $r(s(t+1))$ from the environment $s(t+1)$, and outputs the TD error δ to the actor.

$$\delta = r(s(t+1)) + \gamma V(s(t+1)) - V(s(t)) \quad (24)$$

where γ ($0 \leq \gamma \leq 1$) is the decay parameter, and $V(s(t))$ is the value function for the state $s(t)$.

**Fig. 2.** Trajectory and Transition of Steps

- (5) The value for state s $V(s)$ is updated by

$$V(s) \leftarrow V(s) + \xi \delta \quad (25)$$

where ξ ($0 \leq \xi \leq 1$) is the learning rate.

- (6) The connection weights in the actor network are updated based on the reward and the TD error (See 3.2).
- (7) Back to (2).

4 Computer Experiment Results

Here, we show the computer experiment results to demonstrate the effectiveness of the proposed method. We applied the proposed method to the pursuit problem. In this experiment, two hunter agents pursue a prey. The agent can observe the distance to a prey, and can move eight-ways. Figure 2 shows the routes of the hunter agents and the prey and the transition of number of average steps in the same trials. We carried out the similar experiments under the various conditions, and confirmed that the proposed method can learn faster than the conventional actor-critic method.

5 Conclusions

In this paper, we have proposed the reinforcement learning method using KFMPAM-WD-NID. The proposed method is based on the actor-critic method, and the actor is realized by the KFMPAM-WD-NID. We carried out a series of computer experiments, and confirmed the effectiveness of the proposed method in pursuit problem.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning, An Introduction. The MIT Press, Cambridge (1998)
2. Witten, I.H.: An adaptive optimal controller for discrete-time Markov environments. *Information and Control* 34, 286–295 (1977)
3. Ishii, S., Shidara, M., Shibata, K.: A model of emergence of reward expectancy neurons by reinforcement learning. In: Proceedings of the 10th International Symposium on Artificial Life and Robotics, GS21-5 (2005)
4. Osana, Y.: Reinforcement learning using Kohonen feature map probabilistic associative Memory based on weights distribution. Nolta, Sapporo (2009)
5. Noguchi, S., Osana, Y.: Improved Kohonen feature map probabilistic associative memory based on weights distribution. IJCNN, Barcelona (2010)
6. Hada, T., Osana, Y.: Kohonen feature map probabilistic associative memory based on weights distribution and area neuron increase and decrease. Nolta, Krakow (2010)
7. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (1994)

Extraction of Reward-Related Feature Space Using Correlation-Based and Reward-Based Learning Methods

Poramate Manoonpong^{2,1}, Florentin Wörgötter¹, and Jun Morimoto²

¹ Bernstein Center for Computational Neuroscience (BCCN),

III. Institute of Physics,

University of Göttingen, Göttingen 37077, Germany

{poramate,worgott}@physik3.gwdg.de

² ATR Computational Neuroscience Laboratories, 2-2-2 Hikaridai Seika-cho,

Soraku-gun, Kyoto 619-0288, Japan

xmorimo@atr.jp

Abstract. The purpose of this article is to present a novel learning paradigm that extracts reward-related low-dimensional state space by combining correlation-based learning like Input Correlation Learning (ICO learning) and reward-based learning like Reinforcement Learning (RL). Since ICO learning can quickly find a correlation between a state and an unwanted condition (e.g., failure), we use it to extract low-dimensional feature space in which we can find a failure avoidance policy. Then, the extracted feature space is used as a prior for RL. If we can extract proper feature space for a given task, a model of the policy can be simple and the policy can be easily improved. The performance of this learning paradigm is evaluated through simulation of a cart-pole system. As a result, we show that the proposed method can enhance the feature extraction process to find the proper feature space for a pole balancing policy. That is it allows a policy to effectively stabilize the pole in the largest domain of initial conditions compared to only using ICO learning or only using RL without any prior knowledge.

Keywords: Unsupervised learning, Reinforcement learning, Neural control, Sequential combination, Pole balancing.

1 Introduction

Living creatures, like humans and animals, can effectively learn solving a variety of tasks. They can learn a correlation between an earlier stimulus (called predictive signal) and a later one (called reflex signal) to react to the earlier stimulus, not having to wait for the later signal. For example, heat radiation (predictive signal) precedes a pain signal (reflex signal) when touching a hot surface. Thus, they learn an anticipatory action to avoid the late unwanted stimulus (i.e., avoiding to touch the hot surface). Such a learning mechanism is known as correlation-based learning (or temporal sequence learning). Furthermore, the

living creatures also have the ability to learn to react appropriately to particular stimuli on the basis of associated rewards or punishments. This kind of learning strategy is minimally supervised since they are not explicitly taught. Instead, they must work this out for themselves on the basis of their past experiences (exploitation), new choices (exploration), and the reinforcement. This learning mechanism is known as reinforcement learning (RL). These two biological learning mechanisms have been applied to Artificial Intelligence (AI) from several points of view including the development of adaptive autonomous robots [1]. Most AI studies have *separately* used such learning mechanisms to allow robots to learn solving their tasks [2][3]. As a consequence, they might fail to solve some tasks required evaluative feedback when using only correlation-based learning. On the other hand, using RL without any prior knowledge (predefined control parameters) for high-dimensional continuous-state systems often requires long learning times. Thus a number of investigators have focused on building various low-level control parameters before applying RL [4][5].

In contrast to the robot learning strategy, living creatures probably combine both learning mechanisms in a way that fast correlation-based learning automates their intuition (i.e., providing low-level control parameters) which will guide RL for effectively solving complex tasks. Following this, we propose here how correlation-based learning (e.g., input correlation learning (ICO learning) [2]) and RL (e.g., actor-critic RL [6]) can be combined in a sequential way such that this learning paradigm extracts reward-related features to allow a policy to accomplish a given task. If we can extract proper feature space for a given task, a model of the policy can be simple, i.e., the policy with small number of parameters can be used to accomplish the given task. Advantage of using the simple policy is that it can be easily improved. We have chosen a pole balancing problem as a first test since balancing an inverted pendulum provides a well known class of control problems and often serves as a benchmark problem for dynamical control. However, the main purpose of this article is not to demonstrate the use of the combination between ICO learning and actor-critic RL for the pole balancing system but to suggest that this learning paradigm can be an efficient way to find reward-related feature space to solve dynamic sensorimotor control problems. Note that in this study, policy parameters are fixed and only the feature space is updated.

Before presenting the proposed learning strategy and its performance, in the following section we first show how ICO leaning can be applied and quickly learn to find the feature space for the pole balancing problem. Afterwards in section 3 we show how we sequentially combine ICO learning with actor-critic RL to modify the feature space to achieve better task performance. In section 4 we provide comparison results of different learning mechanisms, followed by conclusions.

2 Correlation-Based Learning to Extract Feature Space

Here we present how ICO learning can be applied to extract the reward-related feature space where its learning rule considers only cross-correlating two types

of input signals with each other: earlier signals and a later one. As a concrete example, we consider the pole balancing problem [7] (see Fig. 11a). The task is to balance an inverted pendulum, which is mounted on a cart moving freely in a one-dimensional interval, and to simultaneously avoid the interval boundaries. The pole is free to move only in the vertical plane of the cart and track. This cart-pole system is simulated on a desktop PC with dual-core Intel processors at 2.4 GHz and updated by using Euler discretization with time steps of 0.01 s. The system provides four state variables: angle of the pole with the vertical (θ), pole angular velocity ($\dot{\theta}$), position of the cart on the track (x), and cart velocity (\dot{x}). The cart is bound to move in the interval $-2.4 < x < 2.4$ [m] and the angle is allowed to vary in the interval $-12 < \theta < 12$ [$^{\circ}$]. The simulated model includes all nonlinearities of the physical system (see [7] for the equations of this physical cart-pole system).

The feature extraction method based on ICO learning (see Fig. 11a) for this dynamical system is modelled as a linear projection of four earlier signals (called predictive signals) which are the state variables ($\theta, \dot{\theta}, x, \dot{x}$) to one-dimensional feature space. To update the feature space, we use a later signal (called a reward (penalty) signal, r) which is a signal given just before the system fails. The reward signal has a negative value (-1.0), if $x < -2.35$ m, $x > 2.35$ m, $\theta > 11.5^{\circ}$, or $\theta < -11.5^{\circ}$, and 0 otherwise. All the state inputs ($\theta, \dot{\theta}, x, \dot{x}$) are scaled onto the interval $[-1, 1]$ as described in [8]. A projection from original state space to the low-dimensional feature space \mathcal{Z} is specified by:

$$z(t) = \mathbf{w}^T \mathbf{x}(t), \quad (1)$$

where $z \in \mathcal{Z} \subset \mathcal{R}$, and \mathbf{x} is the original state vector while \mathbf{w} represents synaptic weights (projection vector). These weights which are initially set to 0.0 get changed by ICO learning using the cross-correlation between the predictive signals and a change of the reward signal. They are given by:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu |\mathbf{x}(t)| q, \quad (2)$$

$$q = |\min(0, \Delta r(t))|, \quad (3)$$

where $\Delta r(t) = r(t+1) - r(t)$ denotes the change of the reward signal and $\mu = 1.0 \times 10^{-4}$ is a learning rate.

Here we consider the reflex output $U(t)$ as a part of control output as suggested in [2]:

$$U(t) = \begin{cases} 1.0 & : x(t) < -2.35 \text{ m or } \theta(t) > 11.5^{\circ} \\ -1.0 & : x(t) > 2.35 \text{ m or } \theta(t) < -11.5^{\circ} \\ 0.0 & : \text{otherwise.} \end{cases} \quad (4)$$

By using domain knowledge, we can construct a failure avoidance policy of the cart-pole system in the low-dimensional feature space:

$$u(t) = Gz(t) + U(t), \quad (5)$$

where G is a parameter of the policy (i.e., gain in this case). Here, it is set to 10.0 [7]. According to this setup ICO learning will gradually develop the

synaptic weights \mathbf{w} (see Fig. II) to obtain the appropriate projection vector from the original state space to the feature space for balancing the pole and also avoiding the cart to hit the interval boundaries (i.e., failure avoidance policy).

To test the performance of the proposed feature extraction method using ICO learning for this dynamical pole balancing task, we let it learn to balance the pole on 25×49 initial conditions (θ, x) represented by squares in Fig. IIb while $\dot{\theta}$ and \dot{x}

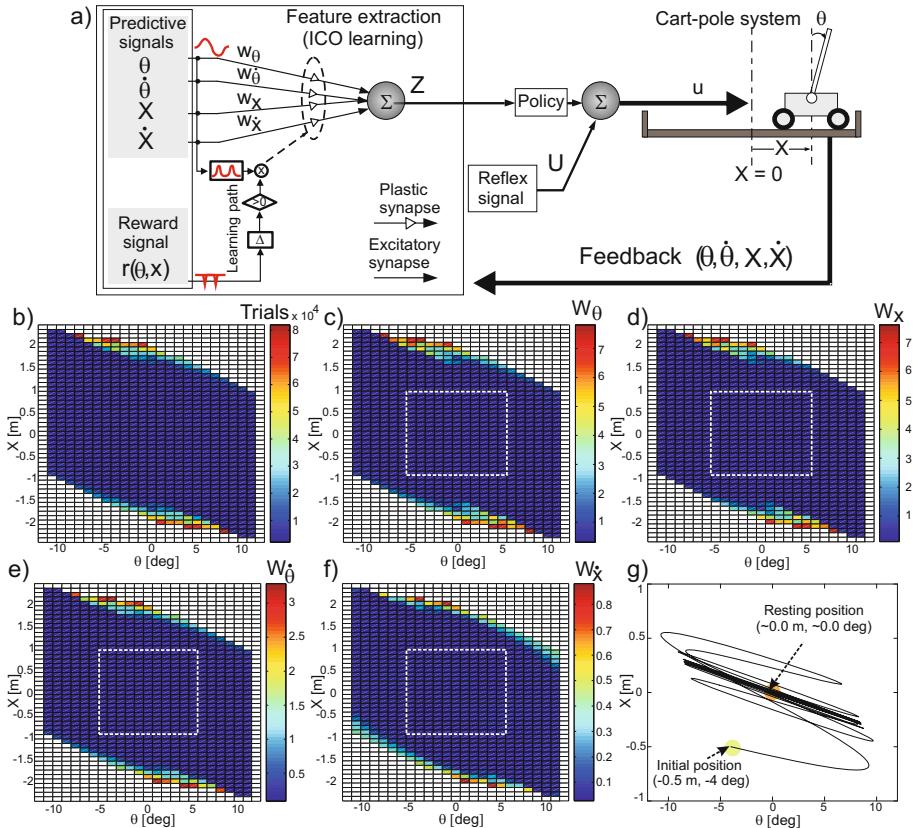


Fig. 1. (a) Feature extraction using ICO learning for the cart-pole system. Only one learning path instead of four is shown for clarity (see text for more details). (b) Performance of the policy on benchmark initial conditions. Colored area denotes a typical domain for successful control ($\approx 68\%$) where the color-coded bar presents the number of trials from start to success. White area represents a domain in which the policy fails to balance. (c)–(f) Resulting weights, i.e., projection vectors, ($w_\theta, w_x, w_{\dot{\theta}}, w_{\dot{x}}$) after learning. Note that all weight values in the white lower left and upper right corners, where the system fails, are removed for clarity. The area inside a dashed frame of each diagram shows the weights which will serve to generate prior weight distribution for the reward-based learning. (g) Path in (x, θ) -space for arbitrary initial condition ($x = -0.5 \text{ m}$, $\theta = -4 \text{ deg}$) under control of the weight distribution.

are initially set to small random values representing the system noise. Each trial during a run starts with a given initial state and ends either in “success” (which occurs when the pole is kept in balance for at least 1000 seconds) or “failure” (which occurs when the pole falls 12 deg to either side or the cart moves 2.4 m to either side). A run at each initial condition is terminated when either a successful trial achieved or the maximum number of trials is reached (e.g., here 1.0×10^5 trials). During this learning process the system is reset to the same initial state at failure. We observe that the policy defined in the extracted feature space (see Eq. (5)) is able to balance the pole and avoid the ends of the interval in a relatively large (x, θ) -domain of initial conditions (see Fig. IIb). It is important to note that each trial generally uses so much less computing power resulting in a fast learning speed compared to other techniques (see comparison section below). Figures IIc–f show the resulting projection vector (or learned weights, $w_\theta, w_x, w_{\dot{\theta}}, w_{\dot{x}}$) in the successful domain. Figure IIg exemplifies the behavior of the system displaying a path in (x, θ) -space for arbitrary initial condition (e.g., $x = -0.5$ m, $\theta = -4$ deg) of the cart-pole system.

3 Reward-Based Learning to Extract Feature Space

As shown in a previous section, one can see that the reward-related feature space extracted by ICO learning can be efficiently used for the pole balancing problem in a relatively large domain of initial conditions (see Fig. IIb). However, it still fails to stabilize the system at initial conditions in the critical corners of the benchmark domain (upper-right and lower-left areas in Fig. IIb). This is because correlation-based learning can only adapt the weights by recognizing a correlation between immediate reward (punishment) and it can not evaluate future (delayed) reward. Thus here we investigate whether the extracted feature space can be further modified so that the policy can stabilize the system in this domain.

To do so, we apply (continuous-state) reinforcement learning (RL) [6], [7] since its learning rule considering an association between stimuli and/or actions with the reinforcement that an agent receives can evaluate the future (delayed) reward. We use the actor-critic type RL that can be divided into two sub-mechanisms: the learning of the feature space (actor) and the learning of an evaluation function (critic). The feature extraction part is designed to have the same circuit as the feature extraction process of ICO learning (compare Figs. IIa and 2a).

For the critic network, we use a normalized Gaussian neural network (see Fig. 2a) as a function approximator to represent the value function or the prediction (V , see [6] for more details including equations). In this cart-pole system, the network has 162 hidden neurons ($H_{1,\dots,162}$, see Fig. 2a) where centers are fixed on a grid according to the boxes approach [7]. The learning rate of this critic network is manually adjusted. It is set to, e.g., 0.6. The TD error is computed from the prediction as $\delta(t) = r(t) + \gamma V(t) - V(t-1)$. r is an external reinforcement signal (-1 when failure occurs, 0 otherwise) [7]. γ is a discount factor, i.e., distant rewards are less important. We set it to 0.95 based on [7].

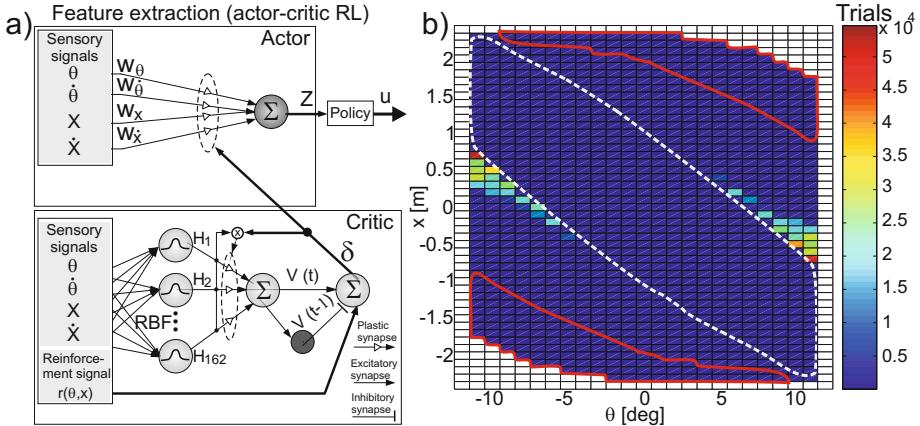


Fig. 2. (a) Feature extraction using actor-critic RL for the cart-pole system. (b) Performance of the policy defined in the extracted feature space. The prior weight distribution are given by the learned weights by ICO learning. Colored area denotes a typical domain for successful control ($\approx 84\%$) where the color-coded bar presents the number of trials until success. White area represents the domain in which the policy fails to balance. Area inside a white dashed frame shows the initial conditions on which the policy can stabilize the system without actor-critic RL. Area inside red frames shows improvement of the feature space achieved by actor-critic RL.

Here we modify the feature space \mathcal{Z} through Bayesian update given by:

$$P(\mathbf{w}|u^{ob}, \mathbf{x}) = \frac{P(u^{ob}|\mathbf{w}, \mathbf{x})P(\mathbf{w})}{\int P(u^{ob}|\mathbf{w}, \mathbf{x})P(\mathbf{w})d\mathbf{w}}, \quad (6)$$

where $P(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_{\mathbf{w}}, \Sigma_{\mathbf{w}})$ is the prior probability of weight distribution given by ICO learning for the first iteration. Since ICO learning tends to acquire different weighs when the learning process starts from different initial conditions, we estimate mean $\boldsymbol{\mu}_{\mathbf{w}}$ and variance $\Sigma_{\mathbf{w}}$ from the learned weight vectors. $P(u|\mathbf{w}, \mathbf{x}) = \mathcal{N}(u|\mu_u, \sigma_u)$ is the control policy. Note that we use the result of ICO learning as the prior only for the first update. From the second update, we use the weight distribution derived in previous iteration as the prior.

Here we consider the feature $z = \mathbf{w}^T \mathbf{x}$. The mean output of the policy is designed as $\mu_u = Gz$. The observation of the control output u^{ob} in Eq. (6) is acquired from the RL framework:

$$u^{ob} = \mu_u + \Delta u, \quad (7)$$

where

$$\Delta u = \alpha \delta(u - \mu_u). \quad (8)$$

Here δ is the TD error and α is a scaling parameter that corresponds to the learning rate.

Figure 2b shows the performance of the policy defined in the feature space using actor-critic RL where the prior weight distribution $P(\mathbf{w})$ are derived from ICO learning. It can be seen that the policy can now stabilize the system in a larger domain ($\approx 84\%$, i.e., $\approx 16\%$ more, see red frames in Fig. 2b) including some parts of the critical initial conditions. The remaining parts (white areas) seems to be difficult to achieve by using linear control. The results we obtained here are comparable to [8] where linear control using an evolutionary algorithm for weight adaptation is employed. Furthermore, we observe that actor-critic RL only starts to optimize the weights in the domain where the initial feature space is not proper to stabilize the system (colored area outside a white dashed frame in Fig. 2b). In fact there are a few initial conditions of the system where actor-critic RL requires a lot of trials (> 2000 trials, see triangular areas near the dashed frame in Fig. 2b) while most of them can be achieved after around 5–2000 trials. By contrast, if the actor weights are not appropriately given at the beginning (e.g., initially setting them to 0.0) actor-critic RL needs to learn in the whole domain. It also requires much more trials for each given initial condition and the policy can stabilize the system in a smaller domain (see Fig. 3). From this point of view, our experimental results suggest that providing the appropriate prior to actor-critic RL can speed up the learning process and also allows it to efficiently extract feature space.

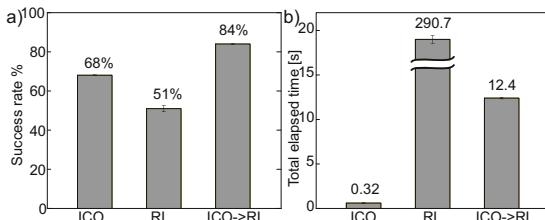


Fig. 3. (a) Histogram showing the average of success rate of each learning model; i.e., percentage of success in the total 25×49 set of initial conditions $((x, \theta)\text{-domain})$. (b) Histogram showing the average of the total elapsed simulated physical time of all success for each learning model. The elapsed times are recorded from starting until ending in success where failure cases are ignored. *ICO* and *RL*: the feature extraction using ICO learning and actor-critic RL, respectively, where all weights are initially set to 0.0, *ICO* → *RL*: the feature extraction using actor-critic RL but all weights are predefined by the weight distribution obtained from ICO learning. Note that in this comparison, all learning models use the same parameters, like learning rate and discount factor.

4 Comparison Results

In this section, we compare the performance of this learning model with the original ones. The results are shown in Fig. 3. It can be seen that ICO learning can quickly learn to find appropriate feature space in a relatively large domain

of initial conditions while actor-critic RL is very slow and can achieve success only in a smaller domain if we limit the maximum number of trials. However, the performance of the policy can be strongly improved by using prior weight distribution generated by ICO learning for actor-critic RL. As a consequence, the policy succeeds for the larger initial condition domain.

5 Conclusions

In this study, we proposed a new learning paradigm that sequentially combines ICO learning and actor-critic RL to extract feature space for a dynamical system. In concrete, we consider the pole balancing task as the dynamical system. To a certain extent the experimental studies pursued here sharpen our understanding of how correlation-based learning can be combined with RL to find the low-dimensional feature space. In future work, we will investigate the theoretical properties of this learning model and its dynamical behavior. We will also apply this learning strategy to real robotic tasks, like adaptive walking or mobile robot control.

Acknowledgments. This research was supported by Japan Society for the Promotion of Science (JSPS), the EU PACO-PLUS project, Bernstein Center for Computational Neuroscience II Göttingen (BCCN, project D1), the German Research Foundation (DFG, SE1042/05) and the Strategic Research Program for Brain Sciences (SRPBS).

References

1. Pfeifer, R., Lungarella, M., Iida, F.: Self-Organization, Embodiment, and Biologically Inspired Robotics. *Science* 318(5853), 1088–1093 (2007)
2. Porr, B., Wörgötter, F.: Strongly Improved Stability and Faster Convergence of Temporal Sequence Learning by Using Input Correlations Only. *Neural Comput.* 18(6), 1380–1412 (2006)
3. Phon-Amnuaisuk, S.: Learning Cooperative Behaviours in Multiagent Reinforcement Learning. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) *ICONIP 2009. LNCS*, vol. 5863, pp. 570–579. Springer, Heidelberg (2009)
4. Kolter, J.Z., Ng, A.Y.: Policy Search via the Signed Derivative. In: *The Proc. of Robotics: Science and Systems (RSS)*, p. 27 (2009)
5. Melo, F.S., Lopes, M., Santos-Victor, J., Ribeiro, M.I.: A Unified Framework for Imitation-Like Behaviours. In: *The Proc. of 4th International Symposium on Imitation in Animals and Artifacts*, pp. 241–250 (2007)
6. Doya, K.: Reinforcement Learning in Continuous Time and Space. *Neural Comput.* 12, 219–245 (2000)
7. Barto, A.G., Sutton, R.S., Anderson, C.: Neuron-Like Adaptive Elements That Can Solve Difficult Learning Control Problems. *IEEE Transactions on Systems, Man, and Cybernetics* 13, 834–846 (1983)
8. Pasemann, F.: Evolving Neuropolicies for Balancing an Inverted Pendulum. *Network: Computation in Neural Systems* 9, 495–511 (1998)

Stationary Subspace Analysis as a Generalized Eigenvalue Problem

Satoshi Hara¹, Yoshinobu Kawahara¹, Takashi Washio¹, and Paul von Bünaus²

¹ The Institute of Scientific and Industrial Research (ISIR), Osaka University, Japan
`{hara,kawahara,washio}@car.sanken.osaka-u.ac.jp`

² Machine Learning Group, Computer Science Department, TU Berlin, Germany
`buenau@cs.tu-berlin.de`

Abstract. Understanding non-stationary effects is one of the key challenges in data analysis. However, in many settings the observation is a mixture of stationary and non-stationary sources. The aim of Stationary Subspace Analysis (SSA) is to factorize multivariate data into its stationary and non-stationary components. In this paper, we propose a novel SSA algorithm (ASSA) that extracts stationary sources from multiple time series blocks. It has a globally optimal solution under certain assumptions that can be obtained by solving a generalized eigenvalue problem. Apart from the numerical advantages, we also show that compared to the existing method, fewer blocks are required in ASSA to guarantee the identifiability of the solution. We demonstrate the validity of our approach in simulations and in an application to domain adaptation.

Keywords: Non-stationarity, Dimensionality Reduction, Feature Extraction, Eigenvalue Problem, Stationary Subspace Analysis.

1 Introduction

Non-stationarities are ubiquitous phenomena in real world data such as e.g. biomedical measurement [1], stock market time series [2] or web mining data [3]. Discerning stationary from non-stationary components in multivariate time series is a key step towards characterizing the underlying mechanisms [24].

In the Stationary Subspace Analysis (SSA) [4] model, the observation is a linear mixture of stationary and non-stationary sources. The aim of SSA is to estimate a linear transformation which separates the two groups of sources. This separation is useful in a wide range of applications, e.g. for removing non-stationary effects to make prediction methods robust under distribution changes [4,5] or reducing the number of dimensions when only the stationary or non-stationary part is of interest.

The SSA algorithm proposed in [4] (KL-SSA) is based on finding ergodic and weakly stationary sources [6] from multiple time series blocks. This is a non-convex optimization problem which requires an iterative optimization procedure that can get stuck in local optima. Therefore, we propose a novel algorithm, Analytic SSA (ASSA), where the unique global solution is obtained by solving

a generalized eigenvalue problem. The solution is optimal under the assumption that the stationary and non-stationary sources are uncorrelated. The results of our simulations show that ASSA significantly outperforms KL-SSA when the assumption is correct and still yields reasonable results when it does not hold exactly. Moreover, we prove that ASSA requires fewer time series blocks than KL-SSA to guarantee the identifiability of the solution. This is particularly relevant in practice, where data is often scarce or costly to obtain.

The remainder of this paper is structured as follows. In Section 2, we review the SSA model and the algorithm presented in [4]; the novel ASSA algorithm is introduced in Section 3, followed by the experimental results described in Section 4 and our conclusions in Section 5.

2 Stationary Subspace Analysis

In this section, we review the problem formulation of SSA and the KL-SSA algorithm [4]. We also explain the phenomenon of spurious stationarity, which occurs when there are not enough number of time series blocks.

2.1 Problem Formulation

In the SSA setting, we have a d -dimensional time series and split it into N consecutive blocks where the observation in the i -th block at time point t is modeled as a linear mixture of stationary and non-stationary components:

$$\mathbf{x}_i(t) = \begin{bmatrix} A^s & A^n \end{bmatrix} \begin{bmatrix} s_i^s(t) \\ s_i^n(t) \end{bmatrix} \quad \text{with } 1 \leq i \leq N, \quad (1)$$

where $s_i^s(t) \in \mathbb{R}^m$ and $s_i^n(t) \in \mathbb{R}^{(d-m)}$ are latent random vectors, $A^s \in \mathbb{R}^{d \times m}$ and $A^n \in \mathbb{R}^{d \times (d-m)}$ are mixing matrices and $[A^s \ A^n]$ is invertible. We refer to s_i^s and s_i^n as s -sources (stationary sources) and n -sources (non-stationary sources) respectively. The goal of SSA is to separate the s -sources from the n -sources, i.e. to find a demixing matrix $W \in \mathbb{R}^{m \times d}$ such that $\hat{s}_i^s(t) = W\mathbf{x}_i(t)$. In order to remove the non-stationary effects, W^\top must lie in the orthogonal complement of A^n , i.e. $WA^n = 0_{m \times (d-m)}$ where 0 is a zero matrix. Given the SSA model, the true s -sources are identifiable up to linear transformations [7].

2.2 The KL-SSA Algorithm

In the KL-SSA algorithm [4], the s -sources are assumed to be ergodic and weakly stationary [6], i.e. the mean and covariance matrix remains constant over time and are asymptotically computable from the given time series. These assumptions lead to the following two conditions on the demixing matrix W that the estimated s -sources have constant mean and covariance matrix both in each block and over the entire time series:

$$W\mu_i = W\bar{\mu}, \quad (2)$$

$$W\Sigma_i W^\top = W\bar{\Sigma}W^\top \quad (3)$$

where μ_i and Σ_i are the time average of mean $\mu_i = \mathbb{E}_t[\mathbf{x}_i(t)]$ and covariance matrix $\Sigma_i = \text{Cov}_t[\mathbf{x}_i(t)]$ in each block $1 \leq i \leq N$, $\bar{\mu}$ and $\bar{\Sigma}$ are the average mean and covariance matrix over the entire time series, $\bar{\mu} = \frac{1}{N} \sum_{i=1}^N \mu_i$, $\bar{\Sigma} = \frac{1}{N} \sum_{i=1}^N \Sigma_i$. The KL-SSA algorithm finds the demixing matrix W by minimizing the Kullback-Leibler divergence between the Gaussians $\mathcal{N}(W\mu_i, W\Sigma_i W^\top)$ and the Gaussian $\mathcal{N}(W\bar{\mu}, W\bar{\Sigma}W^\top)$,

$$\begin{aligned}\widehat{W}_{\text{KL}} &= \underset{W \in \mathbb{R}^{m \times d}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N \left\{ \|W(\mu_i - \bar{\mu})\|^2 - \log \det(W\Sigma_i W^\top)\right\} \\ &\text{subject to } W\bar{\Sigma}W^\top = I_m\end{aligned}\quad (4)$$

where I_m is the $m \times m$ identity matrix. The constraint term is imposed to limit the scaling freedom of W . In practice, the mean and covariance are replaced with their empirical estimators $\hat{\mu}_i$ and $\hat{\Sigma}_i$. The optimization problem (4) is non-convex and a local solution is derived by a gradient-based method [4]. The convergence to the global solution is not guaranteed in general, therefore the optimization procedure is repeated several times with different initializations.

2.3 Spurious Stationarity

The feasibility of KL-SSA depends on the number of n -sources $d - m$ and the number of time series blocks N . If N is too small, there are spurious stationary sources in which their distribution seem to be stationary over the entire time series, which actually contain contributions from non-stationary sources. If spurious stationary sources exist, the true stationary sources cannot be identified. The following theorem [47] tells us the number of time series blocks necessary to guarantee the identifiability of the true stationary sources.

Theorem 1 (Spurious Stationarity in KL-SSA). *In the KL-SSA setting, given d -dimensional observation with m stationary sources, the number of time series blocks N required to avoid the existence of spurious stationary sources is*

$$N \geq \frac{d - m + 1}{2} + 1 . \quad (5)$$

3 Analytic SSA as a Generalized Eigenvalue Problem

Since KL-SSA involves a non-convex optimization problem, it may not find the global optimum due to the existence of local optima. Moreover, the iterative procedure is computationally fairly costly. Therefore, we propose a novel SSA algorithm, Analytic SSA (ASSA), which is based on an additional assumption that s - and n -sources are group-wise uncorrelated. This assumption leads to a solution that can be obtained by solving a generalized eigenvalue problem.

In the KL-SSA setting, the non-convexity is caused by (3) where each condition defines a non-convex feasible region of the solution. The following theorem shows that we can avoid this non-convexity if we impose the ASSA assumption.

Theorem 2. If the \mathfrak{s} - and the \mathfrak{n} -sources are uncorrelated, the following condition is equivalent to condition (3):

$$W\Sigma_i = W\bar{\Sigma}. \quad (6)$$

Proof. If \mathfrak{s} - and \mathfrak{n} -sources are uncorrelated, their covariances are zeroes. Therefore, the covariance matrix Σ_i is in the form of

$$\Sigma_i = [A^{\mathfrak{s}} \ A^{\mathfrak{n}}] \begin{bmatrix} \Sigma^{\mathfrak{s}\mathfrak{s}} & 0_{m \times (d-m)} \\ 0_{(d-m) \times m} & \Sigma_i^{\mathfrak{n}\mathfrak{n}} \end{bmatrix} [A^{\mathfrak{s}} \ A^{\mathfrak{n}}]^{\top} \quad (7)$$

where $\Sigma^{\mathfrak{s}\mathfrak{s}}$ and $\Sigma_i^{\mathfrak{n}\mathfrak{n}}$ are the covariance matrices of \mathfrak{s} - and \mathfrak{n} -sources respectively. Since $\Sigma^{\mathfrak{s}\mathfrak{s}}$ is common for the entire time series, its index i is omitted. Because W^{\top} is an orthogonal complement of $A^{\mathfrak{n}}$, $W\Sigma_i = WA^{\mathfrak{s}}\Sigma^{\mathfrak{s}\mathfrak{s}}A^{\mathfrak{s}\top}$ holds which is independent of the index i and thus the condition (6) follows. The opposite (6) \Rightarrow (3) is obvious. \square

The condition (6) is linear over W and defines a convex feasible region of the solution. Therefore, we adopt it as a replacement for (3) for the SSA solution. Then, the following Analytic SSA (ASSA) is derived as a second order Taylor approximation of KL-SSA (4):

$$\begin{aligned} \widehat{W}_{\text{ASSA}} &= \underset{W \in \mathbb{R}^{m \times d}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N \left\{ \|W(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})\|^2 + \frac{1}{2} \operatorname{tr} \left[W(\Sigma_i - \bar{\Sigma}) \bar{\Sigma}^{-1} (\Sigma_i - \bar{\Sigma}) W^{\top} \right] \right\} \\ &= \underset{W \in \mathbb{R}^{m \times d}}{\operatorname{argmin}} \operatorname{tr} [WSW^{\top}] \\ &\text{subject to } W\bar{\Sigma}W^{\top} = I_m \end{aligned} \quad (8)$$

where $\operatorname{tr} [\cdot]$ denotes the trace of a matrix and the matrix S is defined as

$$S = \frac{1}{N} \sum_{i=1}^N \left\{ \boldsymbol{\mu}_i \boldsymbol{\mu}_i^{\top} + \frac{1}{2} \Sigma_i \bar{\Sigma}^{-1} \Sigma_i \right\} - \bar{\boldsymbol{\mu}} \bar{\boldsymbol{\mu}}^{\top} - \frac{1}{2} \bar{\Sigma}. \quad (9)$$

In the ASSA objective function (8), the log-term in KL-SSA is replaced with the variance of covariance, i.e. ASSA seeks the transformation W so that the variance of the moments across time series blocks is minimized. The Lagrangian of (8) is

$$\mathcal{L}(W, \Lambda) = \operatorname{tr} [WSW^{\top}] - \operatorname{tr} [\Lambda(W\bar{\Sigma}W^{\top} - I_m)] \quad (10)$$

where $\Lambda \in \mathbb{R}^{m \times m}$ is the Lagrange multiplier. By setting its derivative to zero, we obtain the following generalized eigenvalue problem:

$$S\boldsymbol{\varphi} = \lambda \bar{\Sigma} \boldsymbol{\varphi}. \quad (11)$$

Let $\{\boldsymbol{\varphi}_j\}_{j=1}^d$ be the generalized eigenvectors associated with the generalized eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_d$ of (11), then $\widehat{W}_{\text{ASSA}}$ is analytically given as:

$$\widehat{W}_{\text{ASSA}} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \dots, \boldsymbol{\varphi}_m]^{\top}. \quad (12)$$

Thanks to this analytical solution, ASSA avoids the local solution problem and does not require any iterations of the optimization procedure. Furthermore, there is no need of a priori choice of algorithmic parameters such as initial values or a number of optimization step. This is a great advantage of ASSA over KL-SSA.

Note the replacement of the condition (3) with its alternative (6) is acceptable in a rigorous sense when the correlation between s - and n -sources are low. We further investigate this in simulations in Section 4.1.

3.1 Spurious Stationarity in ASSA

Similar to the KL-SSA setting, the solution to ASSA is not determined if there exist spurious stationary sources. For ASSA, we can show that fewer time series blocks are required to uniquely identify the true solution than for KL-SSA, given that the s - and the n -sources are sufficiently uncorrelated.

Theorem 3 (Spurious Stationarity in ASSA). *The number of time series blocks N required to guarantee that there exists no spurious stationary sources in d dimensions with $d - m$ non-stationary sources is given as follows.*

$$N \geq \frac{2(d - m + 1)}{\ell + 1} \quad (13)$$

where $\ell = \frac{1}{N} \sum_{i=1}^N \text{rank}(\Sigma_i - \bar{\Sigma})$.

Proof. The set of ASSA solutions \mathcal{W} is determined by the conditions (2) and (6). If $\text{span}(W^\top) = \text{span}(W'^\top)$ holds for any solutions $W, W' \in \mathcal{W}$, then the s -sources are uniquely recovered up to linear transformations. It holds when $w \in \text{span}(W^\top)$ has degrees of freedom equal to or less than $m - 1$ for any $W \in \mathcal{W}$, where the -1 stems from the constraint term in (8).

The conditions (2) and (6) impose the following $N(d + 1)$ constraints:

$$(\mu_i - \bar{\mu})^\top w = 0 , \quad (14)$$

$$(\Sigma_i - \bar{\Sigma})w = 0_{d \times 1} \quad (15)$$

where $1 \leq i \leq N$. However, note that not all of the constraints are independent. Since at most $\text{rank}(\Sigma_i - \bar{\Sigma})$ equations are independent in (15), the expected number of independent constraints is $N(\ell + 1)$. Conditions (14) and (15) also include $d - m + 1$ dependent constraints since their sums are obviously zeros from the definitions of $\bar{\mu}$ and $\bar{\Sigma}$, i.e. $(\sum_{i=1}^N \mu_i - N\bar{\mu})^\top w = 0$ and $(\sum_{i=1}^N \Sigma_i - N\bar{\Sigma})w = 0_{d \times 1}$. Therefore, the total number of independent constraints is $N(\ell + 1) - (d - m + 1)$ and w has $d - N(\ell + 1) + (d - m + 1)$ degrees of freedom. Since this has to be equal to or less than $m - 1$, the theorem follows. \square

Compared to the requirement of KL-SSA (5), the requirement of ASSA (13) is looser when $\ell \geq 3(1 - \frac{\ell}{d-m})$. When s - and n -sources are uncorrelated, the value of ℓ is close to $d - m$ and therefore this inequality almost always holds. In particular, $N \geq 3$ is sufficient in most cases. This is an important property since the number of available time series blocks would be limited in practice. Also note that if the equality $\ell = d - m$ holds, ASSA requires only $N \geq 2$ time series blocks.

4 Numerical Experiments

4.1 Simulation

We first compared the accuracy of ASSA and KL-SSA on artificial data. As an error measure, we used the subspace angle between the estimated demixing matrix \widehat{W}^\top and A^n defined as $90 - \theta(\widehat{W}^\top, A^n)$ where θ denotes the smallest canonical angle [8]. In the ideal case where the true demixing matrix is estimated, this error gets zero. The artificial data was generated according to the SSA model (1). We used the white noise process [6] for simplicity, where the distribution of s - and n -sources are $\mathcal{N}(0_{m \times 1}, \Sigma^{ss})$ and $\mathcal{N}(\mu_i^n, \Sigma_i^{nn})$ respectively. The parameters Σ^{ss} , μ_i^n and Σ_i^{nn} and mixing matrices A^s and A^n were chosen at random. The correlations between the s - and the n -sources were chosen uniformly at random from the interval $(-1, 1)$. We set the number of s -sources m to 5, the dimensionality of data d to 10 and the number of observations in each block to 100. For KL-SSA, we chose the solution with the smallest objective function value among 20 restarts with random initialization.

The results are shown in Figure 1(a) and Figure 1(b). When the s - and n -sources are uncorrelated, ASSA significantly outperformed KL-SSA. In this case, the error of ASSA almost converged for $N \geq 3$ as implied by Theorem 3. In the case where s - and n -sources are correlated, the assumption underlying ASSA is not valid. Even so, as the number of time series blocks N grows, the error decreased. We conjecture that this is the case because the average correlations converges to zero as the number of time series blocks grows.

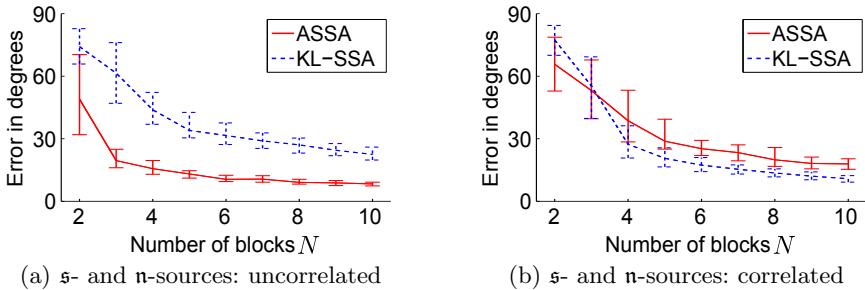


Fig. 1. Median errors of \widehat{W} estimated by ASSA and KL-SSA: The error bars stretch from the 25% to 75% quantile computed over 100 random realizations of observations

4.2 Domain Adaptation

We also compared ASSA and KL-SSA in domain adaptation [9] which is one specific application of SSA into the non-time series scenario [5]. The main objective of Machine Learning is to make accurate predictions on unseen data. Therefore one usually assumes that the characteristics of the system is stable and the current knowledge is directly applicable to new observations. However, if the system

characteristic changes from where the training data was acquired, the prediction gets biased and results in deteriorated generalization performance [10]. In domain adaptation, the task is to reduce the prediction bias and obtain better generalization performance using additional datasets. It can be interpreted as a special case of SSA if we take each dataset as an independent process. We can use SSA to identify the variables in which their distribution remains unchanged across datasets and confine learning and prediction to these variables [45].

For our experiments, we have obtained several binary classification datasets from the LIBSVM [11] repository and added non-stationary effects as follows. First, we randomly divided each dataset into $N + 1$ subsets. Then, we appended non-stationary components (generated as in Section 4.1) according to the SSA model [11]. The first dataset is labeled, while the remaining N datasets are unlabeled. The task is to predict the class labels of the last dataset using the first N -th sets. We set the number of datasets N to 3 and the number of n-sources to 3 and adopted a linear SVM [11] for the classifier.

The results are summarized in Table 1. In the ideal cases where s- and n-sources are uncorrelated, ASSA outperformed KL-SSA. This is because the original signals were well recovered by ASSA as indicated by subspace errors except for b-cancer. In the correlated cases, though the assumption is not valid, ASSA still captured reasonable subspaces and improved the classification performances.

Table 1. Means and standard deviations of the classification error using a linear SVM: The experimental processes were repeated 100 times for each dataset. ‘None’ denotes the result without SSA procedure. For each dataset, the best result and comparable ones based on the t-test at the significance level 5% are indicated by bold font. The right two columns show the median of the subspace errors defined in Section 4.1

(a) s- and n-sources: uncorrelated

Data name	Dim	Classification Error			Subspace Error	
		ASSA	KL-SSA	None	ASSA	KL-SSA
b-cancer	10	0.05±0.03	0.06±0.03	0.05±0.03	86.8	66.0
diabetes	8	0.25±0.03	0.26±0.03	0.27±0.05	14.3	46.1
german	24	0.26±0.03	0.27±0.03	0.28±0.05	21.6	52.0
l-disorders	6	0.39±0.06	0.40±0.06	0.39±0.07	42.7	56.0

(b) s- and n-sources: correlated

Data name	Dim	Classification Error			Subspace Error	
		ASSA	KL-SSA	None	ASSA	KL-SSA
b-cancer	10	0.06±0.04	0.06±0.03	0.09±0.07	73.0	61.8
diabetes	8	0.27±0.04	0.26±0.04	0.28±0.05	37.1	30.1
german	24	0.27±0.03	0.27±0.03	0.30±0.06	29.1	39.3
l-disorders	6	0.37±0.06	0.39±0.05	0.41±0.06	32.7	48.8

5 Conclusion and Future Work

We have proposed a new algorithm for Stationary Subspace Analysis, Analytic SSA (ASSA), which is based on an additional assumption that s - and n -sources are group-wise uncorrelated. Since the solution to ASSA is obtained through a generalized eigenvalue problem, it is numerically attractive. Moreover, we have shown that ASSA outperforms KL-SSA when the additional assumption is fulfilled and still yields good results when this is not the case. In particular, we have observed in our simulations that the effect of correlations between s - and n -sources is counteracted by a large number of time series blocks. Future work will focus on investigating this effect theoretically, in order to obtain an error bound which relates the number of time series blocks to the correlation between stationary and non-stationary sources. Secondly, we conjecture that the eigenvalue spectrum can provide useful guidance for choosing the number of stationary sources.

Acknowledgment

This work was partially supported by JSPS Grant-in-Aid for Scientific Research(B) #22300054. We thank Shohei Shimizu for helpful comments.

References

1. Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P.N., Müller, K.R.: Towards Adaptive Classification for BCI. *Journal of Neural Engineering*, 3, 13–23 (2006)
2. Engle, R.F., Granger, C.W.J.: Co-integration and error correction: representation, estimation, and testing. *Econometrica* 55(2), 251–276 (1987)
3. Cohen, L., Ayrahami-Bakish, G., Last, M., Kandel, A., Kipersztok, O.: Real-time data mining of non-stationary data streams from sensor networks. *Information Fusion* 9(3), 344–353 (2008)
4. von Bünau, P., Meinecke, F., Meinecke, F.C., Kiráry, F.C., Müller, K.R.: Finding Stationary Subspaces in Multivariate Time Series. *Phys. Rev. Lett.* 103(21), 21401 (2009)
5. Meinecke, F.C., von Bünau, P., Kawanabe, M., Müller, K.R.: Learning invariances with Stationary Subspace Analysis. In: *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 87–92 (2009)
6. Hamilton, J.D.: *Time series analysis*. Princeton University Press, Princeton (1994)
7. von Bünau, P., Meinecke, F.C., Kiráry, F.C., Müller, K.R.: Finding Stationary Subspaces in Multivariate Time Series, Supplemental Materials. EPAPPS Document, no.E-PRLTA-103-014948 (EPAPS, <http://www.aip.org/pubservs/epaps.html>)
8. Chatelin, F.: *Eigenvalues of matrices*. John Wiley and Sons, Chichester (1993)
9. Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* 26(1), 101–126 (2006)
10. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90(2), 227–244 (2000)
11. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *Software* (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

A Multi-class Object Classifier Using Boosted Gaussian Mixture Model

Wono Lee and Minho Lee

School of Electronics Engineering, Kyungpook National University,
1370 Sankyu-Dong, Puk-gu, Taegu 702-701, Korea
wolee@ee.knu.ac.kr, mholee@knu.ac.kr

Abstract. We propose a new object classification model, which is applied to a computer-vision-based traffic surveillance system. The main issue in this paper is to recognize various objects on a road such as vehicles, pedestrians and unknown backgrounds. In order to achieve robust classification performance against translation and scale variation of the objects, we propose new C1-like features which modify the conventional C1 features in the Hierarchical MAX model to get the computational efficiency. Also, we develop a new adaptively boosted Gaussian mixture model to build a classifier for multi-class objects recognition in real road environments. Experimental results show the excellence of the proposed model for multi-class object recognition and can be successfully used for constructing a traffic surveillance system.

Keywords: Object classification, Gaussian mixture model, Adaptive boosting, Traffic surveillance system.

1 Introduction

Recently, traffic surveillance technologies have been hot research topics for developing intelligent transportation systems. Among various technologies, camera-based systems such as CCTV monitor are most popular. In the field of researches for traffic surveillance systems with intelligent computer vision technologies, main issues are object detection, recognition and tracking. In this paper, we focus on object recognition problems in a traffic surveillance system.

In previous work, model-based object classification using wire-frame models for video surveillance is proposed [1]. However this approach has disadvantages that wire-frame models must be designed by an external modeling tool and the computational requirement grows linearly with the number of object models. In other study, a patch-based algorithm using a hierarchical feed-forward architecture shows high performance for object recognition [2, 3]. But the algorithm needs much computational load, which makes a difficulty to apply for real time systems. Also, in our previous work, the system using the biologically motivated feature extraction method and support vector machine (SVM) classifiers shows a good performance [4]. A 2-class SVM shows good performance on classification problems but it is not easy to apply for multi-class problems [5]. Depending on the number of class (n), many SVMs are required to classify all of the classes $(n^*(n+1)/2)$. In order to solve multi-class problems, we develop a Gaussian

mixture model (GMM) based classifier. A GMM is a probabilistic model for density estimation, and can be used to do both clustering and categorization [6]. After training a GMM, we can obtain a probability of an object class, and it is used to compare a feature characteristic of an input object with the Gaussian components in the GMM. Then, we can recognize an object if the maximum similarity of a GMM output is over a threshold for a specific trained object class. However, the conventional GMM based classifier just learns an object class with each GMM, it is difficult to generalize for different object classes and inefficient to build the GMs without considering the characteristics of the other object class. In order to overcome those limitations, we adopt the Adaptive boosting (Adaboost) training for constructing the GMMs [7, 8]. Each Gaussian component of GMMs is considered as a weak classifier that has low accuracy. The Adaboost combines the weak classifiers to builds a strong classifier to efficiently recognize the multi-class objects in real traffic environments.

This paper organized as follows; in section 2, we describe the feature extraction method and the proposed boosted GMM classifier. In section 3, experimental results will be shown. At last, summary and conclusion are discussed in section 4.

2 The Proposed Model

The proposed model consists of two main stages. The first is the feature extraction stage from object images. The feature extraction model extracts global and local features from the orientation MAX pooling which is based on the C1 features of the Hierarchical MAX (HMAX) model [9]. The HMAX model proposed by Riesenhuber and Poggio is based on the biological object perception mechanism of the visual cortex of a brain. The C1 feature which is one of the layers in the conventional HMAX model has robust characteristics to scale variation and translation. The second stage is object classification using the extracted features. The classification procedure is performed by probabilities calculation using a GMM. The Expectation-Maximization (EM) algorithm is well known method to update the parameters of a GMM [10, 11]. In this study, we use the greedy learning algorithm of GMM and partial EM searches to create a GMM [12]. After building the GMM for each object class, we can recognize multi-class objects using the probabilities of GMMs. In order to achieve a higher accuracy of classification, the Adaboost algorithm is applied to the GMMs.

2.1 Features Extraction

In the proposed model, we use the C1-like features based on the C1 features. However, we use edge orientation information based on Sobel operator instead of using Garbor filters, which can reduce the computation time. The orientation maps relevant to the S1 units are obtained by calculating of edge orientation as shown in Eq. (1).

$$\begin{aligned} G_x &= HO_n * I, \quad G_y = VO_n * I \\ G &= \sqrt{G_x^2 + G_y^2}, \quad \theta = \arctan(G_y / G_x) \end{aligned} \quad (1)$$

where G is the magnitude of gradient, θ is the direction of gradient, HO is the horizontal operator and VO is the vertical operator, I is intensity of an object image

and n is the size of Sobel operator. While the S1 units have 8 bands, the orientation maps consist of 6 bands, thus 6 sizes of Sobel operators are used (band 1: 3x3, band 2: 5x5, band 3: 7x7, band 4: 9x9, band 5: 11x11, band 6: 13x13). For each band, by taking the maximum value over a 50% overlapped window with cells of different sizes for 2 adjacent bands (band 1 and 2: 8x8, band 3 and 4: 10x10, band 5 and 6: 12x12), the C1-like features consisting of 3 bands are obtained.

Using the similar way of our previous work [4], the proposed model extracts global and local features. Global features are obtained by the modified GIST algorithm [4], however, global features from 3 bands are not combined into 1 feature vector. They are separately used. Each band has 4 directions. Since each band means different scale of an object, this method is more robust to scale variation. Therefore, each global feature is 64 dimensions equally. Local feature is extracted by the same way of our previous work [4], but the parameters are modified to reduce the dimension of local feature. It is reduced from about 320 dimensions to about 70 dimensions because appropriate dimension is depending on training samples.

2.2 Gaussian Mixture Model

A GMM is a popular method for density estimation and clustering. It is defined as a weighted combination of Gaussian distributions. In d -dimensional space, probabilistic density of a Gaussian distribution is defined by Eq. (2).

$$\phi(x; \theta) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right) \quad (2)$$

$$\mu = E[x], \quad \Sigma = E[(x - \mu)^T (x - \mu)]$$

where θ is a model with mean μ and covariance matrix Σ . A GMM with k Gaussian distributions is defined as:

$$f_k(x) = \sum_{i=1}^k \pi_i \phi(x; \theta_j) \quad \text{where} \quad \sum_{i=1}^k \pi_i = 1 \quad (3)$$

where π_i are the mixing weights and have non-negative values.

As a learning algorithm of a GMM, the EM algorithm is generally used. The EM algorithm is known to converge to a locally optimal solution. However, it does not guarantee a globally optimal solution. Therefore we use the greedy learning algorithm with the partial EM searches for an efficient learning [12]. The procedure of the greedy learning algorithm is described as follows:

For the training data $X = \{x_1, x_2, \dots, x_n\}$, the log-likelihood is defined as:

$$L(X, f_k) = \sum_{i=1}^n \log f_k(x_i) \quad (4)$$

1. Compute an optimal one-component GMM f_1 which maximize Eq. (4).
2. Find the optimal new component $\theta^* = \{\mu^*, \Sigma^*\}$ and α^* using Eq. (5)

$$\{\theta^*, \alpha^*\} = \arg \max_{\theta, \alpha} \sum_{i=1}^n \log[(1-\alpha)f_k(x_i) + \alpha\phi(x_i; \theta)] \quad (5)$$

3. Set $f_{k+1}(x) = (1-\alpha^*)f_k(x) + \alpha^*\phi(x; \theta^*)$

4. Update $\{\theta^*, \alpha^*\}$ using the partial EM searches in Eq. (6).

$$p(k+1|x_i) = \frac{\alpha\phi(x_i; \mu_{k+1}, \Sigma_{k+1})}{(1-\alpha)f_k(x_i) + \alpha\phi(x_i; \mu_{k+1}, \Sigma_{k+1})}, \quad \mu_{k+1} = \frac{\sum_{i \in A_j} p(k+1|x_i)x_i}{\sum_{i \in A_j} p(k+1|x_i)} \quad (6)$$

$$\Sigma_{k+1} = \frac{\sum_{i \in A_j} p(k+1|x_i)(x_i - \mu_{k+1})(x_i - \mu_{k+1})^T}{\sum_{i \in A_j} p(k+1|x_i)}, \quad \alpha = \frac{\sum_{i \in A_j} p(k+1|x_i)x_i}{n}$$

where A_j is a subset of X corresponding to a new component.

5. $k \leftarrow k+1$, according to the stopping criterion, stop or repeat from step 2.

2.3 Adaptive Boosting for Gaussian Mixture Models

After training of a GMM for an object, GMM can estimate the probabilities for the inputs. The higher probability means more probable to be involved with the learned class. Since the learning of each GMM is performed for each different class, the GMM by the specific class learning may have poor classification accuracy when the objects in different classes are very similar to each other. In order to solve this problem, we applied the Adaboost algorithm to efficiently collect each GMM for constructing a strong classifier [8]. As a weak learning algorithm for the Adaboost algorithm, we build a simple classifier $h_i(x)$ using a component of GMMs

$$h_i(x) = \begin{cases} 1 & \text{if } \phi(x; \theta_i) \geq h_{th} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where h_{th} is a threshold. The final classification result $H(x)$ is obtained by the weighted sum of weak classifier's results as shown in Eq. (8).

$$H(x) = \sum_{i=1}^k w_i h_i(x) \quad \text{where} \quad \sum_{i=1}^k w_i = 1 \quad (8)$$

And the weights w_i is calculated by Eq. (9)

$$w_i = \log\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right) \quad (9)$$

where ε_i is the error rate of the weak classifier. After learning procedure, the weights are normalized to meet the condition of Eq. (8). The error rate is obtained by both the

positive class trained by the GMM and the negative classes that are not trained by the GMM, thus weak classifiers which can mistake over classes will have lower weights.

2.4 Object Classification for a Traffic Surveillance System

The proposed model is developed for traffic surveillance systems. We build the 3-class object classifier recognizing vehicles, pedestrians and unknown backgrounds. The procedure of classification is shown in Fig. 1. Since the backgrounds do not have specific shapes or structures, we construct clusters of boosted GMM classifiers for 2 classes such as vehicles and pedestrians. Each cluster has 4 boosted GMM classifiers of 3 global feature bands and 1 local feature. Outputs of boosted GMM classifiers, $H_n(x)$, for each object class are summed and the classification result is obtained by the decision module as shown in Eq. (10).

$$\text{Result} = \begin{cases} \text{Vehicle} & \text{if } H_V > H_P \text{ and } H_V > R_{th} \\ \text{Pedestrian} & \text{if } H_V < H_P \text{ and } H_P > R_{th} \\ \text{Background} & \text{otherwise} \end{cases} \quad (10)$$

where R_{th} is a threshold for the decision module. H_V and H_P are the results from the clusters of boosted GMM classifiers for vehicles and pedestrians, respectively.

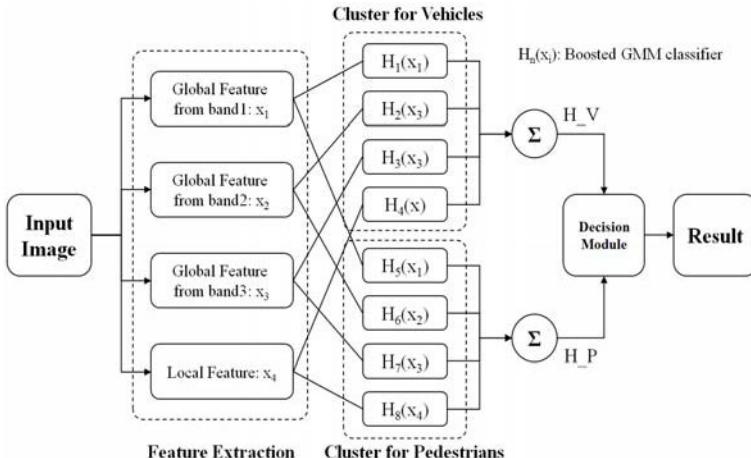


Fig. 1. The procedure of 3-class object classification

3 Experimental Results

In order to evaluate the proposed multi-class object classifier, we use 4 datasets. The dataset from 1 to 3 are shown in table 1 [13-16]. Background images of dataset 1 and 2 are collected in the part of background images of Caltech vehicle database. Dataset 3 is segmented object images from the ABR database that contains a real traffic environment [16]. Fig. 2 shows some of examples of 3 datasets. The train sets for dataset

1 and 2 consist of 900 images including 200 images for each class and 300 images for codebook generation. And the train set for dataset 3 consists of 1000 images including 300 images for vehicle class, 200 images for pedestrian class, 200 images for background and 300 images for codebook generation. For the experiments, we use the same number of training images for each class.

Table 1. The sources of the datasets for experiments

No.	Vehicles	Pedestrians	Background
Dataset 1	CBCL vehicle	CBCL pedestrian	Caltech DB
Dataset 2	Caltech vehicle	Daimler pedestrian	Caltech DB
Dataset 3	ABR DB	ABR DB	ABR DB

Table 2 shows the comparison of experimental results of the proposed model and our previous model [4] for the datasets. The proposed model has similar accuracy for vehicles and pedestrians, but higher performance for background images on average. Moreover, computational speed is much faster than the previous model [4], in which the proposed model takes 64msec on average, while the previous model takes 137 msec for the 78x78 size object images. Fig. 3 shows the averaged receiver operating characteristic (ROC) curve of the proposed model.

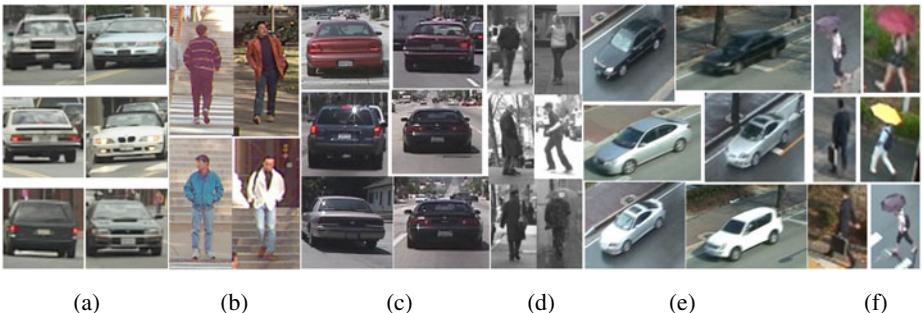


Fig. 2. Examples of datasets for experiments. (a) CBCL vehicles, (b) CBCL pedestrians, (c) Caltech vehicles, (d) Daimler pedestrians, (e) ABR DB vehicles, (f) ABR DB pedestrians.

Table 2. The comparative experimental results of the proposed model and our previous model

		Vehicles	Pedestrians	Background
Dataset 1	Proposed Model	98.5%	99.5%	88%
	Previous Model	98.5%	99%	75.5%
Dataset 2	Proposed Model	97.5%	99%	74.5%
	Previous Model	100%	94.5	79.5%
Dataset 3	Proposed Model	96%	98.5%	72%
	Previous Model	99%	97%	62.5%

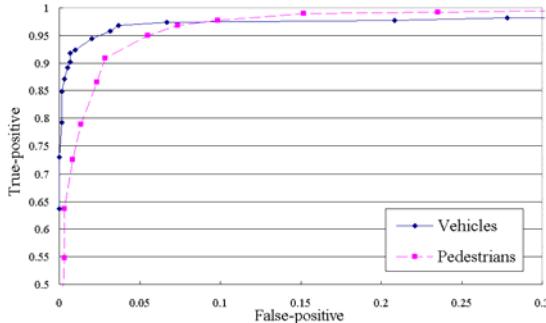


Fig. 3. Averaged ROC curve of the proposed model

Also, in order to compare the proposed model with the previous approach for multi-class object recognition, we conduct additional experiments. We compare the proposed model with the object recognition model by Zisserman et al [17]. For experiments, the dataset 4 consisting of 800 object images for each class (100 for generating the codebook, 350 for training and 350 for test) is used [14]. The proposed model shows higher performance than the Zisserman's model as shown in table 3.

Table 3. The comparative experimental results of the proposed model and the object recognition model by Zisserman et al.

	Motorbikes	Airplanes	Cars (Rear)
Proposed Model	97.7%	94.6%	96%
Zisserman's Model (unscaled)	93.3%	93%	90.3%

4 Conclusion

We proposed a new multi-class object classifier for traffic surveillance systems using boosted GMMs. Owing to the biologically motivated and efficiently modified feature extraction method, the proposed model has not only robustness of translation and scale variation of objects but also reliable computational speed. In our experiments, the classification algorithm based on GMMs with the Adaboost algorithm shows higher performance on multi-class recognition problems. Using the boosted GMM classifier, it is easy to implement a multi-class classifier.

As further works, we need to develop more efficient object detection and tracking algorithm to complete the traffic surveillance system. Also, we plan to develop an incremental object classifier which can learn objects incrementally.

Acknowledgments. This work was supported by the Industrial Strategic Technology Development Program funded by the Ministry of Knowledge Economy (MKE, Korea) (70%) and was also supported by the Converging Research Center Program through the Converging Research Headquarter for Human, Cognition and Environment funded by the Ministry of Education, Science and Technology (2010K001130) (30%).

References

1. Wijnhoven, R., De With, P.H.N.: 3D Wire-frame Object-modeling Experiments for Video Surveillance. In: Proc. of 27th Symp. Inform. Theory in the Benelux, pp. 101–108 (2006)
2. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust Object Recognition with Cortex-like Mechanisms. Trans. on Pattern Analysis and Machine Intelligence (PAMI) 29(3), 411–426 (2007)
3. Wijnhoven, R., De With, P.H.N.: Patch-based Experiments with Object Classification in Video Surveillance. In: Blanc-Talon, J., et al. (eds.) ACIVS 2007. LNCS, vol. 4678, pp. 285–296. Springer, Heidelberg (2007)
4. Woo, J.-W., Lim, Y.-C., Lee, M.: Obstacle Categorization Based on Hybridizing Global and Local Features. In: Chan, J.H. (ed.) ICONIP 2009, Part II. LNCS, vol. 5864, pp. 1–10. Springer, Heidelberg (2009)
5. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. J. Data Min. Knowl. Discov. 2, 121–167 (1998)
6. Scherrer, B.: Gaussian Mixture Model Classifiers (2007)
7. Schapire, R.E.: The Strength of Weak Learnability. Machine Learning 5, 197–227 (1990)
8. Freund, Y., Schapire, R.E.: A Short Introduction to Boosting. The Japanese Society for Artificial Intelligence 14(5), 771–780 (1999)
9. Riesenhuber, M., Poggio, T.: Hierarchical Models of Object Recognition in Cortex. J. Neurosci. 2, 1019–1025 (1999)
10. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. J. of the Royal Statistical Society 39(1), 1–38 (1977)
11. Blimes, J. A.: A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report ICSI-TR-97-021 (1997)
12. Verbeek, J., Vlassis, N., Krose, B.: Efficient Greedy Learning of Gaussian Mixture Models. Neural Computation 15, 469–485 (2003)
13. Center for Biological & Computational Learning, MIT, <http://cbcl.mit.edu>
14. Computational Vision Lab, Caltech, <http://www.vision.caltech.edu>
15. Daimler Pedestrian Detection Benchmark Data Set, <http://www.gavrila.net>
16. Artificial Brain Research Lab, Kyungpook National University, http://abr.knu.ac.kr/ABR_surveillance_DB.html
17. Fergus, R., Perona, P., Zisserman, A.: Object Class Recognition by Unsupervised Scale-Invariant Learning. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 264–271 (2003)

Adaptive Ensemble Based Learning in Non-stationary Environments with Variable Concept Drift

Teo Susnjak, Andre L.C. Barczak, and Ken A. Hawick

Massey University, Albany, New Zealand

{T.Susnjak,A.L.Barczak,k.a.hawick}@massey.ac.nz

Abstract. The aim of this paper is to present an alternative ensemble-based drift learning method that is applicable to cascaded ensemble classifiers. It is a hybrid of *detect-and-retrain* and *constant-update* approaches, thus being equally responsive to both gradual and abrupt concept drifts. It is designed to address the issues of concept forgetting, experienced when altering weights of individual ensembles, as well as real-time adaptability limitations of classifiers that are not always possible with ensemble structure-modifying approaches. The algorithm achieves an effective trade-off between accuracy and speed of adaptations in time-evolving environments with unknown rates of change and is capable of handling large volume data-streams in real-time.

1 Introduction

A considerable body of research has been devoted to developing classification systems whose operating domain has been assumed to be static. Many effective classification systems have already been created for non-evolving environments however their brittleness becomes apparent as unpredictable changes in their application domains occur with time. Classification in an ever increasing number of non-stationary domains such as email spam filters, economic and financial data analysis systems through to computer vision and network monitoring applications is still an unsolved problem for current machine learning approaches.

The unpredictable changes in the underlying data distribution of the operating domain, referred to as *concept drift*, can manifest themselves as gradual or sudden concept drifts, potentially causing radical changes in the class description of a *target concept* Π , resulting in severe degradations in accuracy. While *recurring context drift* may take place, whereby previously active concepts reappear in a randomly cyclical fashion. Data streams may also be large and the speed of adaptability of highest importance, as is often encountered in the field of robotics that employs computer vision.

The challenge is therefore to design learning systems capable of detecting drifts and rapidly adapting to them, while being able to distinguish between actual drift and noise which must not be modeled by the classifier Π . Ideally the concept drift handling algorithm will also be capable of recognizing recurring

contexts, where it will provide a robust balance between *stability* in its capacity to appropriately treat previously learned information and *plasticity* in its approach to integrating currently relevant information [2].

Ensemble based solutions comprise the largest body of approaches due to their ability to retain relevant historical information and their effectiveness at incorporating new knowledge [3]. They generally fall into the category of *constant-update* methods [4] which are not concerned with detecting or quantifying the magnitude of drift, but continuously adapt classifiers to current snapshots of data. Conversely, the *detect-and-retrain* class of strategies actively monitor drifts and update the classifier once a predetermined criteria or performance threshold is surpassed. While the former category of approaches is more amiable to gradual drifts in the environment, the latter is more responsive to sudden drifts [4].

The majority of ensemble based systems realize adaptability by either dynamically updating competence weights of their experts, which limits its handling of recurring contexts and causes unnecessary computational overheads in time-critical domains, or by modifying the ensemble structure by adding additional experts. The latter becomes unfeasible in real-time domains if ensemble training involves massive feature dimensionalities.

We present a novel concept drift handling algorithm that achieves real-time adaptability on large data streams without altering competence weights of ensembles or explicitly training new ones. We achieve this through the idea of learning *layer confidence thresholds* using a combination of *constant-update* and *detect-and-retrain* methods. We demonstrate the effectiveness of our algorithm to handle all drift types of varying magnitudes on a binary-class problem in the domain of computer vision and face detection. Due to the lack of face datasets for drift learning and the shortcomings of many other existing real-world datasets [5], we created our own. We show that initial experimental results in this domain type are promising and warrant further research and expansion of the algorithm.

2 Proposed Framework

We begin by extending the cascade training framework proposed by Barczak et al. [6], which constructs a nested cascade within each layer through independent executions of AdaBoost rounds, predetermined by a maximum number of weak classifiers. We term each independent set of weak classifiers *ensemble-clusters*, whose assignment of weights, together with their combination forms the foundation of our concept drift handling system. The novelty and the effectiveness of our system lies with the development of *layer confidence thresholds*. Interested readers are referred to [7], which presents a detailed implementation of the cascaded ensemble training framework for static datasets used here to construct classifiers as input to our algorithm.

Algorithm 1 details our method. Once an initial nested cascaded classifier has been trained (Step 1), each ensemble-cluster is tested against the same static dataset and competence values are assigned to each one corresponding to their performance accuracy. This competence measure is calculated in the form of *alpha* values used by [8]

Algorithm 1. Concept Drift Learning

Given: SD_n = static dataset vector $\{(x_1, y_1), \dots, (x_n, y_n)\}$ where each sample x corresponds to a class label $y \in \{-1, 1\}$, D_n^t = t^{th} snapshot of non-stationary dataset containing n samples, α_j = vector of sums of confidences $\{z_1, \dots, z_{2^J}\}$ of ensemble-clusters for layer where $z \in \mathbb{R}$, ec = ensemble-cluster classifier, fp = false positive, tp = true positive, ϵ = error rate, $thresh$ = layer threshold value, θ = false positives weight adjustment, ϕ = true positives weight adjustment

Step

- 1 $ec = Train_Classifier(SD_n)$
- $Test_Classifier(SD_n, ec)$ // test ensemble-clusters on static dataset
- 2 **foreach** i^{th} cascade layer **do**
 - foreach** $j^{th} ec_i$ **do**

$$\epsilon_{ij}^{ec} = \frac{1}{2}(fp_{ij} - 1 - tp_{ij})$$
 // calculate error for ensemble-cluster

$$\epsilon_{ij}^{conf} = \frac{1}{2} \ln \frac{1 - \epsilon_{ij}^{ec}}{\epsilon_{ij}^{ec}}$$
 // assign confidence for ensemble-cluster
- 3 **foreach** t^{th} snapshot of dataset stream $D_n^{1, \dots, T}$ **do**
 - $\epsilon^t = ec(D_n^t)$ // calculate error on current dataset
 - if** $\epsilon^t > \epsilon$ **then**
 - $Adjust_Initial_Layer(ec_{ij}(x))$ // error above acceptable threshold
 - if** gradual drift handling effective **then goto step II**
 - foreach** x^{th} sample in dataset stream $D_{1, \dots, n}^t$ **do**
 - foreach** i^{th} cascade layer **do**

$$\alpha_{j++} = \sum_{j=1}^{2^J} ec_{ij}(x)$$
 // insert unique layer alpha value

$$update(thresh_{ij}^{stats})$$
 // update stats for layer alpha value
 - foreach** i^{th} cascade layer **do**
 - foreach** k^{th} distinct $thresh_i$ **do**

$$Calculate_Performance(thresh_{ik}^{stats}, thresh^{tp}, thresh^{fp})$$

$$\epsilon_{ik}^{thresh} = \frac{1}{2}((thresh^{fp} * \theta) - 1 - (thresh^{tp} * \phi))$$

$$\text{if } \epsilon_{ik}^{thresh} < \epsilon^{current_min_thresh} \text{ then make } \epsilon_{ik}^{thresh} \text{ current min.}$$

$$h_{ij}\alpha = 0.5 \ln\left(\frac{1 - \epsilon_{ij}^{error}}{\epsilon_{ij}^{error}}\right)$$

where each confidence value α is assigned to a j^{th} ensemble h on an i^{th} cascade layer. Usually ensemble combination rules consist of a simple *weighted majority vote* as in

$$H_i(x) = sign(\sum_{j=1}^n \alpha_{ij} h_{ij}(x))$$

where H represents a prediction for a layer i using the sum of n number of ensembles applied to on an instance x whose prediction is determined by the *sign*. We modify this by defining a unique minimum confidence threshold value $thresh_i$ for every layer

$$H_i(x) = \begin{cases} 1 & \text{if } \sum_{j=1}^n \alpha_{ij} ec_{ij}(x) > thresh_i \\ 0 & \text{otherwise} \end{cases}$$

in which a sample instance x is positively predicted and passed to succeeding layers for more rigorous testing only if the the sum of confidence values for the current layer surpass it thus producing more robust collective decisions.

In contrast to ensemble-cluster confidence weights, the layer confidence thresholds are computed on the incoming data streams from the application domain in which drift is present. In Step II an optimal layer threshold $thresh_conf_i$ for layer i is computed by first calculating all possible sums of ensemble-clusters. By treating each each distinct sum as a threshold, a generalization error can then be calculated for each one. Additionally, the confidence threshold value can be set to either favour higher hit rates or lower false positive rates by varying the weights of θ and ϕ values in (Step II). Once the algorithm has completed classifying all instances from a current datastream and all layer confidence sums with their respective errors have been computed, then the sum with the lowest error rate for each layer is selected as the optimal threshold (Step II).

Once threshold learning is finished, the classifier is ready to be redeployed and begin handling drift (Step II). Unlike most ensemble based methods, our algorithm makes use of a trigger mechanism to inform it that drift is occurring in the environment. In its current form, our algorithm uses the classification error rate as a trigger for drift handling to begin if the generalization ability falls beyond a predefined level (Step II).

Initial experiments have shown that employing layer confidence thresholds achieves an aggressive strategy for eliminating false positive detections that can sometimes also reduce positive hit rates if not applied optimally. We can combine this observation with the flexibility of cascaded classifiers which allows us to use a strategy that varies the number of layers that utilize confidence thresholds depending on its current generalization. During runtime the drift learning algorithm can progressively increase the number of layers to which layer thresholds are applied (Step II) until a sufficient number of false positives has been eliminated for practical usage while preceding layers are calculated in original form. The ability of the framework to progressive increase the number of layers that use confidence thresholds becomes the algorithm's facility for handling gradual drifts.

As data streams begin to drift more acutely, the algorithm activates proportionally larger numbers of layer confidences in respect to increasing false positive rates. This proceeds until the drift stabilizes or until all available layers with confidence thresholds are exhausted. If all layer thresholds have been deployed and the error rate is still above an acceptable level, then the layer thresholds have been rendered irrelevant to current conditions and subsequently optimal layer threshold learning is re-initiated (Step II).

Work by [3] which uses the Learn++.NSE algorithm and its variants, bear closest resemblance to our method. Their solution constructs *composite hypotheses* which are similar to our idea of ensemble-clusters. Our algorithm differs in that it does not alter the weights of individual ensemble-clusters as new datasets

become available, but instead focuses primarily on leveraging the cascaded structure to formulate optimal confidence layer thresholds. Also in contrast to our method, [3] train new composite hypotheses for each incoming dataset and augment them to the existing set. Our algorithm does not explicitly train new ensembles to incorporate into its structure. For many problems this is currently not feasible in real-time. Therefore, the strength and novelty of our method lies in its rapid adaptability to concept drifts in the negative samples making it highly applicable to rare-event detection domains where high sensitivity to false positive detections is required.

The domain of adaptability to concept drift in vision detection has received a limited amount of work and is currently an emerging area of research. [9] propose sequential learning algorithms for cascaded face detectors through a strategy of continuous updates of ensemble weights. However, this approach comes with varying degrees of target concept forgetting [4] and the challenge of handling recurring contexts. While [10] approach the problem of concept drift by developing a hybrid tracker system.

3 Experiment Design

We tested our algorithm on a collection of images containing faces and compared the results against the identical classifier without concept learning capabilities. The underlying classifier was first trained offline using the [7] method with Haar-like features, on 5000 positive faces, collected from FERET and Yale *Face Database B* [11] datasets, against a negative dataset of 2000 samples, bootstrapped from a pool of 2500 negative images. The training and test images used for concept learning were completely independent. The resulting classifier comprised of 24 layers with 3-6 ensemble-clusters per layer.

The test dataset consisted of 100 images with 640x480 pixel dimensions, created using a web cam from varying environments. The dataset was compiled with instances of gradual and abrupt shifts as well as recurring contexts and data distribution changes over time. The nature of the non-stationary environment can be seen in the image examples from our test dataset in Figure [1].

Face coordinates were extracted from images using markers. The initial frame in the test dataset was programmed to trigger threshold learning and thereafter all learning was conducted automatically by the algorithm without explicit intervention. Concept learning of layer thresholds occurred on a single frame only. Each frame provided 336,980 training samples in the form of image sub-windows. Initial subwindow size was 24x24 pixels and increased by a factor of 1.2 after exhaustive raster scanning of an image at an increment of 2 pixels per scan.

Due to the rare-event operating domain of face detectors, higher weights were assigned to positive samples during layer threshold learning. The purpose of this was to preserve high hit rates, however in images where faces were absent, the algorithm was equally capable of learning only on negatives. The algorithm initiated gradual drift adaptability each time false positives were detected. Concept learning for abrupt shifts was initiated if a false negative detection occurred or if

false positives occurred on the lowest cascade layer set for calculating thresholds. In our experiments using a classifier with 24 layers, the eighth layer was set as the lowest for layer threshold learning.

4 Results

By first examining the total false positive detections per frame, we see in Figure 2a that the false positive detections for the base classifier seldom attained zero while the upper range exceeded 50 false detections per frame. Our drift learning algorithm removed false detections for all but 18 frames while their total per frame never exceeded two false positive detections.

The effectiveness of layer threshold learning can be seen in Figure 2b in which the reduction of the false positives in relation to the calculation of each successive cascade layer can be seen. The concept learning algorithm demonstrates its ability to rapidly remove false positive detections at each succeeding layer. The total false positive count over 100 test images for our algorithm was 33 and confined to 18 images, as opposed to a total of 2295 false positives for the base classifier. The base classifier however correctly classified all positive samples while our drift handling algorithm produced three false negative detections.

We can see the pattern of our algorithm's adaptation to gradual concept drifts per frame together with the locations of triggers which initiated re-learning of layer thresholds in Figure 3a. From it we see the manner in which an increasing number of layer thresholds are taken into consideration as accuracy decreases and is complemented by Figure 2a which shows the magnitude of false positive detections which initiate the gradual concept-drift-handling facility.

Figure 3a shows also that between frames 55-75, a high concentration of abrupt concept drifts took place. This caused higher error rates to occur, resulting in the initiation of four layer threshold learning phases in short succession. Despite this, the algorithm demonstrated robustness in its responsiveness to both frequent sudden drifts as well as slow variations in data.

The runtimes for the concept learning algorithm in comparison to the base classifier can be seen in Figure 3b. The computational time for learning layer



Fig. 1. Sample test image sequence with various forms of drift. Row a) Example of gradual drifts becoming abrupt due to illumination changes in an indoor environment succeeded by sudden drifts into an outdoor setting, containing both data distribution changes through absences of faces and gradual changes modelled by camera panning into Row b) concluding with a sudden change to indoors with a combination of the above drifts.

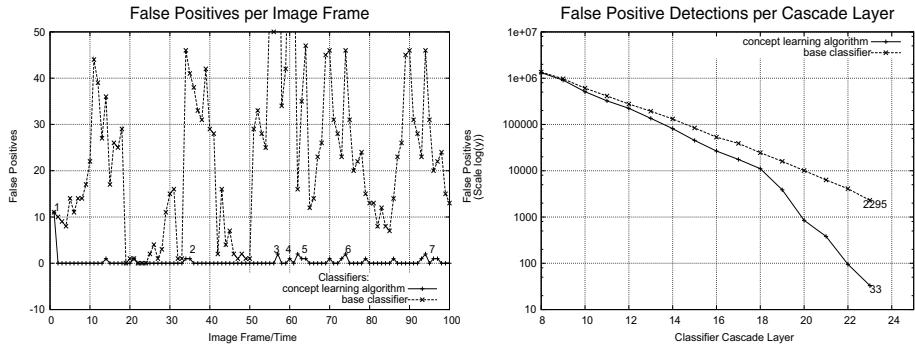


Fig. 2. a) Total number of false positive detections per frame. Positions of the learning phases from 1-7. b) Number of false positive detections in respect to cascade layers with the total sum of false positive detections emphasized for both classifiers.

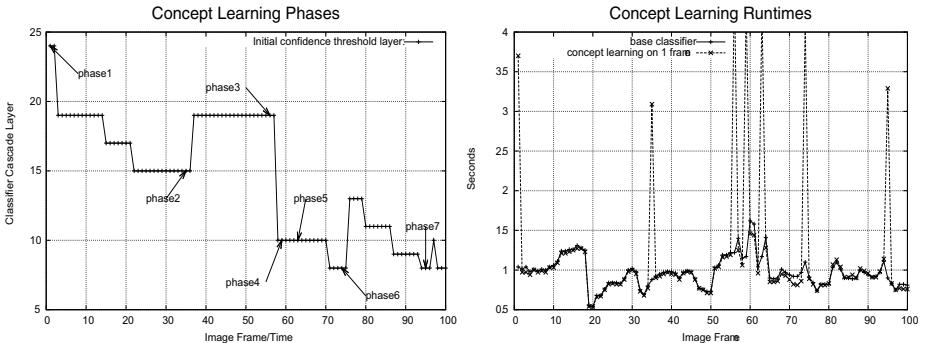


Fig. 3. Figure a) The pattern of the learning algorithm's adaptation to gradual drifts for each successive frame. As detection difficulty increases, the algorithm increases the number of layers for which threshold confidences are used by selecting lower layers as initial starting points. b) Comparison of classifier runtimes and learning phases.

thresholds ranges from between 3-4 times that of the normal classification time and for most applications is likely to be an acceptable performance penalty given a considerable improvement in accuracy.

5 Conclusions

The main contribution of this paper is the introduction of a concept learning algorithm for non-stationary environments, designed specifically for cascade classifiers that consist of clusters of ensembles. Its novelty lies in learning optimal cascade layer thresholds in order to achieve adaptability instead of updating ensemble confidence weights themselves.

Experiments demonstrated our system's capacity to achieve real-time adaptability for problem domains where explicit training of new ensembles is not feasible due time constraints imposed by large data streams and immense feature dimensionalities. We applied our algorithm to face detection and found that it was effective and timely at handling diverse forms of concept drifts without concept forgetting, while maintaining both sustainable plasticity and stability over time.

Future work will focus on applying this algorithm to other domains as well as extending its learning capabilities in a way that will allow it to integrate novel positive sample information into the classifier.

References

1. Kuncheva, L.: Classifier ensembles for changing environments. *Multiple Classifier Systems*, 1–15 (2004)
2. Grossberg, S.: Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks* 1(1), 17–61 (1988)
3. Elwell, R., Polikar, R.: Incremental Learning of Variable Rate Concept Drift. *Multiple Classifier Systems*, 142–151 (2009)
4. Rodriguez, J., Kuncheva, L.: Combining online classification approaches for changing environments. *Structural, Syntactic, and Statistical Pattern Recognition*, 520–529 (2010)
5. Tsymbal, A.: The problem of concept drift: definitions and related work. In: TCD-CS-2004-15, Dep. of Comp. Sci., Trinity College, Dublin, Ireland, vol. 4 (2004)
6. Barczak, A.L.C., Johnson, M.J., Messom, C.H.: Empirical evaluation of a new structure for adaboost. In: SAC 2008: Proceedings of the 2008 ACM Symposium on Applied Computing, pp. 1764–1765. ACM, Fortaleza (2008)
7. Susnjak, T., Barczak, A., Hawick, K.: A modular approach to training cascades of boosted ensembles. In: Hancock, E.R., et al. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 640–649. Springer, Heidelberg (2010)
8. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
9. Huang, C., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Incremental learning of boosted face detector. In: ICCV, pp. 1–8. IEEE, Los Alamitos (2007)
10. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Computer Vision–ECCV, pp. 234–247 (2008)
11. Georghiades, A., Belhumeur, P., Kriegman, D.: 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)

High Dimensional Non-linear Modeling with Bayesian Mixture of CCA

Tikara Hosino

Nihon Unisys, Ltd.

Abstract. In the high dimensional regression and classification, we often need the feature selection and the dimensionality reduction to cope with the huge computational cost and the over-fitting of the parameters. Canonical Correlation Analysis (CCA) and their hierarchical extension (includes Bayesian method) was proposed for this purpose. However, the real data set often violates the assumption of the linearity of CCA. Thus, we need the non-linear extension of them. To solve this problem, we propose the Bayesian mixture of CCA and give the efficient inference algorithm by Gibbs sampling. We show that the proposed method is the scalable natural extension of CCA and RBF type neural networks for the high dimensional non-linear problems.

1 Introduction

Many machine learning tasks break into the regression or the classification problems. In regression and classification, the statistical methods (especially Bayesian method) show the effectiveness at the point of theory and the practices. However, the real data set, such as natural language processing, bioinformatics and neural information processing, often has the high dimensional feature vector and the non-linearity of the target process. In these case, the direct application of Bayesian method is suffered from huge computational cost and over-fitting by the scarcity of the samples. In ordinary process, the feature selection and the dimensionality reduction are used for these problems.

Especially, Canonical Correlation Analysis (CCA) and their hierarchical extension is widely used. CCA has the nice property which belongs to the exponential family with the hidden variables. Thus, the inference algorithm such as Gibbs sampling and the variational Bayesian method is exactly derived. However, CCA assume the linearity of the process, if the target is nonlinear then we need some extension of CCA. To overcome the difficulty, we propose the Bayesian mixture of CCA and derive the efficient inference algorithm by Gibbs sampling. We show that the proposed method is the scalable natural extension of CCA and RBF type neural networks (Normalized Gaussian Networks [8][9]) against the high dimensional non-linear processes. Finally, we execute the numerical experiments which show the effectiveness of the proposed method.

2 Mixture of Canonical Correlation Analysis

In this section, we define the mixture of CCA which is the proposed method of the paper. As the proposed method is the extension of the ordinal CCA, we firstly describe CCA, then, we extend the CCA to the mixture one.

2.1 Canonical Correlation Analysis

CCA is the model which the given two multivariate vectors $y \in R^D$, $x \in R^M$ are caused by common low dimensional hidden vector $z \in R^L$ which ordinary satisfies $L \ll \max(D, M)$. The relation of the observation and the hidden variables are assumed linear and additional gaussian noise which is written by the form

$$p(y, x, z|\theta) = N(y; \mu_y + V_A A z, \Sigma_y^{-1}) N(x; \mu_x + V_B B z, \Sigma_x^{-1}) N(z; 0, I_L) \quad (1)$$

where $\theta = \{\mu_y, V_A, A, \Sigma_y, \mu_x, V_B, B, \Sigma_x\}$ are model parameters. Additionally, $V_a, \Sigma_y, V_b, \Sigma_x$ is constrained to diagonal matrix. $N(x; \mu, \Sigma^{-1})$ is multivariate gaussian which is given by

$$N(x; \mu, \Sigma^{-1}) = \frac{|\Sigma|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}} \exp(-\frac{1}{2}(x - \mu)^T \Sigma (x - \mu)). \quad (2)$$

We divide the regression weight parameter into the V and A . This separation is proposed by [2], for achieving the column wise and row wise sparsity with Automatic Relevance Determination (ARD) priors [1]. Then, we can integrate out the hidden variable z .

$$\begin{aligned} p(y, x|\theta) &= N(X; \mu, \Sigma^{-1} + WW^T) \\ X &= \begin{pmatrix} y \\ x \end{pmatrix}, \quad \mu = \begin{pmatrix} \mu_y \\ \mu_x \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_y & \\ & \Sigma_x \end{pmatrix}, \quad W = \begin{pmatrix} V_A A \\ V_B B \end{pmatrix} \end{aligned} \quad (3)$$

Additionally, for the regression problem, the distribution of y given x is derived by Bayes rule

$$p(y|x, \theta) = N(y; \mu_y + V_A A \hat{z}, \Sigma_y^{-1} + V_A A \hat{\Sigma}_z A^T V_A^T) \quad (4)$$

$$\hat{\Sigma}_z = (B^t V_B^T \Sigma_x V_B B + I_L)^{-1}, \quad \hat{z} = \hat{\Sigma}_z (B^t V_B^T \Sigma_x (x - \mu_x)). \quad (5)$$

It is noted that the efficiency of CCA is originated from the matrix inversion lemma which is given by

$$(A + XBX^T)^{-1} = A^{-1} - A^{-1}X(B^{-1} + X^T A^{-1}X)^{-1}X^T A^{-1} \quad (6)$$

$$|A + XBX^T| = |B||A||B^{-1} + X^T A^{-1}X|. \quad (7)$$

Using this lemma reduces the matrix inversion computation such as $(\Sigma^{-1} + WW^T)^{-1}$ from $O((D + M)^3)$ to $O(L^3)$.

2.2 Mixture of CCA

Next, we extend CCA to the mixture case. For the non-linear problem, several mixture extensions were proposed for factor analysis [6][7]. Let the number of mixture of CCA to K . Then, the complete likelihood is written by

$$p(y, x, z, l|\theta) = \prod_{k=1}^K \{a_k N(y; \mu_y^{(k)} + V_A^{(k)} A^{(k)} z, \Sigma_y^{(k)-1}) \\ N(x; \mu_x^{(k)} + V_B^{(k)} B^{(k)} z, \Sigma_x^{(k)-1}) N(z; 0, I_L)\}^{l_k} \quad (8)$$

where l_k is hidden variable which satisfies $l_k = \{0, 1\}$, $\sum_{k=1}^K l_k = 1$ and is multinomial distribution having parameter a_k . We can integrate out the hidden variable z and sum out the hidden variable l .

$$p(y, x|\theta) = \sum_{k=1}^K a_k N(X; \mu^{(k)}, \Sigma^{(k)-1} + W^{(k)} W^{(k)T}) \\ X = \begin{pmatrix} y \\ x \end{pmatrix}, \mu^{(k)} = \begin{pmatrix} \mu_y^{(k)} \\ \mu_x^{(k)} \end{pmatrix}, \Sigma^{(k)} = \begin{pmatrix} \Sigma_y^{(k)} & \\ & \Sigma_x^{(k)} \end{pmatrix}, W^{(k)} = \begin{pmatrix} V_A^{(k)} A^{(k)} \\ V_B^{(k)} B^{(k)} \end{pmatrix}$$

Additionally, for the prediction task, the distribution of y given x is written by

$$p(y|x, \theta) = \sum_{k=1}^K \frac{a_k N(x; \mu_x^{(k)}, \Sigma_x^{(k)-1} + V_B^{(k)} B^{(k)T} V_B^{(k)T})}{\sum_{j=1}^K a_j N(x; \mu_x^{(j)}, \Sigma_x^{(j)-1} + V_B^{(j)} B^{(j)T} V_B^{(j)T})} \\ N(y; V_A^{(k)} A^{(k)} \hat{z}^{(k)} + \mu_y^{(k)}, \Sigma_y^{(k)-1} + V_A^{(k)} A^{(k)} \hat{\Sigma}^{(k)} A^{(k)T} V_A^{(k)T}) \quad (9)$$

$$\hat{\Sigma}^{(k)} = (B^{(k)T} V_B^{(k)T} \Sigma_x^{(k)} V_B^{(k)} B^{(k)} + I_L)^{-1} \quad (10)$$

$$\hat{z}^{(k)} = \hat{\Sigma}^{(k)} (B^{(k)T} V_B^{(k)T} \Sigma_x^{(k)} (x - \mu_x^{(k)})). \quad (11)$$

The equation (9) is correspondent to RBF type neural networks (Normalized Gaussian network[8][9]) where $\hat{z}^{(k)} \in R^L$ matches the input vector $x \in R^M$. By the same argument as CCA, we can manage the scalability of the inference from $O((D + M)^3)$ to $O(L^3)$ by using the matrix inversion lemma.

3 Inference Algorithm

We derive the parameter inference algorithm using Gibbs sampling. Let the N observable samples are $\{(y_1, x_1), \dots, (y_N, x_N)\}$. Additionally we can assume that the prior distribution of the parameters are the conjugate of the likelihoods. Then, Gibbs sampling of the proposed method is given by iterating the following three steps.

- Sample the hidden component l given y, x, θ
- Sample the hidden vector z given y, x, l, θ
- Sample the parameters θ given complete data y, x, l, z

3.1 Inference of the Component

First, we sample the hidden component index l . The conditional probability of the component given x, y, θ is written by

$$p(l_{nk}|x, y, \theta) = \frac{a_k N(X_n; \mu^{(k)}, \Sigma^{(k)-1} + W^{(k)}W^{(k)T})}{\sum_{j=1}^K a_j N(X_n; \mu^{(j)}, \Sigma^{(j)-1} + W^{(j)}W^{(j)T})}$$

Once the component index l is sampled. We can gather the sufficient statistics for the mixture parameter a_k . The prior distribution for a_k is given by Dirichlet distribution

$$p(a|\phi_0) = \frac{\Gamma(K\phi_0)}{\Gamma(\phi_0)^K} \prod_{k=1}^K a_k^{\phi_0-1}. \quad (12)$$

Then, the posterior distribution for a is written by

$$p(a|\phi_0, x, y) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1) \dots \Gamma(\alpha_K)} \prod_{k=1}^K a_k^{\alpha_k-1}, \quad (13)$$

where

$$\alpha_k = \phi_0 + n_k, \quad \alpha_0 = \sum_{k=1}^K \alpha_k, \quad n_k = \sum_{n=1}^N l_{nk}. \quad (14)$$

3.2 Inference of the Hidden Vector

Second, we sample the hidden vector z . The conditional probability of the hidden vector given x, y, l, θ is written by

$$p(z_n|y_n, x_n, l_n, \theta) = N(z_n; \hat{\mu}_z, \hat{\Sigma}_z) \quad (15)$$

$$\hat{\Sigma}_z = (I_L + W^{(k)T} \Sigma^{(k)} W^{(k)})^{-1}, \quad \hat{\mu}_z = \hat{\Sigma}_z (W^{(k)T} \Sigma^{(k)} (X_n - \mu^{(k)})) \quad (16)$$

3.3 Inference Algorithm for Each Component

Finally, we sample the parameter θ given complete data x, y, z, l . We only show the sampling procedure V_A, A, μ_y, Σ_y . Similar formula holds on V_B, B, μ_z, Σ_z . The prior distributions for V_A, A are given by

$$p(V_A^{(k)}|\alpha_v) = \prod_{i=1}^D \left(\frac{\alpha_{v_{ai}}}{2\pi} \right)^{\frac{1}{2}} \exp\left(-\frac{1}{2}\alpha_{v_{ai}} v_{ai}^{(k)2}\right) \quad (17)$$

$$p(A^{(k)}|\alpha_a) = \prod_{i=1}^L \left(\frac{\alpha_{a_i}}{2\pi} \right)^{\frac{D}{2}} \exp\left(-\frac{1}{2}\alpha_{a_i} \|\tilde{a}_i^{(k)}\|^2\right) \quad (18)$$

where a_i indicates i th row of A and \tilde{a}_i indicates i th column of A .

Then, the posterior distributions for V_A , A is given by

$$p(V_A^{(k)}|\theta_{\sim}) = \prod_{i=1}^D N(v_{ai}^{(k)}; m_{v_{ai}}^{(k)}, \sigma_{v_{ai}}^{(k)}), \quad p(A^{(k)}|\theta_{\sim}) = \prod_{i=1}^D N(a_i^{(k)}; m_{a_i}^{(k)}, \Sigma_{a_i}^{(k)}) \quad (19)$$

where

$$\begin{aligned} m_{v_{ai}}^{(k)} &= \sigma_{v_{ai}}^{(k)} \Sigma_{yi}^{(k)} a_i^{(k)T} \sum_{n=1}^N l_{nk} z_n^{(k)} (y_{ni} - \mu_{yi}^{(k)}) \\ \sigma_{v_{ai}}^{(k)} &= (\Sigma_{yi}^{(k)} \text{trace}(a_i^{(k)} a_i^{(k)T} \sum_{n=1}^N l_{nk} z_n^{(k)} z_n^{(k)T}) + \alpha_{v_{ai}}^{(k)})^{-1} \\ m_{a_i}^{(k)} &= \Sigma_{a_i}^{(k)} \Sigma_{yi}^{(k)} v_{ai}^{(k)} \sum_{n=1}^N l_{nk} z_n^{(k)} (y_{ni} - \mu_{yi}^{(k)}) \\ \Sigma_{a_i}^{(k)} &= (\Sigma_{yi}^{(k)} v_{ai}^{(k)2} \sum_{n=1}^N l_{nk} z_n^{(k)} z_n^{(k)T} + \text{diag } \alpha_a^{(k)})^{-1}. \end{aligned} \quad (20) \quad (21)$$

The prior distributions for α_v , α_a is given by

$$p(\alpha_v^{(k)}) = \prod_{i=1}^D Ga(\alpha_{vi}^{(k)}; a_{\alpha v 0}, b_{\alpha v 0}), \quad p(\alpha_a^{(k)}) = \prod_{i=1}^L Ga(\alpha_{a_i}; a_{\alpha a 0}, b_{\alpha a 0}) \quad (22)$$

where $Ga(\theta; a, b) \propto \theta^{a-1} \exp(-b\theta)$ is gamma distribution. Then, the posterior distributions for α_v , α_a given by

$$p(\alpha_v^{(k)}|\theta_{\sim}) = \prod_{i=1}^D Ga(\alpha_{v_{ai}}^{(k)}; a_{\alpha v}^{(k)}, b_{\alpha v}^{(k)}), \quad p(\alpha_A^{(k)}|\theta_{\sim}) = \prod_{i=1}^L Ga(\alpha_{a_i}^{(k)}; a_{\alpha a}^{(k)}, b_{\alpha a}^{(k)}) \quad (23)$$

where

$$a_{\alpha_v}^{(k)} = a_{\alpha v 0} + \frac{1}{2}, \quad b_{\alpha_v}^{(k)} = b_{\alpha v 0} + \frac{v_{ai}^{(k)2}}{2}, \quad a_{\alpha_a}^{(k)} = a_{\alpha a 0} + \frac{D}{2}, \quad b_{\alpha_a}^{(k)} = b_{\alpha a 0} + \frac{\|\tilde{a}_i^{(k)}\|^2}{2}.$$

The prior distributions for μ_y , Σ_y are given by

$$p(\Sigma_{yj}^{(k)}) = Ga(\Sigma_{yj}^{(k)}; \frac{b_0}{2}, \frac{c_0}{2}), \quad p(\mu_{yj}^{(k)}) = N(\mu_{yj}^{(k)}; 0, \xi_0^{-1})$$

Then, the posterior distributions for μ_{yj} , Σ_{yj} are given by

$$p(\Sigma_{yj}^{(k)}|\theta_{\sim}) = Ga(\Sigma_{yj}^{(k)}; \frac{b_j^{(k)}}{2}, \frac{c_j^{(k)}}{2}), \quad p(\mu_{yj}^{(k)}|\theta_{\sim}) = N(\mu_{yj}^{(k)}; \gamma_{yj}^{(k)}, \xi_j^{(k)-1})$$

where θ_{\setminus} is the set of parameters excludes left hand side, $n_k = \sum_{n=1}^N l_{nk}$ and

$$b_j^{(k)} = b_0 + n_k, c_j^{(k)} = c_0 + \sum_{n=1}^N l_{nk} (y_{nj} - v_{aj}^{(k)T} a_j^{(k)} z_n^{(k)} - \mu_{yj})^2 \quad (24)$$

$$\gamma_{yj}^{(k)} = \xi_j^{(k)-1} \Sigma_{yj}^{(k)} \sum_{n=1}^N l_{nk} (y_{nj} - v_{aj}^{(k)T} a_j^{(k)} z_n^{(k)}), \xi_j^{(k)} = \xi_0 + n_k \Sigma_{yj}^{(k)} \quad (25)$$

4 Classification Case

In the classification case, we use multinomial probit model. If the class label D is large, then the estimation of the output covariance matrix $O(D^3)$ becomes bottleneck. Same as the regression case, we constrain the covariance matrix to the diagonal. Moreover, to escape the identification problem, we set $\Sigma_{y11} = 1$.

Let the observe variable y ($y = \{0, 1\}$, $\sum_{j=0}^{D-1} y_j = 1$) and the corresponding hidden variables $\hat{y} \in R^{D-1}$. The hidden variable \hat{y} given y, x, z, l is distributed by truncated normal

$$p(\hat{y}_j | \hat{y}_{\setminus}, y, x, z, l) = TN(\hat{y}_j; V_{Aj}^{(k)} A_j^{(k)} z^{(k)} + \mu_{yj}^{(k)}, \Sigma_{yj}^{(k)-1}) \quad (26)$$

with truncation

$$\begin{cases} \hat{y}_j < 0 & \text{if } y_0 = 1 \\ \hat{y}_j > \max\{\max_{h \neq j}(\hat{y}_h), 0\} & \text{if } y_j = 1, y_0 \neq 1 \\ \hat{y}_j < \max_{h \neq j}(\hat{y}_h) & \text{if } y_j \neq 1, y_0 \neq 1. \end{cases} \quad (27)$$

Once \hat{y} is given, then we can return the inference of the regression case.

5 Experiments

We execute the numerical experiments to show the effectiveness of the proposed method. In this paper, we focus the non-linear property of the proposed method. Therefore, we compare the linear CCA to the proposed method. The concrete learning procedure is as follows. Firstly, we apply 1000 cycles of Gibbs sampling steps for burn-in period. Then, we gather the parameters of subsequent 200 sampling steps for the posterior distribution. Using these posterior, we predict the test set and measure the accuracy by mean square error (MSE). The number of mixture k is fixed at 3 through all experiments.

5.1 Dataset

We use the three dataset for the experiments. All datasets is downloaded from LIBSVM Data: Regression [1]. For all dataset, we divide the samples to learning and test set. Number of learning samples, test samples, dimension for each dataset are shown by table [2]

¹ <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html>

Table 1. Datasets for Experiments

Dataset	Learning	Tests	Dimension
cpusmall	7500	692	12
abalon	4000	177	8
housing	450	56	13

5.2 Result

The result of the experiments is summarized as table 2. In sampling step, if the number of the assignment to the component n_k is 0, then we regard the component is irrelevant. The improvement is measured at the $\frac{\text{Linear CCA}}{\text{proposed}} - 1$.

Table 2. MSE of Methods

Dataset	Relevant mixture	Linear CCA	Proposed	Improvement
cpusmall	3	5.02	2.66	0.88
abalon	2	5.29	4.7	0.12
housing	1	3.78	3.88	-0.02

5.3 Discussion

As shown by the result, the non-linear effect is improve the performance in cpusmall and abalon dataset. Moreover, we show that the strength of non-linearity, which is indicated by the relevant mixtures, is proportional to the improvement.

6 Conclusion

We proposed Bayesian mixture of CCA for the high dimensional non-linear statistical modeling. Additionally, we derive the efficient Gibbs sampling algorithm for the proposed method. The proposed method effectively achieves the feature selection and the dimensionality reduction via ARD, and cope with the non-linearity by the mixture distribution. In the numerical experiments of the real world datasets, we show that the non-linear extension improves the performance. The model selection using WAIC [3] and non-parametric setting via Dirichlet process [4] and Indian buffet process [5] are the future works.

Acknowledgments. This research is supported by Promotion program for Reducing global Environmental load through ICT innovation (PREDICT), Ministry of Internal Affairs and Communications of Japan.

References

1. Bishop, C.M.: Bayesian PCA. Advances in Neural Information. In: Processing Systems, vol. 11, pp. 382–388 (1999)
2. Katahira, K., Matsumoto, N., Sugase-Miyamoto, Y., Okanoya, K., Okada, M.: Doubly Sparse Factor Models for Unifying Feature Transformation and Feature Selection. Journal of Physics: Conference Series (in press)
3. Watanabe, S.: Equations of states in singular statistical estimation. Neural Networks 23, 1 (2010)
4. Ferguson, T.S.: A Bayesian analysis of some nonparametric problems. Annals of Statistics 1, 209–230 (1973)
5. Griffiths, T., Ghahramani, Z.: Infinite latent feature models and the Indian buffet process. In: Advances in Neural Information Processing System (NIPS), vol. 18 (2005)
6. Ghahramani, Z., Hinton, G.E.: The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Dept. of Comp. Sci., Univ. of Toronto (1996)
7. Ghahramani, Z., Beal, M.J.: Variational Inference for Bayesian Mixtures of Factor Analyzers. In: Advances in Neural Information Processing System (NIPS), vol. 12 (2000)
8. Xu, L., Jordan, M.I., Hinton, G.E.: An Alternative Model for Mixtures of Experts. In: Advances in Neural Information Processing System (NIPS), vol. 7 (1995)
9. Sato, M., Ishii, S.: On-line EM Algorithm for the Normalized Gaussian Network. Neural Computation 12(2), 407–432 (2000)

The Iso-regularization Descent Algorithm for the LASSO

Manuel Loth and Philippe Preux

Université de Lille,
INRIA Lille - Nord Europe,
LIFL (UMR CNRS 8022)

Abstract. Following the introduction by Tibshirani of the LASSO technique for feature selection in regression, two algorithms were proposed by Osborne et al. for solving the associated problem. One is an homotopy method that gained popularity as the LASSO modification of the LARS algorithm. The other is a finite-step descent method that follows a path on the constraint polytope, and seems to have been largely ignored. One of the reason may be that it solves the constrained formulation of the LASSO, as opposed to the more practical regularized formulation. We give here an adaptation of this algorithm that solves the regularized problem, has a simpler formulation, and outperforms state-of-the-art algorithms in terms of speed.

Keywords: Lasso, algorithm, descent, regularization.

1 Introduction

The Least Absolute Selection and Shrinkage Operator (LASSO) was proposed in [1] as an efficient and feasible way to produce sparse linear models in regression. Let us begin by recalling its definition and fixing notations.

Let \mathbf{v}' , \mathbf{M}' denote the transpose of a vector \mathbf{v} or matrix \mathbf{M} , $\mathbf{u}'\mathbf{v}$ the scalar product of \mathbf{u} and \mathbf{v} , $\|\mathbf{v}\|_2^2 = \sum_{i=1}^{\dim(\mathbf{v})} v_i^2 = \mathbf{v}'\mathbf{v}$ the squared ℓ^2 norm, and $\|\mathbf{v}\|_1 = \sum_{i=1}^{\dim(\mathbf{v})} |v_i| = \text{sign}(\mathbf{v})'\mathbf{v}$ the ℓ^1 -norm, with $\text{sign}(\mathbf{v}) = (\text{sign}(v_1), \dots)'$. Let B^A denote the set of all functions from A to B .

Problem statement. *Given*

- n samples $(x_i, y_i)_{1 \leq i \leq n}$ from joint variables $X \in \mathcal{X}$ and $Y \in \mathbb{R}$,
- a set of feature functions $\mathcal{D} \subset \mathbb{R}^{\mathcal{X}}$,
- a constraint $t \in \mathbb{R}_+$,

the LASSO associates to the samples a model $Y = f(X)$ defined as a linear combination of features from \mathcal{D} that minimizes the squared residual subject to an ℓ^1 -norm constraint on the linear coefficient vector:

$$\min_{(\sigma, \beta) \in 2^{\mathcal{D}} \times \mathbb{R}} \|\mathbf{y} - \mathbf{X}_{\sigma}'\beta\|_2^2 \quad \text{s.t. } \|\beta\|_1 \leq t , \quad (1)$$

where σ , referred to as the active set, is a finite, ordered subset of features:

$\sigma = \{\phi_1, \dots, \phi_k\} \subset \mathcal{D}$, β is an associated coefficient vector, and

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X}_\sigma = \begin{bmatrix} -\phi_1(\mathbf{x}) - \\ \vdots \\ -\phi_k(\mathbf{x}) - \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) \dots \phi_1(x_n) \\ \vdots \\ \phi_k(x_1) \dots \phi_k(x_n) \end{bmatrix}.$$

It should first be noted that the constraint is active only if it disqualifies all *full-least-square* solutions that strictly minimize the squared residual. Let t^* be the supremum of active constraints, and let us assume in the following that $t < t^*$. In this case, the inequality constraint can be turned into an equality, by simple convexity arguments.

Along with defining the operator, two ways to solve the associated problem were proposed in [1]. The first one is an “exterior-point” active-set method that starts from a full-least-square solution and follows a path defined by changes of active set, on which the ℓ^1 -norm of the coefficient decreases down to the constraint. The second one consists in casting the problem into a quadratic program (QP), and applying standard QP methods. This can be done by adding to \mathcal{D} the negative counterpart of all features, thus allowing an additional nonnegativity constraint to all coefficients, which makes the ℓ^1 -norm constraint strictly linear.

In [2], a very different approach was introduced (“shooting” algorithm), that proceeds by means of a cyclic coordinate descent (CCD). Despite its simplicity and efficiency, it lacked popularity, partly because of the unnecessary use of the full-least-square solution as a starting point. However, it gained credit by its presentation in [3] in a more general setting.

In [4] were proposed two algorithms more related to QP, but fitted to the specificities of the problem. One is an interior-point active-set homotopy method that starts from an empty set and follows a path defined by changes of the active set on which the ℓ^1 -norm of the coefficient increases up to reaching the constraint. Its major interest, beside its efficiency, is that the followed path corresponds to all successive LASSO solutions for a constraint going from 0 to t (regularization path). It is best known as the LASSO modification of the LARS algorithm presented in [5], in a more general framework, with more details and clarity, and in a *regularized* formulation. The second algorithm, that we reformulate and transpose in the following, is a *surface* active-set descent method: all points of the path have an ℓ^1 -norm *equal* to the constraint t , and the squared residual decreases down to an optimal solution.

In section 2, we give a quick yet rather detailed exposition of this method, from which we derive in section 3 an algorithm that solves the regularized formulation of the LASSO. Section 4 presents experimental results in which the new algorithm outperforms the homotopy and CCD in terms of running time. We conclude in section 5 by mentioning additional advantages of this algorithm.

2 Iso-norm Descent

The unnamed algorithm introduced in [4], that we may refer to as the iso-norm descent method, is based on the following two facts:

- if the active set as well as the sign of the coefficients (*signed active set*) are known or assumed, the computation reduces to finding the minimizer of the squared residual on the corresponding hyperplane,
- the optimality of the signed active set can then easily be tested.

Let us explicit these two steps, and the algorithm that arises naturally from them.

2.1 Minimizer on a Signed Active Set

If the constraint is tightened by imposing a given signed active set $(\sigma, \boldsymbol{\theta})$, where $\boldsymbol{\theta} = \text{sign}(\boldsymbol{\beta})$, the problem reduces to

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{|\sigma|}} \|\mathbf{y} - \mathbf{X}'\boldsymbol{\beta}\|_2^2 \quad \text{s.t.} \quad \begin{cases} \boldsymbol{\theta}'\boldsymbol{\beta} &= t \\ \text{sign}(\boldsymbol{\beta}) &= \boldsymbol{\theta} \end{cases}. \quad (2)$$

Minimizing a convex function subject to a linear constraint is a simple QP: the minimizer is the only point on the constraint plane where the gradient of the minimized function is normal to the plane, i.e. such that any vector normal to the plane, say $\boldsymbol{\theta}$, is equal to the gradient (or its negative half) up to a (Lagrange) multiplier λ :

$$(2) \implies -\frac{1}{2}\nabla_{\boldsymbol{\beta}}\|\mathbf{y} - \mathbf{X}'\boldsymbol{\beta}\|_2^2 = \lambda\boldsymbol{\theta} \quad (3)$$

$$\implies \mathbf{X}(\mathbf{y} - \mathbf{X}'\boldsymbol{\beta}) = \lambda\boldsymbol{\theta} \quad (4)$$

$$\implies \mathbf{X}\mathbf{X}'\boldsymbol{\beta} = \mathbf{X}\mathbf{y} - \lambda\boldsymbol{\theta} \quad (5)$$

$$\implies \boldsymbol{\beta} = \underbrace{(\mathbf{X}\mathbf{X}')^{-1}\mathbf{X}\mathbf{y}}_{\boldsymbol{\beta}^*} - \lambda \underbrace{(\mathbf{X}\mathbf{X}')^{-1}\boldsymbol{\theta}}_{\Delta\boldsymbol{\beta}} \quad (6)$$

$$(2) \iff \begin{cases} \boldsymbol{\beta} = \boldsymbol{\beta}^* - \lambda\Delta\boldsymbol{\beta} \\ \boldsymbol{\theta}'\boldsymbol{\beta} = t \\ \text{sign}(\boldsymbol{\beta}) = \boldsymbol{\theta} \end{cases} \quad (7)$$

$$\iff \begin{cases} \boldsymbol{\beta} = \boldsymbol{\beta}^* - \lambda\Delta\boldsymbol{\beta} \\ \lambda = \frac{\boldsymbol{\theta}'\boldsymbol{\beta}^* - t}{\boldsymbol{\theta}'\Delta\boldsymbol{\beta}} \\ \text{sign}(\boldsymbol{\beta}) = \boldsymbol{\theta} \end{cases} \quad (8)$$

Note that $\boldsymbol{\beta}^*$ is the partial least-square (PLS) solution (the least-square solution on the selected feature set), $\Delta\boldsymbol{\beta}$ can be seen as a regularization direction, and λ as a regularization parameter.

Thus, after computing λ and $\boldsymbol{\beta}$, it remains for this solution to satisfy the sign constraint. If it does not, this implies that $(\sigma, \boldsymbol{\theta})$ is not optimal: the squared residual being convex in $\boldsymbol{\beta}$, given any point $\boldsymbol{\beta}_0$ in the simplex (the subspace of the plane $\boldsymbol{\theta}'\boldsymbol{\beta}$ defined by the sign constraint), it monotonically decreases on a line from $\boldsymbol{\beta}_0$ to $\boldsymbol{\beta}$, which will intercept the frontier of the simplex at which a coefficient (the same for any starting point $\boldsymbol{\beta}_0$) is zeroed, thus the corresponding

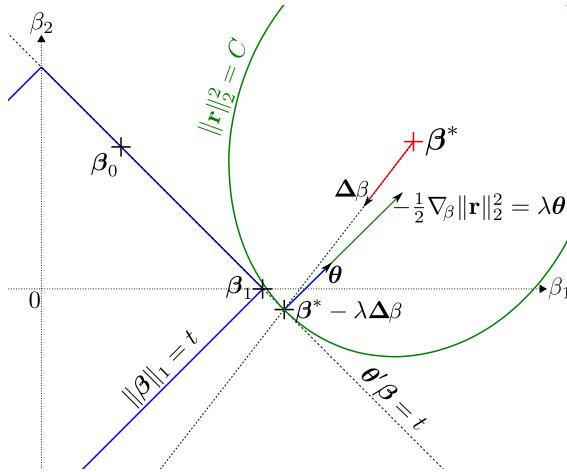


Fig. 1. From a signed active set $(\{\phi_1, \phi_2\}, (+, +))$, the regularized least-square is disagreeing with the sign hypothesis, indicating a better solution β_1 on reduced active set $(\{\phi_1\}, (+))$. \mathbf{r} is the residual $\mathbf{y} - \mathbf{X}'\beta$.

reduced active set contains a better minimizer. The computation can then be started again on the reduced set. An example is given in Fig. II that illustrates the sign disagreement and the different variables involved.

2.2 Karush-Kuhn-Tucker Conditions

If there is no sign disagreement, or when there is none after a sequence of optimization/shrinkage, the optimality of the active set can be tested via the necessary and sufficient Karush-Kuhn-Tucker conditions that reduce to the following (6):

$$\begin{cases} \forall \phi \in \sigma, & \phi(\mathbf{x})'(\mathbf{y} - \mathbf{X}_\sigma'\beta) = \text{sign}(\beta_\phi)\lambda \\ \forall \phi \in \mathcal{D} \setminus \sigma, & |\phi(\mathbf{x})'(\mathbf{y} - \mathbf{X}_\sigma'\beta)| \leq \lambda \end{cases} \quad (9)$$

The first condition was handled by the previous computation, and the optimality test lies in the second one: no inactive feature should have a correlation to the residual higher than that of active features, in absolute value (*over-correlation*). This can be intuitively understood: the correlation being the derivative of the squared residual w.r.t. the coefficient of a feature, if an inactive feature is *over-correlated*, it is possible to reduce the coefficient budget allocated to the active features by a sufficiently small amount and re-allocate it to the latter with greater benefice. A formal proof can be written following this sketch, that also shows that the better minimizer of the augmented active set involves an increased value of λ . Thus, given a signed active set and a sign-compliant solution to the associated (local) QP, the set augmented with an over-correlated feature contains a better solution (lower residual for equal ℓ^1 -norm). The sign imposed to the coefficient of the new active feature is that of its correlation, following equation 3.

2.3 Algorithm

This leads to the Algorithm II that shrinks the active set whenever the local QP solution is not sign-compliant, otherwise expands it with the most over-correlated feature if any, otherwise has converged to a LASSO solution. It is not mandatory to include the *most* over-correlated feature, but intuitively and empirically, this lessens the expected/average number of steps.

The algorithm has the descent property, since the residual decreases at each change of the active set, as shown previously, while the coefficient's ℓ^1 -norm is constant. It converges in at most $2^{|\mathcal{D}|}$ steps which is finite if \mathcal{D} is. The number of steps is however consistently observed to be $O(\min(n, |\mathcal{D}|))$ in experiments.

Algorithm 1. Iso-norm Descent

Input: $\mathbf{x} \in \mathcal{X}^n$, $\mathbf{y} \in \mathbb{R}^n$, $\mathcal{D} \subset \mathbb{R}^{\mathcal{X}}$, $t \in \mathbb{R}^+$, $(\sigma \subset \mathcal{D}, \boldsymbol{\beta} \in \mathbb{R}^{|\sigma|})$ s.t. $\|\boldsymbol{\beta}\|_1 = t$
Output: $(\sigma, \boldsymbol{\beta}) \in \arg \min_{\sigma \subset \mathcal{D}, \boldsymbol{\beta} \in \mathbb{R}^{|\sigma|}} \|\mathbf{y} - \mathbf{X}_{\sigma}' \boldsymbol{\beta}\|_2^2$ s.t. $\|\boldsymbol{\beta}\|_1 = t$
define $(x, i) = \min, \arg \min_t f(t)$: $x = \min_t f(t)$; $i \in \arg \min_t f(t)$
 $\theta \leftarrow \text{sign}(\boldsymbol{\beta})$
loop
 $\boldsymbol{\beta}^* = (\mathbf{X}_{\sigma} \mathbf{X}_{\sigma}')^{-1} \mathbf{X}_{\sigma} \mathbf{y}$
 $\Delta \boldsymbol{\beta} = (\mathbf{X}_{\sigma} \mathbf{X}_{\sigma}')^{-1} \theta$
 $\lambda = \frac{\theta \cdot \boldsymbol{\beta}^* - t}{\theta \cdot \Delta \boldsymbol{\beta}}$
 $\boldsymbol{\beta}' = \boldsymbol{\beta}^* - \lambda \Delta \boldsymbol{\beta}$
 $(\gamma, i) = \min, \arg \min_{i \in \{1, \dots, |\sigma|\}, \text{sign}(\beta'_i) \neq \text{sign}(\theta_i)} \frac{\beta_i}{\beta'_i - \beta_i}$
 $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \min(\gamma, 1)(\boldsymbol{\beta}' - \boldsymbol{\beta})$
if $\gamma \leq 1$ **then**
 $\sigma \leftarrow \sigma \setminus \{\phi_i\}$; update θ and $\boldsymbol{\beta}$ accordingly
else
 $(\phi, c) = \arg \max_{\phi \in \mathcal{D}, c = \phi(\mathbf{x})(\mathbf{y} - \mathbf{X}_{\sigma}' \boldsymbol{\beta})} |c|$
 if $|c| > \lambda$ **then**
 $\sigma \leftarrow \sigma \cup \{\phi\}$; $\theta \leftarrow (\theta, \text{sign}(c))'$; $\boldsymbol{\beta} \leftarrow (\boldsymbol{\beta}, 0)'$
 else
 return $(\sigma, \boldsymbol{\beta})$

3 Iso-regularization Descent

The Lagrange multiplier that appears in the resolution of the constrained least-square leads to an alternative *regularized* formulation of the LASSO: for any constraint t ,

$$\min_{\substack{\sigma \in 2^{\mathcal{D}}, \\ \boldsymbol{\beta} \in \mathbb{R}^{|\sigma|}}} \|\mathbf{y} - \mathbf{X}_{\sigma}' \boldsymbol{\beta}\|_2^2 \quad \text{s.t. } \|\boldsymbol{\beta}\|_1 \leq t \iff \min_{\substack{\sigma \in 2^{\mathcal{D}}, \\ \boldsymbol{\beta} \in \mathbb{R}^{|\sigma|}}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}_{\sigma}' \boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 ,$$

where λ is the Lagrange multiplier associated with the minimizer(s) of the constrained formulation. It can be shown that parameters t and λ are strictly decreasing functions of each other in this equivalence relation.

For none of the formulations can the parameter easily be tuned beforehand, but λ presents the advantage that its bounds are known in advance: the parameters such that $\mathbf{0}$ is the unique solution are $t_{\text{null}} = 0$ and $\lambda \geq \lambda_{\text{null}} = \max_{\phi \in \mathcal{D}} |\phi(\mathbf{x})' \mathbf{y}|$, but the supremum t^* of active constraints cannot be computed independently of the full-least-square solution itself, whereas the corresponding regularization parameter is $\lambda^* = 0$. Moreover, λ is more informative about the difference of residual between the regularized and ordinary least-square solutions.

In [7], the authors of the iso-norm descent method mention the possibility of solving the regularized formulation by multiple runs of their algorithm inside a grid or Newton-Raphson search to find the corresponding value of t .

A much simpler possibility appears perhaps more clearly in our exposition of the algorithm, that consists in using a fixed value of λ – the given regularization parameter – rather than computing at each step the value maintaining a constant ℓ^1 -norm. This results in an even simpler algorithm described in Algorithm 2.

It remains to prove that the descent property is preserved by this modification. This can be done by noting that if the ℓ^1 -norm of the tentative solution is smaller than that of the solution, this norm is increasing in all subsequent steps, as a corollary to the fact that the Lagrange multiplier is increasing in the iso-norm descent. If we consider such a step and β_0 and β_1 the corresponding successive coefficients, from the convexity of the squared residual, β_1 is its minimizer not only on the $\beta' \theta = \beta_1' \theta$ plane, but also on the associated half-space that contains β_0 and not the PLS solution. Thus any linear move toward β_1 yields a monotonic improvement.

Algorithm 2. Iso-regularization Descent

Input: $\mathbf{x} \in \mathcal{X}^n$, $\mathbf{y} \in \mathbb{R}^n$, $\mathcal{D} \subset \mathbb{R}^{\mathcal{X}}$, $\lambda \in \mathbb{R}^+$, starting point ($\sigma \subset \mathcal{D}, \boldsymbol{\beta} \in \mathbb{R}^{|\sigma|}$)
Output: $(\sigma, \boldsymbol{\beta}) \in \arg \min_{\sigma \subset \mathcal{D}, \boldsymbol{\beta} \in \mathbb{R}^{|\sigma|}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}_{\sigma}' \boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1$
define $(x, i) = \min, \arg \min_t f(t)$: $x = \min_t f(t)$; $i \in \arg \min_t f(t)$
 $\theta \leftarrow \text{sign}(\boldsymbol{\beta})$
loop
 $\boldsymbol{\beta}' = (\mathbf{X}_{\sigma} \mathbf{X}_{\sigma}')^{-1} (\mathbf{X}_{\sigma} \mathbf{y} - \lambda \theta)$
 $(\gamma, i) = \min, \arg \min_{i \in \{1, \dots, |\sigma|\}, \text{sign}(\beta'_i) \neq \text{sign}(\theta_i)} \frac{\beta_i}{\beta'_i - \beta_i}$
 $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \min(\gamma, 1)(\boldsymbol{\beta}' - \boldsymbol{\beta})$
if $\gamma \leq 1$ **then**
 $\sigma \leftarrow \sigma \setminus \{\phi_i\}$; update θ and $\boldsymbol{\beta}$ accordingly
else
 $(\phi, c) = \arg \max_{\phi \in \mathcal{D}, c = \phi(\mathbf{x})(\mathbf{y} - \mathbf{X}_{\sigma}' \boldsymbol{\beta})} |c|$
 if $|c| > \lambda$ **then**
 $\sigma \leftarrow \sigma \cup \{\phi\}$; $\theta \leftarrow (\theta, \text{sign}(c))'$; $\boldsymbol{\beta} \leftarrow (\boldsymbol{\beta}, 0)'$
 else
 return $(\sigma, \boldsymbol{\beta})$

4 Experiments

We reproduced the “speed trials” experiments described in [3] with the following three algorithms: the cyclic coordinate descent algorithm described in

Table 1. Speed trial experiments with the same settings as in [3]. The running times, in seconds, are averaged over 10 runs. All methods and trials were implemented in C in similar fashions.

n	$ \mathcal{D} $	Method	Population correlation between features					
			0	0.1	0.2	0.5	0.9	0.95
100	1000	homotopy	0.13	0.12	0.13	0.14	0.14	0.14
		iso- λ descent	0.09	0.09	0.09	0.10	0.10	0.10
		CCD	0.21	0.21	0.24	0.46	1.21	2.64
	5000	homotopy	0.63	0.65	0.62	0.67	0.62	0.60
		iso- λ descent	0.53	0.56	0.54	0.59	0.54	0.54
		CCD	1.39	1.39	1.54	2.32	7.39	8.53
	20000	homotopy	2.39	3.06	2.69	3.22	3.30	3.36
		iso- λ descent	2.06	2.56	2.25	2.68	2.76	2.85
		CCD	5.38	7.22	6.06	11.14	37.24	47.07
1000	100	homotopy	0.22	0.22	0.22	0.22	0.20	0.16
		iso- λ descent	0.19	0.19	0.19	0.19	0.17	0.13
		CCD	0.18	0.23	0.30	0.65	1.62	2.13
	5000	homotopy	0.71	0.70	0.70	0.70	0.70	1.05
		iso- λ descent	0.67	0.67	0.67	0.66	0.65	0.97
		CCD	0.61	0.75	0.91	1.44	4.31	8.07

that publication (CCD), the homotopy method, and the iso-regularization descent (iso- λ descent). The detailed settings and source code can be found at http://chercheurs.lille.inria.fr/~loth/iso-lambda-descent_xp.tgz.

The contrast between the results of the homotopy method in our experiments and those in [3] is explained by the fact that we gave it a simplified formulation and a proper implementation in C, closely resembling those of the iso- λ descent. This similarity, together with a lower complexity and comparable number of steps for going from a value of λ to another, indicate a consistently lower run time for the latter, which is confirmed by the experiments. The coordinate descent method shows slightly better results only for $|\mathcal{D}| < n$ and uncorrelated features.

5 Conclusion

The LASSO can thus be computed by three simple and efficient algorithms. Regardless of the running times exhibited in previous section, each offers specific advantages.

The cyclic coordinate descent is a very simple algorithm that is less subject to implementation hazards and conditioning problems. Also, since it does not involve the active Gram matrix, it supports sample reweighting and can thus be used to compute the elastic net operator. On the other hand, its convergence being asymptotic, a stopping criterion/parameter is needed.

The homotopy method gives the exact full regularization path (RP), while keeping a computational cost of the same order as the two others. However, this

is rarely needed, and solutions for a predefined set of regularization parameters are often sufficient.

In this respect, the iso-regularization descent method offers an efficient way to compute such a sequence, using one solution as a warm start for the next one, to which it converges in a small number of steps, very close to the number of steps on the RP. The path followed between the two is itself close to the RP, as it shares the property of simultaneously increasing the coefficient ℓ^1 -norm and decreasing the squared residual. Another interesting property of this algorithm is its ability to handle continuous feature sets, for example the set of all Gaussian functions over $\mathcal{X} = \mathbb{R}^p$, parameterized by their covariance matrix. This will be investigated, together with expositions of the proofs mentioned here, in further publications.

Acknowledgements. This work was supported by Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council and FEDER through the 'Contrat de Projets Etat Region' CPER 2007-2013. We also acknowledge the support of the ANR-09-EMER-007 Lampada project, and Pascal-2 European network of excellence.

References

1. Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58(1), 267–288 (1996)
2. Fu, W.J.: Penalized Regressions: The Bridge versus the Lasso. *Journal of Computational and Graphical Statistics* 7(3), 397 (1998)
3. Friedman, J., Hastie, T., Höfling, H., Tibshirani, R.: Pathwise coordinate optimization. *Annals of Applied Statistics* 1(2), 302–332 (2007)
4. Osborne, M.R., Presnell, B., Turlach, B.A.: A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis* 20(3), 389 (2000)
5. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. *The Annals of Statistics* 32(2), 407–499 (2004)
6. Nash, S.G., Sofer, A.: Linear and nonlinear programming. McGraw-Hill, New York (February 1996)
7. Osborne, M.R., Presnell, B., Turlach, B.A.: On the LASSO and Its Dual. *Journal of Computational and Graphical Statistics* 9(2), 319–337 (2000)

Logistic Label Propagation for Semi-supervised Learning

Kenji Watanabe, Takumi Kobayashi, and Nobuyuki Otsu

National Institute of Advanced Industrial Science and Technology (AIST),
AIST Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan
{kenji-watanabe,takumi.kobayashi,otsu.n}@aist.go.jp

Abstract. Label propagation (LP) is used in the framework of semi-supervised learning. In this paper, we propose a novel method of logistic label propagation (LLP). The proposed method employs logistic functions for accurately estimating the label values as the posterior probabilities. In LLP, the label of newly input sample is efficiently estimated by using the optimized coefficients in the logistic function, without such recomputation of all label values as in original LP. In the experiments on classification, the proposed method produced more reliable label values at the high degree of confidence than LP and ordinary logistic regression. In addition, even for a small portion of the labeled samples, the error rates by LLP were lower than those by the logistic regression.

Keywords: Semi-supervised learning, Label propagation, Logistic function, Posterior probability.

1 Introduction

In the analysis of actual measurement data such as the biological signals [1, 2], the given labels are often unreliable. Because, objects to be measured inherently contain some physical and biological uncertainty, and some labels might be incorrectly assigned by human intuition. Whereas, reliable labels would be available for a small portion of the samples. In such case, a method of semi-supervised learning is effectively applied to analyze the measurement data, estimating the label values of samples. In addition, it is favorable that the estimated label values provide us the degree of confidence of each sample.

The method of label propagation (LP) [3] is used in the framework of semi-supervised learning to estimate label values in, for example, patch labeling [4], image annotation [5] and image matting [6, 7, 8]. In LP, the label values are estimated based on the similarities (affinities) between samples, given a few labeled (reliable) samples. LP is less affected by the unreliable labels in training samples unlike supervised learning methods which totally rely on all the given labels. However, in the case that a sample is newly input, we must recompute all label values for each time in LP unlike (supervised) classifiers such as by logistic regression.

Logistic regression (LR) [9] is one of the well-known binary supervised learning methods and it produces promising performances in biological signals, such as electro encephalography (EEG) [10]. Multi-nominal logistic regression (MLR) [9] is a natural

extension of LR to cope with multi-class classification problems and estimates the posterior probabilities. In LP, when binary (0/1) labels are assigned, the estimated label values roughly range from 0 to 1 and thus can be regarded as the posterior probabilities, as well as MLR.

In this paper, we propose a novel method of semi-supervised learning, incorporating logistic functions into LP in order to accurately estimate the label values as the posterior probabilities. We call this method logistic label propagation (LLP). The method of LLP optimizes the coefficients in the logistic function by using the labeled training sample. Therefore, the labels of newly input samples are estimated by using the logistic function without such recompilation of all label values as in original LP. The LLP is based on similarities between samples without specific prior models and, in that point, differs from model-based methods such as semi-supervised logistic regression [11] which utilizes parametric prior models.

2 Label Propagation

We start with briefly reviewing the method of label propagation (LP) [3]. Suppose we have N samples. For C -class classification problem, let \mathbf{x}_i and α_{ic} be the feature vector and the c -th class label value of the i -th sample, respectively. First N_t samples are given and known (labeled), and the rest N_u samples are unknown (unlabeled). Note that the label values of the labeled samples are fixed to either 0 or 1, $\alpha_{jc} \in \{0,1\}$ ($1 \leq j \leq N_t$). The rest unknown label values (α_{jc} , $N_t + 1 \leq j \leq N_u$) are to be estimated by using the following cost function,

$$\begin{aligned} J(\boldsymbol{\alpha}_{U_c}) &= \frac{1}{2} \sum_c^C \sum_{ij}^N S_{ij} (\alpha_{ic} - \alpha_{jc})^2 = \sum_c^C \boldsymbol{\alpha}_c^T (\mathbf{D} - \mathbf{S}) \boldsymbol{\alpha}_c = \sum_c^C \boldsymbol{\alpha}_c^T \mathbf{L} \boldsymbol{\alpha}_c \\ &= \sum_c^C \begin{bmatrix} \boldsymbol{\alpha}_{K_c} \\ \boldsymbol{\alpha}_{U_c} \end{bmatrix}^T \begin{bmatrix} \mathbf{L}_{KK} & \mathbf{L}_{KU} \\ \mathbf{L}_{UK} & \mathbf{L}_{UU} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{K_c} \\ \boldsymbol{\alpha}_{U_c} \end{bmatrix}, \end{aligned} \quad (1)$$

where S_{ij} is a similarity between the i -th and j -th samples, \mathbf{D} is a diagonal matrix of $D_{ii} = \sum_j^N S_{ij}$ and $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is called Laplacian matrix. The global optimum for the above cost function is obtained by using its derivatives:

$$\frac{\partial J(\boldsymbol{\alpha}_{U_c})}{\partial \boldsymbol{\alpha}_{U_c}} = 2\mathbf{L}_{UU} \boldsymbol{\alpha}_{U_c} + 2\mathbf{L}_{UK} \boldsymbol{\alpha}_{K_c} = 0, \quad (2)$$

Hence,

$$\therefore \boldsymbol{\alpha}_{U_c} = -\mathbf{L}_{UU}^{-1} \mathbf{L}_{UK} \boldsymbol{\alpha}_{K_c}. \quad (3)$$

Since the given label values are fixed to 0 or 1 ($\alpha_{jc} \in \{0,1\}$ $j \in K$), the estimated label values $\boldsymbol{\alpha}_{U_c}$ are almost limited between 0 and 1 which enables us to regard them as posterior probabilities.

As to the similarity, we simply employ the Gaussian kernel similarity in this study,

$$S_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2\right), \quad (4)$$

where σ is the sample standard deviation.

3 Logistic Label Propagation

We describe the proposed method, logistic label propagation (LLP). The method incorporates logistic functions into LP (Sec.2) in order to accurately estimate the label values as the posterior probabilities, optimizing the coefficients in the logistic function.

3.1 Logistic Function

The logistic function is widely employed to approximate the posterior probabilities with the generalized linear model, such as in MLR [9]. Thus, the logistic function is favorably incorporated into the framework of LP for representing the posterior probabilities. The mathematical form of the logistic function is defined as follows:

$$\alpha_{ic} = \frac{\exp(\mathbf{a}_c^\top \mathbf{x}_i + b_c)}{\sum_c \exp(\mathbf{a}_c^\top \mathbf{x}_i + b_c)} = \frac{\exp(\mathbf{w}_c^\top \hat{\mathbf{x}}_i)}{\sum_c \exp(\mathbf{w}_c^\top \hat{\mathbf{x}}_i)}. \quad (5)$$

We include the bias term in the vectors as $\mathbf{w}_c = (\mathbf{a}_c^\top, b_c)$ and $\hat{\mathbf{x}}_i = (\mathbf{x}_i^\top, 1)$ to simplify the notation. The normalization by $\sum_c \exp(\mathbf{w}_c^\top \mathbf{x}_i)$ ensures the probability constraint (i.e., unit sum), but it makes the optimization in LLP complicate. In this study, we introduce the following simpler form:

$$\hat{\alpha}_{ic} = \gamma_i \exp(\mathbf{w}_c^\top \hat{\mathbf{x}}_i), \quad s.t. \quad \sum_c \hat{\alpha}_{ic} = 1. \quad (6)$$

where $\hat{\alpha}_c$ is the c -th class label value, and the coefficient γ_i controls the outputs subject to the probability constraint. In LLP, we optimizes \mathbf{w}_c (and accordingly γ_i).

3.2 Optimization

In the proposed method, we replace the (approximated) posterior probabilities α_{U_c} in equation (1) with the above-defined logistic function in equation (6):

$$\begin{aligned} E(\mathbf{w}) &= \sum_c \begin{bmatrix} \mathbf{a}_{K_c} \\ \hat{\mathbf{a}}_{U_c} \end{bmatrix}^\top \begin{bmatrix} \mathbf{L}_{KK} & \mathbf{L}_{KU} \\ \mathbf{L}_{UK} & \mathbf{L}_{UU} \end{bmatrix} \begin{bmatrix} \mathbf{a}_{K_c} \\ \hat{\mathbf{a}}_{U_c} \end{bmatrix} \\ &= \sum_c \left\{ \mathbf{a}_{K_c} \mathbf{L}_{KK} \mathbf{a}_{K_c} + 2 \mathbf{a}_{K_c} \mathbf{L}_{KU} \hat{\mathbf{a}}_{U_c} + \hat{\mathbf{a}}_{U_c} \mathbf{L}_{UU} \hat{\mathbf{a}}_{U_c} \right\}, \\ \alpha_{jc} &= \gamma_j \exp(\mathbf{w}_c^\top \hat{\mathbf{x}}_j), \quad s.t. \quad \sum_c \alpha_{jc} = 1. \end{aligned} \quad (7)$$

The optimal \mathbf{w} of LLP is obtained by minimizing equation (7). However, it is not analytically solved unlike LP. We simply take an iterative approach using the quasi-Newton method, although the other methods such as conjugate-gradient method would be also applicable. In equation (7), by letting the derivative of the cost function with respect to \mathbf{w}_c be 0,

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}_c} = 2\hat{\mathbf{X}}\text{diag}(\hat{\boldsymbol{\alpha}}_{U_c})[\mathbf{L}_{UK}\boldsymbol{\alpha}_{K_c} + \mathbf{L}_{UU}\hat{\boldsymbol{\alpha}}_{U_c}]. \quad (8)$$

we obtain the following update for \mathbf{w}_c :

$$\mathbf{w}_c^{(\text{new})} \leftarrow \mathbf{w}_c - \eta \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}_c}. \quad (9)$$

where η is the parameter for step size. After updating \mathbf{w}_c (for all c), the normalization variables γ_j are determined such that the (estimated) posterior probabilities have unit sum:

$$\gamma_j = \frac{1}{\sum_c^C \exp(\mathbf{w}_c^{(\text{new})\top} \hat{\mathbf{x}}_j)}. \quad (10)$$

The above iteration is repeated until convergence.

In LLP, we can easily estimate the label values of the newly input sample by using equation (5) with the optimized \mathbf{w}_c , without such recomputation of all label values as in original LP.

$$c_i^* = \arg \max_c \hat{\alpha}_{ic}. \quad (11)$$

The posterior probabilities approximated by LLP, label values $\hat{\alpha}_{ic^*}$, can be regarded as the degree of confidence for the estimated labels c_i^* ; that is, we can consider the estimated label with the higher posterior probability as more reliable.

4 Experiments

In practical scenes, we often need to analyze the unreliably measured samples that are obtained in unreliable measurement process and improperly classified on the basis of a priori knowledge such as the already labeled sample. In such cases, a degree of confidence for the estimated label would help us to cope with the unreliable measurement data. In the following experiments on classification, we show the performances of the proposed method compared to the other methods in view of the degree of confidence.

For comparison, we employ the ordinary multi-nominal logistic regression (MLR) [9] and label propagation (LP) [3]. In MLR, the coefficients in the logistic function are learnt from only the labeled samples in supervised learning, and the labels of the test samples are estimated by using the learnt logistic function. While, in LP and LLP, the label values are estimated by using both the training and the test (unknown) samples via similarities in the framework of semi-supervised learning.

4.1 Distribution of Estimated Labels and Their Degree of Confidence

We present the estimated labels and the degree of confidence produced by MLR, LP and the proposed method (LLP) for *Iris* dataset [12]. The dataset consists of 150 samples of 4-dimensional feature vectors labeled in 3 classes. We randomly draw 30 (labeled) training and 120 (unlabeled) test samples. Fig. 1 shows the sample distribution in the 2-dimensional discriminant space constructed by linear discriminant analysis. The 3 classes of the samples are represented by the circles, triangles and squares, respectively. In Fig. 1(a), the red and black colors denote training and test samples, respectively. In Fig. 1(b, c, d), the colors (blue, purple and green) and tones indicate the estimated labels and the degree of confidence of the test samples, respectively. The deepest colors represent the high degree of confidence (> 0.8) while the lightest colors represent the low degree (< 0.4). The miss-classified samples are indicated by the red contours.

From Fig. 1(b), it is seen that MLR correctly recognized most of the test samples. However, the degrees of confidence produced by MLR were high even for the samples located near the boundary between different classes. Moreover, in the class

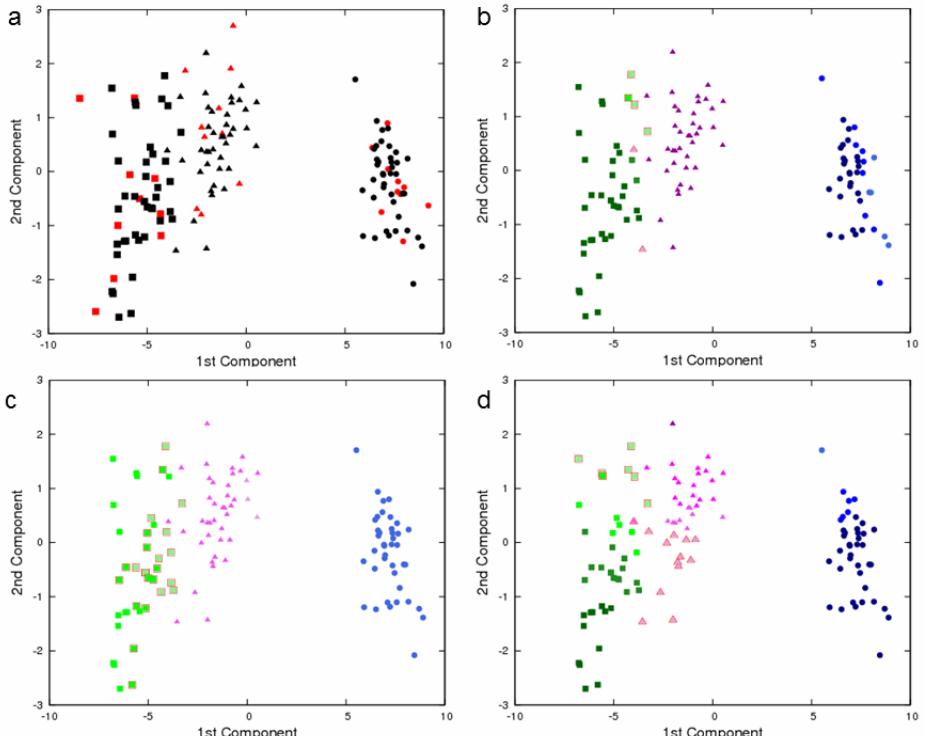


Fig. 1. The estimated label and their degree of confidence on the sample distribution in the discriminant space. (a) shows the distribution of test and training samples, and the results are shown in (b) MLR, (c) LP and (d) LLP. Details are in the text.

which is represented by circles, the degrees of confidence for the samples which located near the boundary were higher than those for the samples far from the boundary. It would be rather desirable to provide lower degrees of confidence for samples which locate near the boundary. Thus, we can say that the degree of confidence produced by MLR is not natural and it is difficult to use it generally as the degree of confidence for the unlabeled samples in unreliable measurement data.

On the other hand, LP miss-classified many samples in the class 3 as shown in Fig. 1(c). Besides, all the degrees of confidence were equally low (< 0.53), which is not suitable for finally deciding labels in the analysis of the unreliable measurement data.

In LLP which integrates LR and LP, the degree of confidence naturally changes according to the sample distribution. This is because the optimization of the logistic function in LLP is performed based on the similarity both of labeled and unlabeled samples. In these points, this experiment shows that LLP provides favorable outputs than MLR and LP.

4.2 Performance Using Small Portion of Training Samples

In the analysis of measurement data, it is often that only a small portion of the samples is labeled and human is engaged to categorize samples in such situations. The degree of confidence facilitates the analysis of the measured samples in a case that the labeled samples are few. Thus, it is necessary to obtain reliable degree of confidence for labeling.

In order to confirm the performance in such cases, we conducted the experiments on classification using a variety of small portion ($p\%$) of training samples, ranging p from 0.5 to 25. The performances are measured on the samples that have high degree of confidence in order to ensure that it really means the low error rate. In this experiment, the threshold for the degree of confidence is set to 0.8.

The error rates are compared by using the benchmark datasets for binary and multi-class classification; *German*, *Heart*, *Balance* and *Segment* [12] (see Table 1). The error rates were calculated by using 5-fold cross validation. We used the identical test set across the various ratios (p) of training samples. The training samples were selected in decreasing order of the following value in order to favorably select the inliers:

$$\varepsilon_i = \sum_{i \neq j} S_{ij}, \quad (12)$$

Table 1. Summary of the benchmark datasets for each validation set

	# of class	# of training	# of test	# of feature
German	2	800	200	24
Heart	2	216	54	13
Balance	3	502	123	3
Segment	7	1848	462	19

Fig. 2 shows the error rates for varied ratio (p) of training samples. The error rates of LLP were lower than those of MLR in all p . The minimum error rate of LLP was 5.40% for *German* (Fig. 2-a), 0% for *Heart* (Fig. 2-b), 0% for *Balance* (Fig. 2-c) and 2.55% for *Segment* (Fig. 2-d). In this experimental setting, the maximum degrees of confidence by LP were almost estimated to be $1/K$ around (data not shown). Thus, LP is not suitable to the evaluation of degree of confidence for the unlabeled samples. These results suggest that LLP is superior to MLR for a small portion of the training samples, and that the degrees of confidence provided by LLP are more reliable than those by MLR.

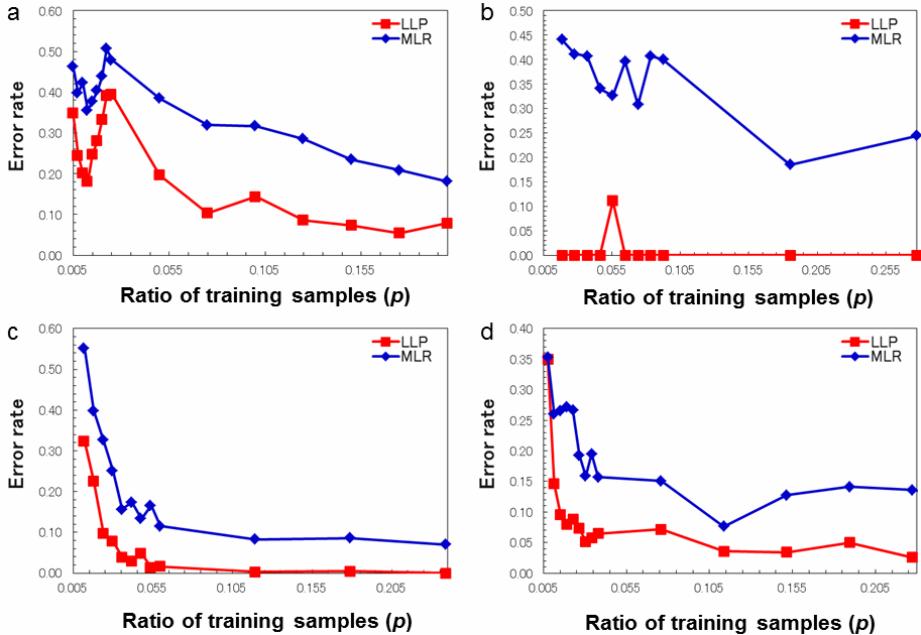


Fig. 2. Error rates for varied ratio (p) of training samples. These figures show the results for *German* (a), *Heart* (b), *Balance* (c) and *Segment* (d). The red and blue lines show the results of LLP and MLR, respectively.

5 Conclusion

In this paper, we proposed logistic label propagation (LLP) in the framework of semi-supervised learning, estimating the label values as the posterior probabilities. By using the benchmark datasets, we showed that the distribution of label values obtained by LLP were natural and favorable than those by MLR and LP. It was also shown that LLP can reliably estimate the degree of confidence (label values) of the unlabeled samples even for a few labeled samples.

The performance of proposed method could be affected by the choice of similarity measure in some datasets. Therefore, our future work includes similarity design.

References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. *J. Molecular Biology* 215, 403–410 (1990)
2. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* 25, 3389–3402 (1997)
3. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of International Conference on Machine Learning (ICML 2003), Washington DC (2003)
4. Bishop, C.M., Ulusoy, I.: Object recognition via local patch labeling. In: Proceedings 2004 Workshop on Machine Learning (2005)
5. Kang, F., Jin, R., Sukthankar, R.: Correlated label propagation with application to multi-label learning. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), New York, vol. 2, pp. 1719–1726 (2006)
6. Levin, A., Lischinski, D., Weiss, Y.: A Closed Form Solution to Natural Image Matting. *IEEE Transactions on PAMI* 30(2), 228–242 (2008)
7. Levin, A., Rav-Acha, A., Lischinski, D.: Spectral Matting. *IEEE Transactions on PAMI* 30(10), 1699–1712 (2008)
8. Cheng, H., Liu, Z., Yang, J.: Sparsity Induced Similarity Measure for Label Propagation. In: 2009 IEEE 12th International Conference on Computer Vision (ICCV), Koyoto, pp. 317–324 (2009)
9. Bishop, C.M.: Pattern Recognition And Machine Learning. Springer Science+Business Media LCC (2006)
10. Tomioka, R., Aihara, K., Muller, K.-R.: Logistic Regression for Single Trial EEG Classification. In: Advances in Neural Information Processing Systems, vol. 19, pp. 1377–1384 (2007)
11. Amini, M.-R., Gallinari, P.: Semi-Supervised Logistic Regression. In: 15th European Conference on Artificial Intelligence, pp. 390–394 (2002)
12. Frank, A., Asuncion, A.: USI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science (2010),
<http://archive.ics.uci.edu/ml>

A New Framework for Small Sample Size Face Recognition Based on Weighted Multiple Decision Templates

Mohammad Sajjad Ghaemi^{1,2}, Saeed Masoudnia⁴, and Reza Ebrahimpour^{3,5}

¹ M.Sc. Student in Computer Science, Department of Mathematical Sciences,
Sharif University of Technology, Tehran, Iran

² Research Center of Intelligent Signal Processing, Tehran, Iran

³ School of Cognitive Sciences, Institute for Studies on Theoretical Physics and
Mathematics, Tehran, Iran

⁴ Islamic Azad University, Central Tehran Branch and Young Researchers Club,
Tehran, Iran

⁵ Department of Electrical Engineering, Shahid Rajaee Teacher Training University,
Tehran, Iran

Abstract. In this paper a holistic method and a local method based on decision template ensemble are investigated. In addition by combining both methods, a new hybrid method for boosting the performance of the system is proposed and evaluated with respect to robustness against small sample size problem in face recognition. Inadequate and substantial variations in the available training samples are the two challenging obstacles in classification of an unknown face image. At first in this novel multi learner framework, a decision template is designed for the global face and a set of decision templates is constructed for each local part of the face as a complement to the previous part. The prominent results demonstrate that, the new hybrid method based on fusion of weighted multiple decision templates is superior to the other classic combining schemes for both ORL and Yale data sets. In addition when the global and the local components of the face are combined together the best performance is achieved.

1 Introduction

Face recognition is one of the active research areas in pattern recognition, computer vision and machine learning [2]. In face recognition and identification systems, the purpose of the system is to identify face images that are not seen by the system before, so comparing a given face image against model of stored face images in the available database of known face images should be done, to obtain the correct match [1][2][3]. Despite the fact that, there are plenty of approaches to solve this problem in constrained conditions with promising results suggested in the past two-decades, but this is still a very troublesome problem because of small number of training samples, diversified direction in light, different facial expression, and variant face poses, in real world face recognition

applications. To overcome the similar problems, the face recognition techniques are divided into three categories: (1) Holistic approach, (2) Local approach and (3) Hybrid approach [7][13].

1. Holistic (global) approach: In this approach the input of a classifier is a whole face image that is stored in a single feature vector to represent the global face. Although, they work well for classifying frontal views of faces, but they are not robust against pose changes because global features are highly sensitive to the face rotation [4].
2. Local (component-based) approach: Local facial component is an alternative to the global approach. In some cases that face poses are changed, recognition based on face's local components has a better performance in contrast with the global approach. In this method instead of using global face as a main input space, some local components of the face like eye, nose, mouth and cheek are considered as the input space [5][11].
3. Hybrid (mixture)approach: To recognize a face, hybrid approaches use both the local and global features. In these methods the advantages of both approaches are taken to improve the recognition rate of the system, because more comprehensive information could be utilized [6][8].

Although global approach provides all the detailed information of face, such as shape or texture that are important to discern the proper class rather than the local components, but when a wide range of pose variation is available, the usage of holistic methods lead the classifier's performance to downfall. On the other hand, despite the local facial components are robust against pose variations, but in these methods, geometrical feature extraction is very hard moreover when feature extraction is applied to a low resolution, 2D and gray scale face image.

From the other point of view, in the holistic method the learning mechanism that is sensitive to the appearance of the global face, is heavily influenced by the number of training samples for each face. In many applications the number of training samples are quite little, while the classic families of learning mechanisms essentially require sufficiently large training set for a good generalization and improve the classifier's performance.

Anyway local component features alone, are not enough to completely represent a face, so the hybrid methods can overcome the above difficulties. The hybrid methods intrinsically have an aptitude to offer better performance than the individual holistic or local methods, since more significant information could be utilized. The major factors that affect the performance of hybrid methods comprise how to determine which features should be selected to combine and how to combine them to preserve their profits and prevent their disadvantages simultaneously [14]. These problems are discussed in the field of machine learning specially in the domain of multiple classifier system and ensemble learning. Nevertheless, these problems are currently unsolved.

A major problem connected to face recognition is the so-called curse of dimensionality. In according to the definition, when the number of features in the training of a classifier is very large, the number of features should reduce to a sufficient minimum. There is more than one reason for this issue, but the

most remarkable reason is that, when the higher the ratio of the number of training patterns N to the number of free classifier parameters, the better the generalization properties of the resulting classifier. A large number of features are immediately following a large number of classifier parameters, thus, for a finite and usually limited number N of training samples, keeping the number of features as small as possible yield to a good generalization capability.

Therefore the choice of the feature extractor is very important. Among various feature extractor, the statistical method considered in this study, is principle component analysis (PCA). Because the PCA is an ordinary feature extractor method whereby the significance of our proposed framework is demonstrated. The PCA is used mainly for dimensionality reduction which can be used for feature extraction to train classifiers. In this method images are projected into a subspace such that the first orthogonal dimension of this subspace captures the greatest amount of variance among the images, whereas the last dimension indicates the least amount of variance. On the other hand multiple classifier fusion may generate more accurate classification than each classifier apart. The classifier fusion can be implemented at two levels, feature level and decision level.

In this Work the fusion occurs at both decision level and feature level in a hierarchical model, at first in feature level a group of homogeneous neural networks which are made diverse through parameters initialization and design for a specific part of the face, fuses to construct a decision template to the corresponding part of the face, and at last in the decision level, fusion of weighted multiple decision templates (WMDT) solves the problem [9].

2 Decision Templates for Multiple Classifier Fusion

In this section we develop the method called decision templates (DTs) which are classifier fusion scheme based on a set of C matrices that combines outputs of classifiers by comparing them to a typical template for each class. For calculating the final support for each class, fusion of DT uses all classifier outputs.

2.1 Decision Profile(DP)

Let $X \in \mathbb{R}^n$ be the representation of feature vector, and $\{D_1, D_2, \dots, D_L\}$ be the set of L classifiers and $\{1, 2, \dots, C\}$ be the set of C class labels, which are adopted based on the outputs of base classifiers that are multi layer perceptron(MLP) in this work. We denote the output of the i th classifier as $D'_i(X) = [d_{i,1}(X), \dots, d_{i,C}(X)]$, where $d_{i,j}(X)$ implies the degree of support that is given by classifier D_i to the hypothesis that X belongs to class j . The classifier outputs can be arranged in a decision profile(DP) as the matrix below:

$$DP(X) = \begin{bmatrix} d_{1,1}(X) & \dots & d_{1,j}(X) & \dots & d_{1,C}(X) \\ \vdots & & & & \\ d_{i,1}(X) & \dots & d_{i,j}(X) & \dots & d_{i,C}(X) \\ \vdots & & & & \\ d_{L,1}(X) & \dots & d_{L,j}(X) & \dots & d_{L,C}(X) \end{bmatrix} \quad (1)$$

In the $DP(X)$ matrix, i th row ($dp_i'(X)$) implies the output of classifier D_i , and j th column ($dp_j(X)$) is representative of the support from classifier D_1, \dots, D_L for class j ,

$$dp_j(X) = [d_{1,j}, \dots, d_{L,j}]^T \quad (2)$$

2.2 Decision Templates(DTs)

Let $Z = \{z_1, z_2, \dots, z_N\}$, $z_i \in \Re^n$, be the label of training data set. The decision template DT_i for class i is the $L \times c$ matrix which is the average of the decision profiles of the elements of the training set Z labeled in class i , therefore we have c matrix templates found as the class means of the classifier outputs.

$$DT_i = \frac{1}{N_i} \left(\sum_{z_j \in Z} DP(z_j) \right), i \in \{1, \dots, c\} \quad (3)$$

where N_i is the number of elements in Z from class j .

When $x \in \Re^n$ is submitted for classification, the DT scheme matches $DP(x)$ to DT_i , $i = 1, 2, \dots, c$. The general idea is to compare the matrix $DP(x)$ to c template matrices (DT_1, \dots, DT_c), any measure that does this might be appropriate. In this study the following suitable measure of similarity is used:

$$S(DP(x), DT(x)) = \min_j \|DP(x) - DT_j(x)\|_F^2, j \in \{1, 2, \dots, c\} \quad (4)$$

and also the j th class would be selected as a desirable output class. The Frobenius norm, denoted by $\|\cdot\|_F$, is defined for an $m \times n$ matrix A by:

$$\|A\|_F = A^T A = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{1/2} \quad (5)$$

2.3 Fusion of Weighted Multiple Decision Templates

In this section we propose our new algorithm, fusion of weighted multiple decision templates. The main idea of this algorithm is minimizing the distance between DTs and DP, or maximizing the similarity (computed in measures of similarity Eq. 4) between DTs and DP. First we should divide the input space into several parts and apply the DT algorithm for each part, then calculating the support from classifiers for each DT and saving the average of each DT's support from classifier as a new weight. By applying these new weights to aggregate a Euclidean distance between DTs and DP as follows:

$$S(DP(x), DT(x)) = \min_j \frac{\sum_{i=1}^p W_{ij} \|DP_i(x) - DT_{ij}(x)\|_F^2}{\sum_{i=1}^p W_{ij}}, j \in \{1, 2, \dots, c\} \quad (6)$$

where p is the number of face parts in addition to global face.

Among the all possible options to select the weight such as, $1/Average(dp_{ij}(x))$, $Average(dp_{ij}(x))$, $1/Max(dp_{ij}(x))$, $Max(dp_{ij}(x))$, $1/Min(dp_{ij}(x))$, $Min(dp_{ij}(x))$, etc. The experimental results show that choosing $1/Average(dp_{ij}(x))$ as a new weight has more influence in minimizing the cost function with respect to the other weight candidates. As we know the support from classifier vector ($dp_{ij}(x)$) implies the degree of assurance for each classifier that x comes from class j , thus assigning $1/Average(dp_{ij}(x))$ to the W_{ij} can minimizes the Eq.(6), consequently the j th class would be selected as the desirable output class.

But the remaining question is, how to divide the input space into several new parts such that when we fuse their weighted decision templates, the total performance of system improves. In face recognition, one obvious solution is that combine the local and global components of the face, but extracting the face components is a difficult problem that needs some preprocessing and statistical or morphable models hence, many people prefer partitioning the face horizontally into three parts[10] to extract face local features. Here we show that this is not the dominant approach in all cases and partitioning the face vertically is better than horizontally in some problems. A better approach in face recognition is combining global face with face local components by WMDT as described in the Eq.(6). In the next section we show that our new idea (dividing face into three parts vertically and also fusing global face with face local components) enhances the face recognition performance on the two popular data sets, Yale and ORL. As shown in the Fig. 1.

Step 1. The learning phase of the system,

Step 2. A new weight is assigned to each DT, by reversing the average of the support from classifier,

Step 3. The distance is computed between the i th DP and the ij th DT and save it as D_i , then test the entire system with the test data and use the

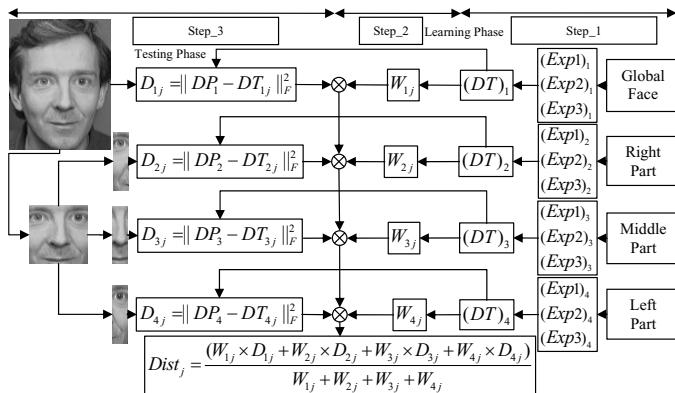


Fig. 1. Architecture of WMDT framework scheme

Eq.(6) to classify an unknown face image to the real class in according to the most similar DT.

3 Experimental Studies

In this section, we inspect the performance of our proposed algorithm WMDT for face recognition. The face recognition task is a multi class, classification problem that is assigning each test image into a nearest class through the measures of similarity rule Eq.(6) that is learned from the training data. We use the Yale and ORL face databases in our experiments. The Yale face database consists of 165 gray scale images of 15 individuals. There are 11 images per subject. The images demonstrate variations in facial expression and lighting condition. The ORL face database contains 400 face images, of 40 people (10 samples per person). These images have different variations in expression such as: smiling or non-smiling, open or closed eyes and also facial details like: glasses or no glasses and also, they were taken with tolerance for some side movement and rotation of the face up to 20 degrees. For decision template of the global face we use the whole face images and for decision templates of the local components we use face images that are cropped and aligned manually. The size of each cropped image is 48x48 (ORL) and 64x64 (Yale) pixels, with 256 gray levels per pixel. The pixel values are then normalized to [0,1]. The role of the partitioning method in face image is considered, by dividing each face image automatically into 3 equal parts horizontally and vertically, respectively. Ultimately the PCA is applied for extracting the relevant features from the global and local parts of the face separately, then the data set is partitioned into two sets: training set and test set. For ease of representation, L_m/U_n means m images per person are randomly selected for training and the remaining n images are for testing. Moreover, we test our proposed framework by allowing the m to take the values in the range of 2, 3, 4, 5 in the training set to analysis that how the performance of the proposed method is influenced by the number of training images per person. In our classification system, each base classifier is a MLP which is trained via BackPropagation(BP) learning algorithm and to construct each DT, 3 MLPs are diversified with ranging the MLP's parameters as following: $Number\ of\ Epoch \in \{40, 50, 60\}$, $Learning\ Rate \in \{0.05, 0.2, 0.35\}$, $Momentum \in \{0.25, 0.35, 0.45\}$. The reported results are averaged over 20 runs on randomly chosen training set.

3.1 Experimental Results

Tables 1, 2 summarize the WMDT algorithm and compare the method of face partitioning in two ways, vertical and horizontal. Then the influence of the template combination of each dividing method in conjunction with global face is investigated. Further more the role of the aggregating these templates with respect to employing the proper weight is taken into account, and the prominent results show that, the average of the support from classifiers for each DT is a good measure of similarity therefore by reversing this measure the minimum

Table 1. Recognition Rate on ORL data sets

ORL	Fusion of face's local parts				Local & global parts fusion				ANN			
	<i>Unweighted</i>		<i>Weighted</i>		<i>Unweighted</i>		<i>Weighted</i>		<i>Face parts</i>			
	Hor	Ver	Hor	Ver	Hor	Ver	Hor	Ver	Hor	Ver	Global	Fuse
L_2/U_8	63.5	74.3	74.3	80.1	81.8	86.6	89.7	92.8	66.9	67.9	74.2	77.9
L_3/U_7	76.4	79.8	82.3	85.1	87.7	91.1	95.4	97.5	75.3	78.5	82.6	85.4
L_4/U_6	79.1	83.8	87.2	90.5	92.6	95.2	96.9	98.8	80.8	82.7	89.5	92.2
L_5/U_5	87.3	92.1	90.8	95.5	94.2	97.2	98.5	99.8	85.2	87.5	91.5	95.0

Table 2. Recognition Rate on Yale data sets

Yale	Fusion of face's local parts				Local & global parts fusion				ANN			
	<i>Unweighted</i>		<i>Weighted</i>		<i>Unweighted</i>		<i>Weighted</i>		<i>Face parts</i>			
	Hor	Ver	Hor	Ver	Hor	Ver	Hor	Ver	Hor	Ver	Global	Fuse
L_2/U_9	73.1	74.9	77.8	79.5	79.4	85.0	86.4	91.8	68.7	62.8	74.0	76.7
L_3/U_8	76.5	80.1	80.5	83.3	82.4	90.6	89.5	94.3	72.8	75.6	78.6	79.1
L_4/U_7	78.2	82.8	83.2	84.4	85.3	91.1	89.1	95.9	74.5	77.5	83.1	87.0
L_5/U_6	79.8	83.4	84.5	86.4	86.4	92.7	90.7	96.2	75.6	78.6	86.4	89.4
L_6/U_5	85.0	87.7	88.5	90.1	90.9	96.0	95.4	98.9	79.6	83.4	89.3	91.2

distance between DTs and DP is obtained. Although in general, global face is more significant than face components however, composition of face components vertically based on WMDT produce better results in contrast with the global face. In addition, in this new hybrid method by combining global face and face local components together the performance is raised remarkably because of the complementary knowledge in both of the features. On the other hand in the Artificial Neural Network (ANN) column the best result of fusion of base classifiers with respect to the classic techniques of combining are reported, such as min, max, average, product, etc. Analogies between WMDT, ANN and other combining strategies demonstrate the efficiency of the proposed framework for small sample size face recognition.

4 Conclusion

In this study a new multi learner framework is proposed to defeat the critical mentioned questions. First, the face images are divided into three parts vertically to extract as much as possible information of the local components. In this hybrid framework, local and global features with absolutely different characteristics, but with containing complementary information assist the hybrid method to improve the generalization ability with respect to pose variations and small number of training samples. In addition the experimental results show that, dividing the face images into vertical subspaces are meaningfully better than horizontal subspaces, but the performance of the proposed system is raised dramatically when both of the local components and global features are fused together in

according to WMDT algorithm, so the combination of these two approaches enhances the recognition rate of the face classification task. Our future work will focus on finding a new dividing method to extract dynamically the maximum possible feature that is hidden in the face components, instead of dividing face image with static methods such as horizontal or vertical.

References

1. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: a literature survey. *ACM Comput. Surv.* December Issue, 399–458 (2003)
2. Chellappa, R., Wilson, C.L., Sirohey, S.: Human and machine recognition of faces: a survey. *Proc. IEEE.* 83, 705–740 (1995)
3. Heisele, B., Ho, P., Wu, J., Poggio, T.: Face recognition: component-based versus global approaches. *Computer Vision and Image Understanding* 91, 6–21 (2003)
4. OToole, A.J., Abdi, H.: Low-dimensional representation of faces in higher dimensions of the face space. *Opt. Soc. Am.* 10, 405–411 (1993)
5. Fraude, F., De la Torre, F., Gross, R., Baker, S., Kumar, V.: Representational oriented component analysis (ROCA) for face recognition with one sample image per training class. In: *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 266–273 (2005)
6. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 226–239 (1998)
7. Fang, Y., Tan, T., Wang, Y.: Fusion of global and local features for face verification. In: *16th International Conference on Pattern Recognition (ICPR 2002)*, vol. 2, pp. 382–385 (2002)
8. Kim, C., Oh, J., Choi, C.-H.: Combined subspace method using global and local features for face recognition. In: *Proceedings of the International Joint Conference on Neural Networks*, pp. 2030–2035 (2005)
9. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition* 34, 299–314 (2001)
10. Toygar, Ö., Acan, A.: Multiple classifier implementation of a divide-and-conquer approach using appearance-based statistical methods for face recognition. *Pattern Recognition Letters* 25, 1421–1430 (2004)
11. Zhang, L., Tjondronegoro, D.: Selecting, Optimizing and Fusing 'Salient' Gabor Features for Facial Expression Recognition. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) *Neural Information Processing. LNCS*, vol. 5863, pp. 724–732. Springer, Heidelberg (2009)
12. Lian, X.-C., Lu, B.-L.: Gender Classification by Combining Facial and Hair Information. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) *ICONIP 2008. LNCS*, vol. 5507, pp. 647–654. Springer, Heidelberg (2009)
13. Tan, X., Chen, S., Zhou, Z., Zhang, F.: Face recognition from a single image per person: A survey. *Pattern Recognition* 39, 1725–1745 (2006)
14. Huang, J., Yuen, P.C., Lai, J.H., Li, C.: Face recognition using local and global features. *EURASIP Journal on Applied Signal Processing* 2004, 530–541 (2004)

An Information-Spectrum Approach to Analysis of Return Maximization in Reinforcement Learning

Kazunori Iwata

Graduate School of Information Sciences, Hiroshima City University,
Hiroshima 731-3194, Japan
kiwata@hiroshima-cu.ac.jp

Abstract. In reinforcement learning, Markov decision processes are the most popular stochastic sequential decision processes. We frequently assume stationarity or ergodicity, or both to the process for its analysis, but most stochastic sequential decision processes arising in reinforcement learning are in fact, not necessarily Markovian, stationary, or ergodic. In this paper, we give an information-spectrum analysis of return maximization in more general processes than stationary or ergodic Markov decision processes. We also present a class of stochastic sequential decision processes with the necessary condition for return maximization. We provide several examples of best sequences in terms of return maximization in the class.

1 Introduction

Stochastic sequential decision processes (SDPs) in reinforcement learning (RL) [1] are usually formulated as Markov decision processes (MDPs). We often assume stationarity or ergodicity, or both to the MDP for its analysis. This assumption ensures taking the expectation of value function estimates, which is needed for simplicity of analysis on stochastic approximation but also for definition of a stationary optimal policy. Most SDPs appearing in RL applications, however, are in fact not necessarily Markovian, stationary, or ergodic. For example, when an environment varies with time, it is not a stationary process. This leads us to question what conditions a non-Markovian, non-stationary, and/or non-ergodic SDP must satisfy to make return maximization (RM) possible. Basically, RM means that state-action-reward sequences generated during an SDP tend to provide a maximum return as time increases.

In this paper, we formulate RL as a much more general SDP than merely an MDP which is stationary or ergodic, or both. In relation to this, we introduce several information-spectrum quantities to deal with the general SDPs by building a novel bridge between RL and information theory (IT). The aim of the paper is to shed further light on RM in RL, without Markovianity, stationarity, or ergodicity assumptions on the SDPs. As a result, we present a class of SDPs with the necessary condition for RM. We give several examples of best sequences in terms of RM in the class.

The organization of this paper is as follows. We review MDPs and consider more general SDPs in Section 2. The main results are described in Section 3. Finally, we give a summary in Section 4.

2 Preliminaries

We focus on discrete-time SDPs with discrete states, actions, and rewards. For simplicity, we consider an episodic task in this paper. Let \mathbb{N} denote the positive integers. Let \mathbb{R} and \mathbb{R}^+ denote the real numbers and the positive real numbers, respectively.

Markov Decision Process. The sets of states, actions, and rewards of SDPs are denoted as $\mathcal{S} \triangleq \{s_1, s_2, \dots, s_I\}$, $\mathcal{A} \triangleq \{a_1, a_2, \dots, a_J\}$, and $\mathcal{R} \triangleq \{r_1, r_2, \dots, r_K\}$, respectively, where $r_k \in \mathbb{R}$ and $|r_k| < \infty$ for all k . I , J , and $K \in \mathbb{N}$ are the numbers of elements in \mathcal{S} , \mathcal{A} , and \mathcal{R} , respectively. We assume that \mathcal{S} , \mathcal{A} , and \mathcal{R} are non-empty finite sets. Let $s(t)$, $a(t)$, and $r(t)$ be a state, action, and reward, respectively, at time $t \in \mathbb{N}$. MDPs [1, Chap. 3] are the most popular SDPs for describing the framework of interactions between an agent and the environment. Whenever we refer to an MDP, we mean a stochastic process defined by

$$\left(\mathcal{S}, \mathcal{A}, \mathcal{R}, \left\{p_{ij}^{(t)}\right\}, \left\{p_{iji'k}^{(t)}\right\}, \left\{p_i^{(1)}\right\}\right),$$

where for all i, j, i', k ,

$$p_{ij}^{(t)} \triangleq \Pr(a(t) = a_j | s(t) = s_i), \quad (1)$$

$$p_{iji'k}^{(t)} \triangleq \Pr(s(t+1) = s_{i'}, r(t+1) = r_k | s(t) = s_i, a(t) = a_j), \quad (2)$$

$$p_i^{(1)} \triangleq \Pr(s(1) = s_i). \quad (3)$$

Eqs. (1) and (2) are called the action-selection probability and state-transition probability, respectively. The sequence of state, action, and reward is drawn according to their Markov transitions. When the state-transition probabilities of an MDP are not time-dependent, its state-transition is said to be stationary. Also, an agent's policy whose action-selection probabilities are not time-dependent, is said to be stationary. An MDP with stationary state-transition and policy is called a stationary MDP. If the Markov chain determined by a state-transition and policy is both irreducible and non-periodic on \mathcal{S} , we say the MDP is an ergodic MDP.

Action-selection Strategy. The method for describing an agent's policy is called an action-selection strategy. The softmax method [1, Chap. 2] is one of the most popular methods. This method depends on the estimates of action-value functions [1, Chap. 3], each of which indicates the (discounted) sum of rewards to be received in the future. Let Q_{ij} denote the action-value function estimate of state-action $(s_i, a_j) \in \mathcal{S} \times \mathcal{A}$. In this method, using a parameter $\beta \in \mathbb{R}$, Eq. (1) can be written as

$$p_{ij}^{(t)} = \frac{\exp(\beta Q_{ij})}{\sum_{j' \in \mathcal{J}_i} \exp(\beta Q_{ij'})}, \quad (4)$$

where \mathcal{J}_i denotes the set of action indices available in state s_i . Note that only when all the estimates of Q_{ij} and the parameter β are not time-dependent, Eq. (4) is not time-dependent.

In fact, Eq. (4) is time-dependent in most MDPs, because an agent improves its policy by updating the estimates, and the parameter needs to be asymptotically increased to promote the acceptance of actions that may yield a higher return than others. However, when an MDP is analyzed, we often assume stationarity or ergodicity, or both to the MDP. Concretely, we assume that the state-transition is stationary, the MDP is ergodic, and at each time, the future MDP is stochastically the same as the current one. This assumption provides the assurance that the estimates would go to their expectations by a stochastic approximation and also there exists a stationary optimal policy. Indeed, most mathematical analyses of RL and most learning algorithms for updating the estimates employ this assumption [23456789].

More General Process. Now, we consider whether the estimates and parameters referred to in the softmax method are time-dependent and obey their respective stochastic processes. For all i, j , let $Q_{ij}(t)$ denote the action-value function estimate of state-action (s_i, a_j) at time t . The set of action-value function estimates on \mathbb{R}^{IJ} is denoted by

$$Q(t) \triangleq \{ Q_{ij}(t) \mid 1 \leq i \leq I, 1 \leq j \leq J \}. \quad (5)$$

Let $\beta(t)$ denote the parameter β on \mathbb{R} at time t . In this case, an action at time t is drawn according to the action-selection probability expressed as

$$p_{ij}^{(t)}(\beta, Q) \triangleq \Pr(a(t) = a_j \mid \beta(t) = \beta, Q(t) = Q, s(t) = s_i). \quad (6)$$

In the same manner, we consider that the state-transition probability is parameterized with a vector of time-dependent parameters. For example, consider

$$p_{iji'k}^{(t)}(\zeta) \triangleq \Pr(s(t+1) = s_{i'}, r(t+1) = r_k \mid \zeta(t) = \zeta, s(t) = s_i, a(t) = a_j), \quad (7)$$

where $\zeta(t)$ denotes a parameter vector on some space at time t . We can see that $p_{ij}^{(t)}$ and $p_{iji'k}^{(t)}$ are drawn according to a probability measure that defines $\beta(t)$, $Q(t)$, and $\zeta(t)$. Accordingly, in general, an SDP defined by $p_{ij}^{(t)}$ and $p_{iji'k}^{(t)}$ is no longer stationary, ergodic, or Markovian with respect to its action-selection and state-transition. Even if we take another action-selection strategy, this argument holds in a similar way.

Definition 1. For all $t \in \mathbb{N}$, we use $\theta(t)$ to denote the all factors that determine the action-selection and state-transition probabilities of an SDP. Θ denotes the sample space of all possible outcomes of $\theta(t)$. For all $n \in \mathbb{N}$, an outcome of $(\theta(1), \dots, \theta(n))$ on Θ^n is simply denoted as θ .

Example 1. When the action-selection and state-transition probabilities of an SDP are given by (6) and (7), respectively, $\theta(t)$ is represented by $\theta(t) = (Q(t), \beta(t), \zeta(t))$. If the SDP is a stationary MDP, there exists a vector $\bar{\theta} \in \Theta$ such that $\theta(t) = \bar{\theta}$ for all $t \in \mathbb{N}$.

Definition 2. For all $n \in \mathbb{N}$, let $\mathcal{X}^n \triangleq \mathcal{S} \times (\mathcal{A} \times \mathcal{S} \times \mathcal{R})^n$. We denote the three-tuple of state, action, and reward at time t by $x(t) \triangleq (a(t), s(t+1), r(t+1))$ for all $t \in \mathbb{N}$, with the convention $x(0) = s(1)$. Using the notation, the sequence of state, action, and reward, $s(1), a(1), s(2), r(2), a(2), \dots, s(n), r(n), a(n), s(n+1), r(n+1)$, is briefly denoted by $x(0), x(1), \dots, x(n)$, where n is the time length of the SDPs.

Definition 3. Let P_θ^n be a probability measure on \mathcal{X}^n given $\theta \in \Theta^n$, expressed as

$$P_\theta^n(x) \triangleq \Pr((x(0), x(1), \dots, x(n)) = x \mid (\theta(1), \dots, \theta(n)) = \theta), \quad (8)$$

for all $x \in \mathcal{X}^n$. We denote the stochastic variables drawn according to P_θ^n by X_θ^n . For all $x \in \mathcal{X}^n$, the mixed measure of P_θ^n is described as

$$P^n(x) \triangleq \int_{\Theta^n} P_\theta^n(x) dy^n(\theta), \quad (9)$$

where dy^n is a probability measure on Θ^n . We denote the stochastic variables drawn according to P^n by X^n .

An SDP X^n is a stochastic process defined by (\mathcal{X}^n, P^n) . Note that an MDP six-tuple can be rewritten as a two-tuple in this manner. In an episodic task discussed here, the sequence of X^n is observed repeatedly. Analysis of stochastic processes via that of their mixed process is well-established in IT as information-spectrum analysis [10]. In this paper, we examine how SDPs $\{X_\theta^n \mid \theta \in \Theta^n\}$ should be drawn for RM via analysis of their mixed SDP X^n . So, we introduce an information-spectrum quantity of SDPs, called the spectral entropy rate [11][10], to analyze general processes.

Definition 4. The spectral superior and inferior entropy rates of P^n are defined by

$$\overline{H}(P^\infty) \triangleq \inf \left\{ b \in \mathbb{R} \mid \lim_{n \rightarrow \infty} \Pr \left(\frac{1}{n} \log \frac{1}{P^n(X^n)} > b \right) = 0 \right\}, \quad (10)$$

$$\underline{H}(P^\infty) \triangleq \sup \left\{ b \in \mathbb{R} \mid \lim_{n \rightarrow \infty} \Pr \left(\frac{1}{n} \log \frac{1}{P^n(X^n)} < b \right) = 0 \right\}, \quad (11)$$

respectively. The spectral superior and inferior entropy rates of P_θ^n are defined similarly.

3 Main Results

The asymptotic equipartition property (AEP) is fundamental in IT, and there have been many studies on it [2][3][4][5][6][7][8]. In particular, the concept of the AEP was ultimately generalized to arbitrary general processes in [18][10]. In this section, we introduce the AEP into RL, and explain the AEP-based analysis of RM.

Henceforth, whenever we refer to the AEP, we mean the generalized AEP described in the following definition [18][10].

Definition 5 (AEP [18]). For all $n \in \mathbb{N}$, let M^n denote a probability measure on a sample space \mathcal{Y}^n . M^n is said to have the AEP if it satisfies the following: for all $\delta \in \mathbb{R}^+$, as $n \rightarrow \infty$,

$$M^n(B_\delta(M^n)) \rightarrow 0 \quad \text{and} \quad M^n(S_\delta(M^n)) \rightarrow 0, \quad (12)$$

where $B_\delta(M^n)$ and $S_\delta(M^n)$ are defined by

$$B_\delta(M^n) \triangleq \{y \in \mathcal{Y}^n \mid M^n(y) \geq \exp(-(1-\delta)H(M^n))\}, \quad (13)$$

$$S_\delta(M^n) \triangleq \{y \in \mathcal{Y}^n \mid M^n(y) \leq \exp(-(1+\delta)H(M^n))\}, \quad (14)$$

where

$$H(M^n) \triangleq \sum_{x \in \mathcal{Y}^n} M^n(x) \log \frac{1}{M^n(x)}. \quad (15)$$

The AEP implies that under some constraint on M^n , there exists a set of sequences in \mathcal{Y}^n with probability nearly one. We are now in a position to clearly describe RM in SDPs in terms of the AEP.

Definition 6. For all $n \in \mathbb{N}$, let $\mathcal{X}_\epsilon^{*n}$ be the set of best sequences in \mathcal{X}^n . RM means that for all $\epsilon \in \mathbb{R}^+$, $P^n(\mathcal{X}_\epsilon^{*n}) \rightarrow 1$ as $n \rightarrow \infty$.

In fact, the AEP-based analysis discussed below holds for any non-empty set of best sequences. In that sense, arbitrary sequences can be defined as the best sequences, but those in the RL context should be sequences which tend to yield a maximum return as time progresses. We shall show several examples of best sequences later.

Theorem 1 specifies the role of the AEP in RM.

Theorem 1. If

1. P^n has the AEP,
2. for all $\delta, \epsilon \in \mathbb{R}^+$,

$$\lim_{n \rightarrow \infty} \{P^n(\mathcal{X}_\epsilon^{*n} \cup C_\delta(P^n)) - P^n(\mathcal{X}_\epsilon^{*n} \cap C_\delta(P^n))\} = 0, \quad (16)$$

where

$$C_\delta(P^n) \triangleq \{x \in \mathcal{X}^n \mid x \notin B_\delta(P^n), x \notin S_\delta(P^n)\}, \quad (17)$$

then RM holds.

Proof. Since P^n has the AEP, for all $\delta \in \mathbb{R}^+$, Eq. (12) gives

$$\lim_{n \rightarrow \infty} P^n(C_\delta(P^n)) = 1. \quad (18)$$

Since

$$P^n(C_\delta(P^n)) \leq P^n(\mathcal{X}_\epsilon^{*n} \cup C_\delta(P^n)), \quad (19)$$

$$P^n(\mathcal{X}_\epsilon^{*n} \cap C_\delta(P^n)) \leq P^n(\mathcal{X}_\epsilon^{*n}) \leq P^n(\mathcal{X}_\epsilon^{*n} \cup C_\delta(P^n)), \quad (20)$$

Eqs. (18) and (16) give

$$\lim_{n \rightarrow \infty} P^n(\mathcal{X}_\epsilon^{*n} \cup C_\delta(P^n)) = 1, \quad (21)$$

$$\lim_{n \rightarrow \infty} \{P^n(\mathcal{X}_\epsilon^{*n} \cup C_\delta(P^n)) - P^n(\mathcal{X}_\epsilon^{*n})\} = 0, \quad (22)$$

respectively. Therefore, we reach the conclusion.

The set of sequences $C_\delta(P^n)$ is referred to as the typical set. Theorem 1 suggests that if $\mathcal{X}_\epsilon^{*n}$ and $C_\delta(P^n)$ intersect, then $P^n(\mathcal{X}_\epsilon^{*n})$ tends to a positive probability. Intuitively, this means that it is “possible” for the agent to achieve RM. Taking account of the fact that stationary ergodic MDPs have the AEP, Fig. 1 summarizes the relationship among

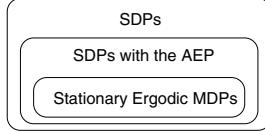


Fig. 1. Relationship among stationary ergodic MDPs, SDPs with the AEP, and SDPs

stationary ergodic MDPs, SDPs with the AEP, and SDPs. The AEP is a necessary condition for RM. Hence, we should note that an SDP arising in RL must be one of the SDPs with the AEP for RM.

In Definition 6, RM is defined in terms of the set of best sequences. In the rest of this section, we describe several examples of best sequences. We start by specifying them for stationary ergodic MDPs using the action-value functions.

Example 2. Assume that X^n is a stationary ergodic MDP. For all $t \in \mathbb{N}$, $s_{X^n}(t)$, $a_{X^n}(t)$ and $r_{X^n}(t)$ denote the stochastic variables of state, action and reward at time t , drawn by X^n . Given $X^n = x \in \mathcal{X}^n$, its types on $s_i \in \mathcal{S}$, $(s_i, a_j) \in \mathcal{S} \times \mathcal{A}$ and $(s_i, a_j, s_{i'}, r_k) \in \mathcal{X}$ [19] are

$$f_i(x) \triangleq \frac{1}{n} \#\{ t \mid s_{X^n}(t) = s_i \}, \quad (23)$$

$$f_{ij}(x) \triangleq \frac{1}{n} \#\{ t \mid (s_{X^n}(t), a_{X^n}(t)) = (s_i, a_j) \}, \quad (24)$$

$$f_{iji'k}(x) \triangleq \frac{1}{n} \#\{ t \mid (s_{X^n}(t), a_{X^n}(t), s_{X^n}(t+1), r_{X^n}(t+1)) = (s_i, a_j, s_{i'}, r_k) \}, \quad (25)$$

respectively, where $\#\{\cdot\}$ denotes the number of elements in a finite set. For all i, j, i', k , we define their conditional types $\hat{p}_{ij}(x)$ and $\hat{p}_{iji'k}(x)$ by $f_{ij}(x) = f_i(x)\hat{p}_{ij}(x)$ and $f_{iji'k}(x) = f_{ij}(x)\hat{p}_{iji'k}(x)$, respectively. Since in this case, the $p_{ij}^{(t)}$ and $p_{iji'k}^{(t)}$ in (1) and (2) are invariant with respect to t , we denote these as p_{ij} and $p_{iji'k}$, respectively, dropping the t . Considering the AEP in stationary ergodic MDPs [89], the best sequences in stationary ergodic MDPs, based on the action-value functions, are described as

$$\mathcal{X}_\epsilon^{*n} \triangleq \{ x \in \mathcal{X}^n \mid D(\hat{p}(x) \| p^* | f(x)) \leq \epsilon \}, \quad (26)$$

for all $\epsilon \in \mathbb{R}^+$, where

$$\begin{aligned} D(\hat{p}(x) \| p^* | f(x)) &\triangleq \sum_{i=1}^I f_i(x) \sum_{j=1}^J \hat{p}_{ij}(x) \log \frac{\hat{p}_{ij}(x)}{p_{ij}^*} \\ &+ \sum_{i=1}^I \sum_{j=1}^J f_{ij}(x) \sum_{i'=1}^I \sum_{k=1}^K \hat{p}_{iji'k}(x) \log \frac{\hat{p}_{iji'k}(x)}{p_{iji'k}}, \end{aligned} \quad (27)$$

and

$$p_{ij}^* \triangleq \begin{cases} 1 & \text{if } j = \operatorname{argmax}_{j' \in \mathcal{J}_i} Q_{ij'}^*, \\ 0 & \text{if } j \neq \operatorname{argmax}_{j' \in \mathcal{J}_i} Q_{ij'}^*, \end{cases} \quad (28)$$

where Q_{ij}^* denotes the action-value function of (s_i, a_j) . Note that any action-value function becomes a fixed real number under stationary ergodic MDPs.

The best sequences in this example are defined by p_{ij}^* which stands for a stationary optimal policy. However, for all i, j , Q_{ij}^* does not always become a real number fixed by P^n under more general SDPs than stationary ergodic MDPs. This implies the fact that in general, an optimal policy is not stationary in such cases. Accordingly, the definition of best sequences in Ex. 2 is no longer applicable to such general SDPs.

Now we consider the relationship between best sequences and optimal policy in more general SDPs. For all $(x, \theta) \in \mathcal{X}^n \times \Theta^n$, let

$$\tilde{P}^n(x, \theta) \triangleq P_\theta^n(x)dy^n(\theta), \quad (29)$$

denote the joint probability measure of action-selection, state-transition and their factors. Let \tilde{P}^* be the joint probability measure given by an optimal policy. Since the agent is able to control only action-selection probabilities (via several elements of the factors), for given state-transition probabilities, the area of \tilde{P}^* obtained by changing the action-selection probabilities is actually restricted over the domain of \tilde{P}^n . For all $x \in \mathcal{X}^n$, we let

$$P^{*n}(x) \triangleq \int_{\Theta^n} \tilde{P}^n(x, \theta). \quad (30)$$

Possible P^{*n} is also restricted over its domain. Corollary 1 gives an example of best sequences defined by the typical set of P^{*n} .

Corollary 1. *If the set of best sequences is given by*

$$\mathcal{X}_\epsilon^{*n} \triangleq C_\epsilon(P^{*n}), \quad (31)$$

and P^n satisfies the following properties:

1. P^n has the AEP,
2. there exist $\kappa \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that for all $n_0 \leq n$,

$$\sum_{x \in \mathcal{X}^n} |P^n(x) - P^{*n}(x)| < \frac{\kappa}{n}, \quad (32)$$

then RM holds.

As space is limited, we omit the proof. This suggests that if the conditions in Corollary 1 are satisfied by a learning algorithm, RM holds in terms of the set of the best sequences defined in (31).

4 Summary

We gave an information-spectrum analysis of RM in RL. Our analysis does not require Markovianity, stationarity, or ergodicity assumptions on the SDPs. The AEP is an important property, because it is a necessary condition for RM in general SDPs. Accordingly, it should be fully taken into account when considering applications of RL or the design thereof. We also showed several examples of best sequences in terms of RM.

Acknowledgments

We wish to thank Profs. Hideaki Sakai and Kazushi Ikeda for their helpful comments on an earlier draft of this work. This work was supported in part by Grants-in-Aid 22700152 for scientific research from the MEXT, Japan.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
2. Watkins, C.J.C.H., Dayan, P.: Technical note: Q-learning. *Machine Learning* 8, 279–292 (1992)
3. Dayan, P.: The convergence of TD(λ) for general λ . *Machine Learning* 8(3), 341–362 (1992)
4. Tsitsiklis, J.N.: Asynchronous stochastic approximation and Q-learning. *Machine Learning* 16(3), 185–202 (1994)
5. Borkar, V.S., Meyn, S.P.: The O.D.E. method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control and Optimization* 38(2), 447–469 (2000)
6. Singh, S., Jaakkola, T., Littman, M.L., Szepesvári, C.: Convergence results for single-step on-policy reinforcement learning algorithms. *Machine Learning* 38(3), 287–308 (2000)
7. Gosavi, A.: Boundedness of iterates in Q-learning. *Syst. & Control Lett.* 55(4), 347–349 (2006)
8. Iwata, K., Ikeda, K., Sakai, H.: The asymptotic equipartition property in reinforcement learning and its relation to return maximization. *Neural Networks* 19(1), 62–75 (2006)
9. Iwata, K., Ikeda, K., Sakai, H.: A statistical property of multiagent learning based on Markov decision process. *IEEE Trans. Neural Networks* 17(4), 829–842 (2006)
10. Han, T.S.: Information-Spectrum Methods in Information Theory. Applications of mathematics, vol. 50. Springer, Berlin (2003)
11. Han, T.S., Verdú, S.: Approximation theory of output statistics. *IEEE Trans. Inf. Theory* 39(3), 752–772 (1993)
12. Shannon, C.E.: A mathematical theory of communication. *Bell. Syst. Tech. J.* 27, 623–656 (1948)
13. McMillan, B.: The basic theorems of information theory. *Annals of Math. Statistics* 24(2), 196–219 (1953)
14. Breiman, L.: The individual ergodic theorem of information theory. *Annals of Math. Statistics* 28(3), 809–811 (1957)
15. Moy, S.C.: Generalizations of Shannon-Mcmillan theorem. *Pacific J. of Math.* 11(2), 705–714 (1961)
16. Kieffer, J.C.: A simple proof of the Moy-Perez generalization of the Shannon-Mcmillan theorem. *Pacific J. of Math.* 51(1), 203–206 (1974)
17. Barron, A.R.: The strong ergodic theorem for densities: Generalized Shannon-Mcmillan-Breiman theorem. *The Annals of Prob.* 13(4), 1292–1303 (1985)
18. Verdú, S., Han, T.S.: The role of the asymptotic equipartition property in noiseless source coding. *IEEE Trans. Inf. Theory* 43(3), 847–857 (1997)
19. Csiszár, I.: The method of types. *IEEE Trans. Inf. Theory* 44(6), 2505–2523 (1998)

Analytical Approach to Noise Effects on Synchronization in a System of Coupled Excitable Elements

Keiji Okumura and Masatoshi Shiino

Department of Physics, Faculty of Science, Tokyo Institute of Technology,
2-12-1 Oh-okayama, Meguro-ku, Tokyo 152-8551, Japan
kokumura@mikan.ap.titech.ac.jp

Abstract. We report relationships between the effects of noise and applied constant currents on the behavior of a system of excitable elements. The analytical approach based on the nonlinear Fokker-Planck equation of a mean-field model allows us to study the effects of noise *without approximations* only by dealing with deterministic nonlinear dynamics. We find the similarity, with respect to the occurrence of oscillations involving subcritical Hopf bifurcations, between the systems of an excitable element with applied constant currents and mean-field coupled excitable elements with noise.

Keywords: Noise induced synchronization, Mean-field model, Nonlinear Fokker-Planck equation, Nonequilibrium phase transitions, Bifurcations, Langevin equations, Stochastic limit cycle.

1 Introduction

The effects of noise on neural systems are one of the great concern for researchers in neurosciences. Among theoretical and computational studies is the influence of noise on excitable systems as well as sustained oscillatory ones. Simulations of coupled Hodgkin-Huxley neurons subjected to noise are one of typical examples to demonstrate its behavior [1]. Numerical solutions of the Fokker-Planck equation of coupled active rotators with excitatory and inhibitory elements have been obtained and analysed [2]. Another approach of Gaussian approximations has also been used to reveal bifurcation structures with changes in noise intensity for investigating a system of coupled active rotators and Fitz-Hugh Nagumo neurons [3,4].

In this paper, we propose a model of coupled excitable elements under the influence of external Langevin noise on the basis of mean-field concept [5,6,7]. Satisfying self-averaging property, the model enables us to describe the time evolution of order parameters of the system *without approximation*. Investigating this dynamic system instead of a set of Langevin equations, we show similarity between noise and applied currents effects on coupled and uncoupled excitable elements from the viewpoint of bifurcation structures.

2 Excitable Element

2.1 Model

First, we consider the effects of applied constant currents on a system of an excitable element. Let us suppose that dynamics of a 2-dimensional system of $z^{(x)}, z^{(y)}$ satisfies the following set of differential equations:

$$\frac{dz^{(x)}}{dt} = -a^{(x)}z^{(x)} + J^{(x)}F^{(x)}(b^{(x,x)}z^{(x)} + b^{(x,y)}z^{(y)}) + I, \quad (1)$$

$$\frac{dz^{(y)}}{dt} = -a^{(y)}z^{(y)} + J^{(y)}F^{(y)}(b^{(y,x)}z^{(x)} + b^{(y,y)}z^{(y)}), \quad (2)$$

where $a^{(\mu)}, b^{(\mu,\nu)}, J^{(\mu)}$ ($\mu = x, y$) are constants, I is the applied current and $F^{(\mu)}(\cdot)$ are coupling functions. To make the system excitable, we specify $F^{(x)}(\cdot)$ and $F^{(y)}(\cdot)$ as nonlinear and linear functions as

$$F^{(x)}(z) = z \exp\left(-\frac{z^2}{2}\right), \quad (3)$$

$$F^{(y)}(z) = z. \quad (4)$$

The model parameter values are $a^{(x)} = 2.5$, $a^{(y)} = 0.0030$, $b^{(x,x)} = 1.5$, $b^{(x,y)} = 0.50$, $b^{(y,x)} = 4.0$, $b^{(y,y)} = 1.0$, $J^{(x)} = 5.0$, $J^{(y)} = -0.0040$. In this situation, $z^{(x)}$ and $z^{(y)}$ correspond to fast and slow variables and the system exhibits excitable properties for the applied constant current $I = -3.0$.

Figure 2 shows the behavior of the system. When the system is perturbed, $z^{(x)}$ ‘gets excited’ to generate a single pulse, while $z^{(x)}$ exhibits periodic pulses with a certain level of applied constant currents I (Fig. 2(a)). We can also easily understand this dynamics in phase plane (Fig. 2(b)). Taking an initial condition right side from the equilibrium point, the trajectory begins its journey to return to the fixed point.

2.2 Bifurcation Diagrams against Applied Constant Currents

To reveal the bifurcation structures with changes in applied constant currents I , we conduct a linear stability analysis (Fig. 2). Let us start with the case $I = -3.0$, where the fixed point of the system is the stable node. With increasing the applied constant current I , this point moves toward origin and its stability changes to the stable spiral, to the unstable spiral, and to the unstable node, yielding stable limit cycles accompanying the subcritical Hopf bifurcation. Note that the fixed points have rotational symmetry as $(z_0^{(x)}, z_0^{(y)}, I) = (-z_0^{(x)}, -z_0^{(y)}, -I)$. In this way, appropriate levels of applied constant currents $|I| < 2.4038$ give rise to oscillatory states, whereas these oscillations disappear in higher or lower amplitude $|I| > 2.4042$. In amplitude $2.4038 < |I| < 2.4042$, the limit cycle and fixed point attractors coexist.

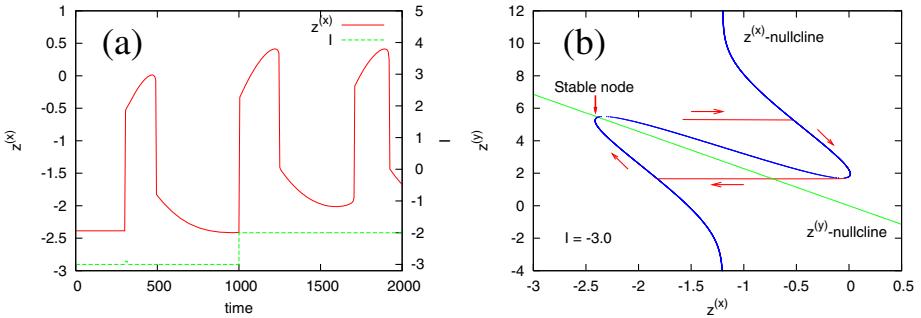


Fig. 1. Dynamics of the system of an excitable element. (a) The time evolution of $z^{(x)}$. $z^{(x)}$ gets excited once by the perturbation at $t = 300.0$. After the changes in amplitude of the applied constant current ($t > 1000.0$), $z^{(x)}$ begins to oscillate. (b) Nullclines in phase plane. At $I = -3.0$, the trajectory converges to the stable node after sufficiently large times.

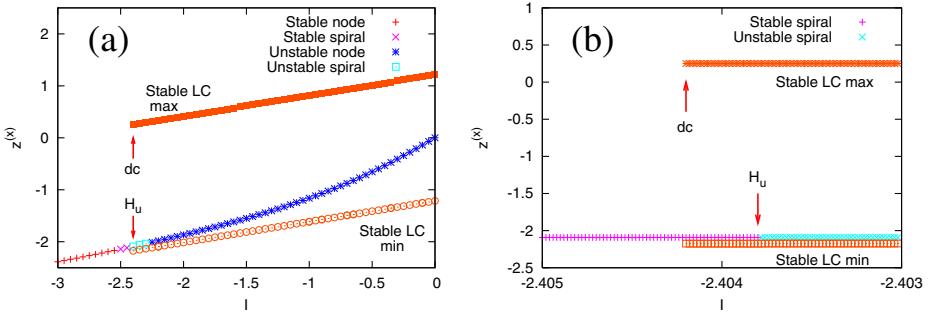


Fig. 2. Bifurcation diagrams of the system of an excitable element. (a) The whole diagram. H_u and dc denote the subcritical Hopf bifurcation and double cycle, respectively. (b) Magnified view of the occurrence of the limit cycle. The limit cycle appears accompanying the subcritical Hopf bifurcation.

3 Coupled Excitable Elements under the Influence of Noise

3.1 Model

To compare the effects of noise and applied constant currents on excitable elements, we extend the system described by Eqs. (1) - (4) to a coupled one. We suppose a system of N excitable elements and assume that they interact with each other via coupling functions $F^{(\mu)}(\cdot)$ with $J^{(\mu)}$ denoting coupling constants. In addition, each element of a system are subjected to the independent Langevin noise $\eta_i^{(\mu)}(t)$. Each of those systems reads

$$\frac{dz_i^{(x)}}{dt} = -a^{(x)}z_i^{(x)} + \frac{1}{N} \sum_{j=1}^N J^{(x)} F^{(x)} (b^{(x,x)} z_j^{(x)} + b^{(x,y)} z_j^{(y)}) + I + \eta_i^{(x)}(t), \quad (5)$$

$$\frac{dz_i^{(y)}}{dt} = -a^{(y)}z_i^{(y)} + \frac{1}{N} \sum_{j=1}^N J^{(y)} F^{(y)} (b^{(y,x)} z_j^{(x)} + b^{(y,y)} z_j^{(y)}) + \eta_i^{(y)}(t), \quad (6)$$

where $i = 1, \dots, N$. We postulate that the Langevin noise $\eta_i^{(\mu)}(t)$ are white Gaussian ones, $\langle \eta_i^{(\mu)}(t) \rangle = 0$, $\langle \eta_i^{(\mu)}(t) \eta_j^{(\nu)}(t') \rangle = 2D^{(\mu)}\delta_{ij}\delta_{\mu\nu}\delta(t-t')$. The coupling functions $F^{(\mu)}(\cdot)$ and the model parameters are the same as described above. The applied constant current is set to $I = -3.0$. In the absence of noise, Eqs. (5) and (6) with $N = 1$ recover Eqs. (1) and (2).

3.2 Nonlinear Fokker-Planck Equation Approach

In the thermodynamic limit $N \rightarrow \infty$, we can take advantage of reducing Eqs. (5) and (6) to a single body equation as seen below. The mean-field coupling terms satisfy the self-average property, which is written by the empirical probability density $P(t, z^{(x)}, z^{(y)})$,

$$\langle F^{(\mu)} \rangle \equiv \int dz^{(x)} dz^{(y)} F^{(\mu)} (b^{(\mu,x)} z^{(x)} + b^{(\mu,y)} z^{(y)}) P(t, z^{(x)}, z^{(y)}). \quad (7)$$

Then the system of Eqs. (5) - (6) is indeed reduced to the 1-body dynamics $z^{(x)}, z^{(y)}$ as

$$\begin{aligned} \frac{dz^{(x)}}{dt} &= -a^{(x)}z^{(x)} + J^{(x)} \langle F^{(x)} \rangle + I + \zeta^{(x)}(t), \\ \frac{dz^{(y)}}{dt} &= -a^{(y)}z^{(y)} + J^{(y)} \langle F^{(y)} \rangle + \zeta^{(y)}(t), \end{aligned}$$

with white Gaussian noise $\zeta^{(\mu)}(t)$, $\langle \zeta^{(\mu)}(t) \rangle = 0$, $\langle \zeta^{(\mu)}(t) \zeta^{(\nu)}(t') \rangle = 2D^{(\mu)}\delta_{\mu\nu}\delta(t-t')$. Thus, one obtains the nonlinear Fokker-Planck equation for the empirical probability density corresponding to the above Langevin equations [8],

$$\begin{aligned} \frac{\partial}{\partial t} P(t, z^{(x)}, z^{(y)}) &= \\ &- \frac{\partial}{\partial z^{(x)}} \left[-a^{(x)}z^{(x)} + J^{(x)} \langle F^{(x)} \rangle + I - D^{(x)} \frac{\partial}{\partial z^{(x)}} \right] P \\ &- \frac{\partial}{\partial z^{(y)}} \left[-a^{(y)}z^{(y)} + J^{(y)} \langle F^{(y)} \rangle - D^{(y)} \frac{\partial}{\partial z^{(y)}} \right] P. \end{aligned} \quad (8)$$

A Gaussian probability density is a special solution of the nonlinear Fokker-Planck equation (8). Furthermore, since the H theorem [5] ensures that the probability density satisfying Eq. (8) converges to the Gaussian-form for sufficiently large times, we are concerned with the Gaussian probability density as

$$P_G(t, z^{(x)}, z^{(y)}) = \frac{1}{2\pi\sqrt{\det C_G(t)}} \exp \left[-\frac{1}{2} \mathbf{s}_G^T C_G^{-1}(t) \mathbf{s}_G \right],$$

$$\mathbf{s}_G^T = (z^{(x)} - \langle z^{(x)} \rangle_G, z^{(y)} - \langle z^{(y)} \rangle_G) \equiv (u^{(x)}, u^{(y)}),$$

$$C_{Gij}(t) = \langle s_i s_j \rangle_G,$$

where $\langle \cdot \rangle_G$ denotes expectation over P_G . Then, the coupling terms Eqs. (7) are described only up to the second moments. We derive a set of closed ordinary differential equations as

$$\frac{d\langle z^{(x)} \rangle_G}{dt} = -a^{(x)} \langle z^{(x)} \rangle_G + J^{(x)} \langle F^{(x)} \rangle_G + I, \quad (9)$$

$$\frac{d\langle z^{(y)} \rangle_G}{dt} = -a^{(y)} \langle z^{(y)} \rangle_G + J^{(y)} \langle F^{(y)} \rangle_G, \quad (10)$$

$$\frac{d\langle u^{(x)^2} \rangle_G}{dt} = -2a^{(x)} \langle u^{(x)^2} \rangle_G + 2D^{(x)}, \quad (11)$$

$$\frac{d\langle u^{(y)^2} \rangle_G}{dt} = -2a^{(y)} \langle u^{(y)^2} \rangle_G + 2D^{(y)}, \quad (12)$$

$$\frac{d\langle u^{(x)} u^{(y)} \rangle_G}{dt} = -(a^{(x)} + a^{(y)}) \langle u^{(x)} u^{(y)} \rangle_G, \quad (13)$$

where one has from Eqs. (3) - (4)

$$\langle F^{(x)} \rangle_G = \frac{m^{(x)}}{(\sigma^2 + 1)^{3/2}} \exp \left[-\frac{m^{(x)^2}}{2(\sigma^2 + 1)} \right], \quad (14)$$

$$\langle F^{(y)} \rangle_G = m^{(y)}, \quad (15)$$

and $m^{(\mu)} = b^{(\mu,x)} \langle z^{(x)} \rangle_G + b^{(\mu,y)} \langle z^{(y)} \rangle_G$, $\sigma^2 = b^{(x,x)^2} \langle u^{(x)^2} \rangle_G + b^{(x,y)^2} \langle u^{(y)^2} \rangle_G$. Note that $\langle u^{(x)} u^{(y)} \rangle_G \rightarrow 0$ and $\langle u^{(\mu)^2} \rangle_G \rightarrow D^{(\mu)}/a^{(\mu)}$ ($t \rightarrow \infty$), implying that the external Langevin noise contributes to the dynamics through the variance. For simplicity, we assume the Langevin noise intensity $D^{(y)} = 0$ in what follows.

Investigating the dynamic system of the order parameter equations written by Eqs. (9)-(13) together with Eqs. (14) and (15), we can compare the effects of noise and applied constant currents on the systems of globally coupled and uncoupled excitable elements.

The dynamics of order parameters of the system is shown in Fig. 3. While the mean value $\langle z^{(x)} \rangle_G$ converges to the stable fixed point for lower intensity of the Langevin noise $D^{(x)}$, it oscillates for relatively higher noise intensity $D^{(x)}$ (Fig. 3 (a)). This oscillatory state suggests that individual excitable elements almost simultaneously get excited under the influence of noise, *i.e.* noise-induced synchronization in coupled excitable elements. The portrait of phase space (Fig. 3 (b)) is similar to the case of uncoupled deterministic system (Fig. 1 (b)).

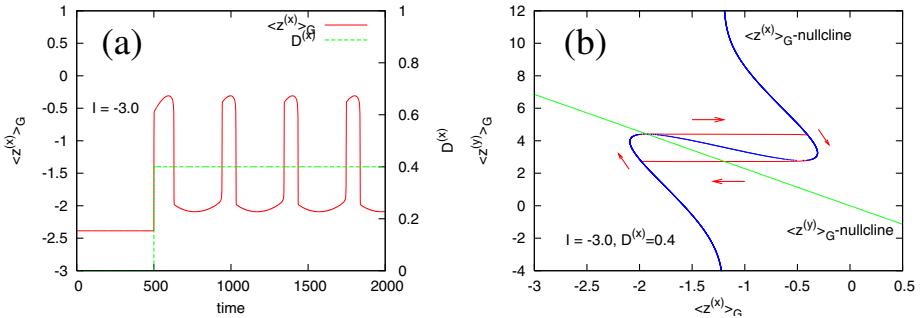


Fig. 3. Dynamics of the system of coupled excitable elements under the influence of noise. (a) The time evolution of $\langle z^{(x)} \rangle_G$. Whereas $z^{(x)}$ does not oscillate for lower noise intensity $D^{(x)}$ ($t < 500.0$), it begins to oscillate for an appropriate noise level ($t > 500.0$). (b) Nullclines in phase plane with the noise intensity $D^{(x)} = 0.4$. In both figures, the applied constant current is set to $I = -3.0$.

3.3 Nonequilibrium Phase Transitions and Synchronization Induced by Langevin Noise

Using the nonlinear Fokker-Planck equation approach with the mean-field model, we can identify the occurrence of genuine bifurcations of order parameters with changes in noise intensity. The result of numerical survey is shown in Fig. 4.

In the deterministic limit $D^{(x)} \rightarrow 0$, the fixed point of the system is a stable node. With increasing $D^{(x)}$, this stability changes to a stable spiral and further to an unstable spiral. A limit cycle attractor occurs accompanying the subcritical Hopf bifurcation (Fig. 4 (b)), similar to the deterministic uncoupled case with changes in applied constant currents (Fig. 2 (b)). In contrast, the disappearance of the oscillatory state is the result of the supercritical Hopf bifurcation (Fig. 4 (c)). So it is clearly seen that a certain level of noise induces synchronized oscillations in the system of mean-field coupled excitable elements under the influence of noise.

4 Summary

We have studied relationships between the roles of noise and applied constant currents in a system of excitable elements. The analytical approach using the nonlinear Fokker-Planck equation associated with the mean-field model allows us to obtain genuine bifurcation diagrams against noise intensity. In a system of an excitable element without noise, applied constant currents of a certain amplitude have given rise to oscillatory states, which originate from the subcritical Hopf bifurcation. The similar structure has been found in coupled excitable elements under the influence of noise.

Further analysis of the effects of the multiplicative noise on a system of mean-field coupled excitable elements will be reported elsewhere.

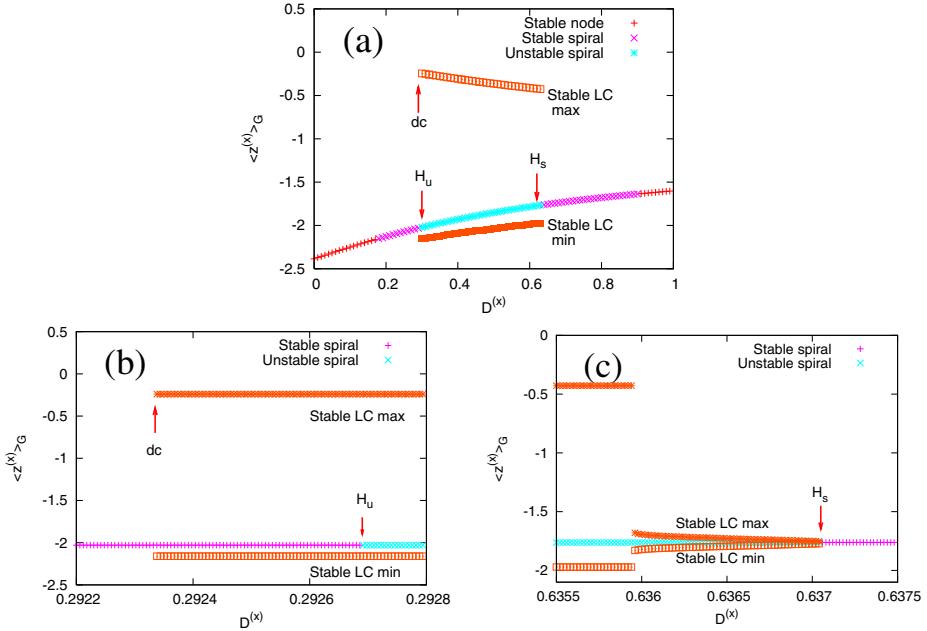


Fig. 4. Bifurcation diagrams of the system of coupled excitable elements under the influence of noise. (a) The whole diagram. H_s and H_u , and dc denote the supercritical and subcritical Hopf bifurcation, and double cycle, respectively. (b) Magnified view of the occurrence of the limit cycle. The limit cycle appears accompanying the subcritical Hopf bifurcation. (c) Magnified view of the disappearance of the limit cycle. The limit cycle vanishes with the supercritical Hopf bifurcation.

K.O. acknowledges the financial support from the Global Center of Excellence Program by MEXT, Japan through the “Nanoscience and Quantum Physics” Project of the Tokyo Institute of Technology.

References

1. Wang, Y., Chik, D.T.W., Wang, Z.D.: Coherence resonance and noise-induced synchronization in globally coupled Hodgkin-Huxley neurons. *Phys. Rev. E* 61, 740–746 (2000)
2. Kanamaru, T., Sekine, M.: Analysis of globally connected active rotators with excitatory and inhibitory connections using the Fokker-Planck equation. *Phys. Rev. E* 67, 31916 (2003)
3. Kurrer, C., Schulten, K.: Noise-induced synchronous neuronal oscillations. *Phys. Rev. E* 51, 6213–6218 (1995)
4. Zaks, M.A., Sailer, X., Schimansky-Geier, L., Neiman, A.B.: Noise induced complexity: From subthreshold oscillations to spiking in coupled excitable systems. *Chaos* 15, 26117 (2005)

5. Shiino, M., Yoshida, K.: Chaos-nonchaos phase transitions induced by external noise in ensembles of nonlinearly coupled oscillators. *Phys. Rev. E* 63, 26210 (2001)
6. Ichiki, A., Ito, H., Shiino, M.: Chaos-nonchaos phase transitions induced by multiplicative noise in ensembles of coupled two-dimensional oscillators. *Physica E* 40, 402–405 (2007)
7. Okumura, K., Ichiki, A., Shiino, M.: Effects of noise on synchronization phenomena exhibited by mean-field coupled limit cycle oscillators with two natural frequencies. *Physica E* (in Press) doi:10.1016/j.physe.2010.07.054
8. Frank, T.D.: Nonlinear Fokker-Planck Equations. Springer, Berlin (2005)

Learning ECOC and Dichotomizers Jointly from Data

Guoqiang Zhong, Kaizhu Huang, and Cheng-Lin Liu

National Laboratory of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences
No. 95 Zhongguancun East Road, Beijing 100190, P.R. China
`{gqzhong, kzhuang, liucl}@nlpr.ia.ac.cn`

Abstract. In this paper, we present a first study which learns the ECOC matrix as well as dichotomizers simultaneously from data; these two steps are usually conducted independently in previous methods. We formulate our learning model as a sequence of concave-convex programming problems and develop an efficient alternative minimization algorithm to solve it. Extensive experiments over eight real data sets and one image analysis problem demonstrate the advantage of our model over other state-of-the-art ECOC methods in multi-class classification.

Keywords: error correcting output codes (ECOC), dichotomizer, concave-convex procedure (CCCP).

1 Introduction

The error correcting output codes (ECOC) technique is a general tool to combine binary classifiers to address multi-class problems [12]. For applying the ECOC framework in different situations, a number of ECOC design methods have been established [345678]. In general, the ECOC framework includes three steps: coding, training dichotomizers, and decoding. More specifically, a coding matrix is first obtained, then different binary classifiers or dichotomizers are constructed based on the class partition according to the coding matrix. Finally, a new sample is assigned to a class using the dichotomizers by the decoding process. In these steps, coding design plays a crucial role; it determines how the classes are partitioned so as to train a number of dichotomizers accordingly. Inappropriate design of the coding matrix may greatly hurt the performance for multi-class classification.

However, previous ECOC methods either specify the coding matrix directly or adapt an initial matrix heuristically according to certain criteria. Pre-definition or heuristic adaptation of the coding matrix may not capture the data property appropriately and hence could degrade the performance of the consequent dichotomizers training and decoding steps. Another limitation of these ECOC methods is that, the ECOC coding step and the dichotomizers training step are always conducted *independently*. Without interaction with the dichotomizers,

the ECOC coding matrix is not optimal for the following classifier construction step and may consequently lead to unsatisfactory decoding results.

To alleviate these two problems, we introduce a new model called JointECOC to learn the ECOC matrix and the dichotomizers jointly from data in this paper. JointECOC is capable of learning an ECOC matrix, which takes into account both the problem domain knowledge (learned from data) and classification performance information (obtained from the dichotomizers). Meanwhile, guided by and interacted with the learned ECOC matrix, a group of dichotomizers can be simultaneously obtained in the process. As far as we know, this is the first study which performs joint learning of the ECOC matrix and dichotomizers from data. We successfully formulate the problem as a non-linear programming problem and develop an alternative method to solve it. More particular, the JointECOC can be efficiently solved alternatively with respect to the ECOC matrix and the parameters of the dichotomizers via a sequence of the concave-convex procedures (CCCP) [9]. Extensive experiments demonstrate the proposed JointECOC's advantages over other state-of-the-art ECOC methods on nine real data sets.

The remainder of this paper is organized as follows. In the next section, we present related work briefly. In Section 3, we introduce the notation used throughout the paper. In Section 4, we detail our JointECOC model including the formulation and the learning algorithm. We evaluate our method against other competitive ECOC methods in Section 5. We conclude this paper with remarks in Section 6.

2 Related Work

A number of ECOC methods have been proposed in the literature. One-versus-all and one-versus-one are two standard ECOC design strategies. However, they are both problem independent, i.e., the coding design is predefined without taking into account the particular problem domain knowledge. Discriminative ECOC (DECOC) [5] is a coding design approach based on finding an optimal binary tree structure of the classes. However, it is a heuristic method. ECOC-optimizing node embedding (ECOC-ONE) [6] is a recently-proposed problem dependent ECOC design method. However, its strategy is based on modifying an initialized ECOC design. It does not learn the coding matrix and the dichotomizers systematically. In [10], Singh-Miller and Collins proposed a method for learning label embeddings to model the similarity between labels within a nearest neighbor framework. The learned label embeddings can be seen as an ECOC matrix codified with continuous values. However, extension of their method to other binary classifiers, such as support vector machines (SVMs) and Adaboost, is very difficult.

Different from the above methods, we do not directly specify the coding matrix as done in one-versus-all and one-versus-one; we also do not restrict the coding matrix as a tree structure, since a tree structure is unnecessarily good for constructing dichotomizers; in addition, we do not try to modify an initial coding matrix, which is somehow ad-hoc and heuristic. Instead, we learn the

ECOC coding matrix systematically from data. The *central contrast* between our model and existing ECOC methods is that we aim to learn the ECOC coding matrix and the dichotomizers jointly. Our model can take into account the problem domain knowledge as well as the classification performance of the dichotomizers. Moreover, our model can be optimized using relatively simple methods and achieve very promising classification accuracy against the previous state-of-the-art ECOC methods.

3 Notation

We use boldface uppercase letters, such as \mathbf{A} , to denote matrices, and boldface lowercase letters, such as \mathbf{v} , to denote vectors. The i th row and j th column of a matrix \mathbf{A} are defined as \mathbf{A}_{i*} and \mathbf{A}_{*j} , respectively. \mathbf{A}_{ij} denotes the element of \mathbf{A} at the i th row and j th column. \mathbf{v}_i is the i th element of a vector \mathbf{v} . We use \mathbf{A}^T to denote the transpose of \mathbf{A} , and tr to denote the trace of \mathbf{A} . $|\mathbf{A}_{cl}|$ is the absolute value of \mathbf{A}_{cl} , and $\|\mathbf{A}\|_F$ is the Frobenius norm of \mathbf{A} .

For multi-class classification problems, we are often given a set of samples $\{\{\mathbf{X}_i, \mathbf{y}_i\} \in \Re^D \times \Re^1, i = 1, 2, \dots, N\}$, where $\mathbf{y}_i \in \{1, 2, \dots, C\}$, $C > 2$ is the number of classes and N is the number of samples. We denote $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^C\}$, where \mathbf{X}^c is the data matrix of class c . \mathbf{X}_{*i}^c is the i th sample of class c . We use N_c to denote the number of samples belonging to class c , and $N = \sum_{c=1}^C N_c$. $[1 - z]_+ = \max\{0, 1 - z\}$ is the hinge loss with respect to z .

For ECOC based models, we use $\mathbf{M} \in \{-1, 0, 1\}^{C \times L}$ to denote the ECOC matrix, where L is the length of codewords. Specifically, each of the L dichotomizers will be constructed according to each column of \mathbf{M} respectively. If $\mathbf{M}_{cl} = 1$ (or -1), then the data samples associated with class c will be treated as the positive (or negative) class for the l th binary classifier. $\mathbf{M}_{cl} = 0$ means the data samples associated with class c will not be used for constructing the l th classifier. We use \mathbf{w}_l and \mathbf{b}_l to denote the coefficients and the bias of the l th linear dichotomizer. We define $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_L\}$, and $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_L\}$. $HD(\mathbf{M}_{*j}, \mathbf{M}_{*k})$ is the Hamming distance between \mathbf{M}_{*j} and \mathbf{M}_{*k} , and $\|\mathbf{M}_{*j} - \mathbf{M}_{*k}\|_2^2$ is the squared Euclidean distance between \mathbf{M}_{*j} and \mathbf{M}_{*k} .

4 Learning ECOC and Dichotomizers Jointly (JointECOC)

In this section, we introduce the formulation and learning algorithm of JointECOC in detail. We will discuss our approach with linear dichotomizers throughout this paper and will not present the case with nonlinear dichotomizers due to the space limitation. However, extension of our model to using nonlinear dichotomizers is straightforward. Moreover, similar to many other studies [8], we adopt linear SVMs as dichotomizers throughout this paper.

Intuitively, we would like to learn an ensemble of L linear dichotomizers, $\{\mathbf{w}_l, \mathbf{b}_l\}_{l=1}^L$, which we expect to obtain good classification performance guided

by the ECOC matrix $\mathbf{M} \in \{-1, 0, 1\}^{C \times L}$. Meanwhile, as each column of \mathbf{M} describes a partition of classes, we require the Hamming distance between any pair of columns of \mathbf{M} is maximized in order to cover significant partitions of the classes sufficiently. Motivated from these observations, we formulate the optimization problem as follows:

$$\begin{aligned} & \min_{\mathbf{M}, \mathbf{W}, \mathbf{b}, \xi} \sum_{l=1}^L \sum_{c=1, \mathbf{M}_{cl} \neq 0}^C \sum_{i=1}^{N_c} \xi_{cli} + \frac{C_1}{2} \text{tr}(\mathbf{W}^T \mathbf{W}) - C_2 \sum_{1 \leq j < k \leq L} HD(\mathbf{M}_{*j}, \mathbf{M}_{*k}), \\ & \text{s.t. } \mathbf{M}_{cl}(\mathbf{w}_l^T \mathbf{X}_{*i}^c + \mathbf{b}_l) \geq 1 - \xi_{cli}, \xi_{cli} \geq 0, \mathbf{M}_{cl} \in \{-1, 0, 1\}, \end{aligned} \quad (1)$$

where C_1, C_2 are penalty factors. Since each element of \mathbf{M} , i.e., \mathbf{M}_{cl} , belongs to $\{-1, 0, 1\}$, Problem (1) is a combinational optimization problem. As proved by Crammer [4], finding the optimal discrete codes for Problem (1) is computationally intractable since it is NP-complete. To address this problem, we can relax $\mathbf{M}_{cl} \in [-1, 1]$ for any $c \in \{1, 2, \dots, C\}$, $l \in \{1, 2, \dots, L\}$ and replace the Hamming distance with Euclidean distance for simplicity. In addition, to force the elements of \mathbf{M} to be close to -1 , 0 or 1 , we introduce two regularization terms in the objective function. Hence, the modified objective function can be written as follows:

$$\begin{aligned} & \min_{\mathbf{M}, \mathbf{W}, \mathbf{b}, \xi} \sum_{l=1}^L \sum_{c=1, \mathbf{M}_{cl} \neq 0}^C \sum_{i=1}^{N_c} \xi_{cli} + \frac{C_1}{2} \text{tr}(\mathbf{W}^T \mathbf{W}) + C_2 \sum_{l=1}^L \sum_{c=1}^C |\mathbf{M}_{cl}| \\ & \quad - C_3 \sum_{1 \leq j < k \leq L} \|\mathbf{M}_{*j} - \mathbf{M}_{*k}\|_2^2 - C_4 \|\mathbf{M}\|_F^2, \\ & \text{s.t. } \mathbf{M}_{cl}(\mathbf{w}_l^T \mathbf{X}_{*i}^c + \mathbf{b}_l) \geq 1 - \xi_{cli}, \xi_{cli} \geq 0, -1 \leq \mathbf{M}_{cl} \leq 1, \end{aligned} \quad (2)$$

where C_1, C_2, C_3, C_4 are positive penalty factors. In this problem, by adding the third term, $C_2 \sum_{l=1}^L \sum_{c=1}^C |\mathbf{M}_{cl}|$, and the fifth term, $-C_4 \|\mathbf{M}\|_F^2$, we hope to make elements of \mathbf{M} to be close to 0 , -1 and 1 .

In addition, if we reformulate the fourth term of Problem (2) as a matrix form, we can obtain the following optimization problem:

$$\begin{aligned} & \min_{\mathbf{M}, \mathbf{W}, \mathbf{b}, \xi} \sum_{l=1}^L \sum_{c=1, \mathbf{M}_{cl} \neq 0}^C \sum_{i=1}^{N_c} \xi_{cli} + \frac{C_1}{2} \text{tr}(\mathbf{W}^T \mathbf{W}) + C_2 \sum_{l=1}^L \sum_{c=1}^C |\mathbf{M}_{cl}| \\ & \quad - C_3 \text{tr}(\mathbf{M}^T \mathbf{Q} \mathbf{M}) - C_4 \|\mathbf{M}\|_F^2, \\ & \text{s.t. } \mathbf{M}_{cl}(\mathbf{w}_l^T \mathbf{X}_{*i}^c + \mathbf{b}_l) \geq 1 - \xi_{cli}, \xi_{cli} \geq 0, -1 \leq \mathbf{M}_{cl} \leq 1, \end{aligned} \quad (3)$$

where \mathbf{Q} is a symmetric positive semi-definite matrix with all the elements as -1 , except that the elements on the diagonal are $C - 1$.

For a given ECOC matrix, $\mathbf{M} \in \{-1, 0, 1\}^{C \times L}$, to optimize \mathbf{W} and \mathbf{b} , only the first two terms in Problem (3) are involved. In this situation, minimizing Problem (3) is equivalent to optimizing L independent linear standard SVMs. Alternatively, when the linear classifiers given by $\{\mathbf{w}_l, \mathbf{b}_l\}_{l=1}^L$ are fixed, in order to optimize \mathbf{M} , only the first one and the last three terms in Problem (3) are

involved. Since the first and the third term in Problem (3) are both convex functions with respect to \mathbf{M} and the fourth and the fifth terms are both concave functions of \mathbf{M} , in this situation, Problem (3) can be seen as a difference of convex functions (DC) problem [11] with respect to \mathbf{M} . Furthermore, if we optimize each column of \mathbf{M} independently, optimizing Problem (3) is equivalent to minimizing L independent DC problems, which can be solved efficiently by the concave-convex procedures (CCCP) [9].

4.1 Practical Solving Method

We now introduce how to use an alternative optimization method to solve Problem (3).

In the first step, we initialize the ECOC matrix, $\mathbf{M} \in \{-1, 0, 1\}^{C \times L}$, randomly, but under the condition that at least one -1 and one 1 in each column of \mathbf{M} . In our experiments, we straightforwardly initialize \mathbf{M} using the one-versus-one strategy.

In the second step, we fix \mathbf{M} to solve \mathbf{W} and \mathbf{b} . The optimization problem can be written as

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{b}, \xi} & \sum_{l=1}^L \sum_{c=1, \mathbf{M}_{cl} \neq 0}^C \sum_{i=1}^{N_c} \xi_{cli} + \frac{C_1}{2} \text{tr}(\mathbf{W}^T \mathbf{W}) \\ \text{s.t. } & \mathbf{M}_{cl}(\mathbf{w}_l^T \mathbf{X}_{*i}^c + \mathbf{b}_l) \geq 1 - \xi_{cli}, \xi_{cli} \geq 0. \end{aligned} \quad (4)$$

This problem can be seen as a summation of L quadratic programming problems. In the viewpoint of ECOC design, it is also equivalent to training L independent linear SVM classifiers. Since the ECOC matrix \mathbf{M} is given, this step is as same as the dichotomizers training step in usual ECOC framework.

In the third step, we learn \mathbf{M} by fixing \mathbf{W} and \mathbf{b} using the CCCP method as below

$$\begin{aligned} \min_{\mathbf{M}, \xi} & \sum_{l=1}^L \sum_{c=1}^C \sum_{i=1}^{N_c} \xi_{cli} + C_2 \sum_{l=1}^L \sum_{c=1}^C |\mathbf{M}_{cl}| - C_3 \text{tr}(\mathbf{M}^T \mathbf{Q} \mathbf{M}) - C_4 \|\mathbf{M}\|_F^2, \\ \text{s.t. } & \mathbf{M}_{cl}(\mathbf{w}_l^T \mathbf{X}_{*i}^c + \mathbf{b}_l) \geq 1 - \xi_{cli}, \xi_{cli} \geq 0, -1 \leq \mathbf{M}_{cl} \leq 1. \end{aligned} \quad (5)$$

This problem is a DC problem with respect to \mathbf{M} , where the first two terms in the objective function are both convex functions with respect to \mathbf{M} , while the last two terms are both concave functions with respect to \mathbf{M} . We can employ the CCCP method to address it. We denote

$$f(\mathbf{M}) = \sum_{l=1}^L \sum_{c=1}^C \sum_{i=1}^{N_c} [1 - \mathbf{M}_{cl}(\mathbf{w}_l^T \mathbf{X}_{*i}^c + \mathbf{b}_l)]_+ + C_2 \sum_{l=1}^L \sum_{c=1}^C |\mathbf{M}_{cl}|, \quad (6)$$

and $g(\mathbf{M}) = C_3 \text{tr}(\mathbf{M}^T \mathbf{Q} \mathbf{M}) + C_4 \|\mathbf{M}\|_F^2$. The CCCP algorithm supposes to optimize

$$\mathcal{L}(\mathbf{M}) = f(\mathbf{M}) - g(\mathbf{M}), \quad (7)$$

with constraints in Problem (5). Given \mathbf{W} and \mathbf{b} , the gradient of $g(\mathbf{M})$ with respect to \mathbf{M} can be calculated as

$$\frac{\partial g(\mathbf{M})}{\partial \mathbf{M}} = 2C_3 \mathbf{Q}\mathbf{M} + 2C_4 \mathbf{M}. \quad (8)$$

As a result, we need solve a convex optimization problem

$$\begin{aligned} \min_{\mathbf{M}} \mathcal{L}_{vex}(\mathbf{M}) &= f(\mathbf{M}) - \text{tr}(\mathbf{M}^T \frac{\partial g(\mathbf{M}^{old})}{\partial \mathbf{M}}) \\ &= f(\mathbf{M}) - \text{tr}(2C_3 \mathbf{M}^T \mathbf{Q} \mathbf{M}^{old} + 2C_4 \mathbf{M}^T \mathbf{M}^{old}) \end{aligned} \quad (9)$$

where \mathbf{M}^{old} is the ECOC matrix obtained in the previous iteration. This is a convex optimization problem and can be solved easily by off-the-shelf techniques.

In the fourth step, we discrete the elements of the learned ECOC matrix \mathbf{M} to be within $\{-1, 0, 1\}$. The rule is written as $\mathbf{M}_{cl} = \begin{cases} -1, & \mathbf{M}_{cl} \leq -0.5 \\ 0, & -0.5 < \mathbf{M}_{cl} < 0.5 \\ 1, & 0.5 \leq \mathbf{M}_{cl} \end{cases}$.

As the elements of \mathbf{M} are relaxed to $[-1, 1]$ in the third step, it is possible that all the elements in a column of \mathbf{M} are larger than 0 (or all less than 0, or all closer to 0). We propose an enhanced method to solve this problem. The basic idea is to enhance the learned ECOC matrix by using the ECOC matrix learned in the previous iteration and the learned dichotomizers in the current iteration. This will be introduced in detail in the next subsection. We solve the above steps iteratively until convergence.

4.2 Enhancing the Learned ECOC Matrix

We assume that the ECOC matrix obtained in the previous iteration conveys some confident domain knowledge to current iteration of learning, and the dichotomizers learned in current iteration contain useful discriminative information of the data. Motivated from these two assumptions, we attempt to enhance the learned ECOC matrix. We first present definitions of two operators.

Definition 1. *Union of two ECOC matrices (\cup_{ECOC}):* Let $\mathbf{M}^1 \in \{-1, 0, 1\}^{C \times L}$ and $\mathbf{M}^2 \in \{-1, 0, 1\}^{C \times L}$ are two ECOC coding matrices. Union of these two ECOC matrices ($\cup_{ECOC}(\mathbf{M}^1, \mathbf{M}^2)$) is a kind of partial ordering mapping: If $\mathbf{M}_{cl}^2 \neq 0$ and $\mathbf{M}_{cl}^1 \neq \mathbf{M}_{cl}^2$, then update \mathbf{M}^1 and let $\mathbf{M}_{cl}^1 = \mathbf{M}_{cl}^2$, for any $c = \{1, 2, \dots, C\}$ and $l = \{1, 2, \dots, L\}$. We call the updated \mathbf{M}^1 as the union of \mathbf{M}^1 and \mathbf{M}^2 .

Definition 2. *Intersection of two ECOC matrices (\cap_{ECOC}):* Let $\mathbf{M}^1 \in \{-1, 0, 1\}^{C \times L}$ and $\mathbf{M}^2 \in \{-1, 0, 1\}^{C \times L}$ are two ECOC coding matrices. Intersection of these two ECOC matrices ($\cap_{ECOC}(\mathbf{M}^1, \mathbf{M}^2)$) is a kind of partial ordering mapping: If $\mathbf{M}_{cl}^1 \neq \mathbf{M}_{cl}^2$, then update \mathbf{M}^1 and let $\mathbf{M}_{cl}^1 = 0$, for any $c = \{1, 2, \dots, C\}$ and $l = \{1, 2, \dots, L\}$. We call the updated \mathbf{M}^1 as the intersection of \mathbf{M}^1 and \mathbf{M}^2 .

Union of two ECOC matrices, $\cup_{ECOC}(\mathbf{M}^1, \mathbf{M}^2)$, is mainly for embedding nonzero elements of \mathbf{M}^2 into \mathbf{M}^1 , and intersection of two ECOC matrices,

$\cap_{ECOC}(\mathbf{M}^1, \mathbf{M}^2)$, is mainly for eliminating those elements of \mathbf{M}^1 , which are conflicted with \mathbf{M}^2 .

Based on these definitions, we can describe the enhanced procedure as below. After the second step of the above iterative process, we obtain L dichotomizers. We test each dichotomizer on the whole training data $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^C\}$ and obtain the classification accuracy, \tilde{a} , for each class labeled by '+1'. Then, we can design an empirical ECOC matrix subjected to the following condition:

$$\widehat{\mathbf{M}}_{cl} = \begin{cases} 1, & \text{if } \tilde{a} > \alpha; \\ -1, & \text{if } 1 - \tilde{a} > \alpha; \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where α is a performance threshold. In this paper, we set it to 0.9 according to our experiment. We then calculate the union of the empirical ECOC matrix $\widehat{\mathbf{M}}$ and the ECOC matrix \mathbf{M}_0 learned in the previous iteration: $\widehat{\mathbf{M}} \Leftarrow \cup_{ECOC}(\widehat{\mathbf{M}}, \mathbf{M}_0)$, where $A \Leftarrow B$ means assigning the value of B to A . After the fourth step of the above iterative process, we obtain a discrete ECOC matrix \mathbf{M} . We can then calculate the union of \mathbf{M} and the ECOC matrix \mathbf{M}_0 learned in the previous iteration: $\mathbf{M} \Leftarrow \cup_{ECOC}(\mathbf{M}, \mathbf{M}_0)$. Then, we calculate the intersection between \mathbf{M} and the empirical ECOC matrix $\widehat{\mathbf{M}}$ as $\mathbf{M} \Leftarrow \cap_{ECOC}(\mathbf{M}, \widehat{\mathbf{M}})$.

5 Experiments

5.1 Multi-class Classification for UCI Data Sets

We first evaluate our model on eight multi-class data sets from the UCI Machine Learning Repository database.¹ They are Dermatology, Ecoli, Iris, Glass, Optdigits, Thyroid, Wine and Yeast, respectively.

We compare the classical one-versus-one, one-versus-all, DECOC [5], and ECOC-ONE [6] methods with our model, JointECOC. For all the methods, linear SVMs (OSU-SVM²) are used as base learners. A linear Loss-Weighted (LW) decoding strategy [8] is adopted for all the coding strategies. The ECOC library toolbox³ is used as the platform for all the algorithms. The CCCP algorithm is coded based on the CVX toolbox.⁴ Results for one-versus-one, one-versus-all, DECOC and ECOC-ONE are taken from [8], as we follow the same setup as [8] to measure the performance of the different ECOC methods. More specifically, we apply the stratified 10-fold cross-validation for implementing JointECOC. The parameters C_1, C_2, C_3, C_4 for JointECOC are fixed to 10, 10, 50, 100 respectively according to our experiments. The JointECOC algorithm is stopped when the current ECOC matrix is equal to its value obtained in the previous iteration.

Row 2 ~ 9 of Table I summarize the classification results obtained over the UCI data sets for each coding design strategy. It is observed that the proposed

¹ <http://archive.ics.uci.edu/ml/>

² <http://svm.sourceforge.net/docs/3.00/api/>

³ <http://ecoclib.svn.sourceforge.net/viewvc/ecoclib/>

⁴ <http://cvxr.com/cvx/>

Table 1. Classification rate and standard deviation on UCI data sets

	one-versus-one	one-versus-all	DECOC	ECOC-ONE	JointECOC
Dermathology	95.59 ± 0.74	95.10 ± 1.01	96.20 ± 0.88	96.31 ± 0.87	97.98 ± 2.74
Iris	97.33 ± 1.07	97.33 ± 1.07	97.33 ± 1.07	97.33 ± 1.07	98.00 ± 4.50
Ecoli	78.64 ± 2.49	68.81 ± 3.42	79.22 ± 2.45	80.41 ± 2.60	85.85 ± 7.22
Wine	93.78 ± 1.76	93.23 ± 1.63	95.55 ± 1.49	95.55 ± 1.49	95.63 ± 5.93
Glass	50.57 ± 3.26	41.36 ± 3.64	54.36 ± 3.16	58.03 ± 3.07	64.43 ± 6.93
Thyroid	94.39 ± 2.15	94.39 ± 2.15	94.39 ± 2.15	94.39 ± 2.15	95.70 ± 6.57
Optdigits	96.23 ± 1.19	91.19 ± 2.36	93.09 ± 2.16	96.77 ± 1.22	96.38 ± 0.85
Yeast	51.18 ± 0.42	17.09 ± 3.48	51.14 ± 2.66	52.10 ± 2.37	57.59 ± 3.93
Satimage	83.39 ± 2.04	75.06 ± 2.19	80.93 ± 1.93	83.33 ± 2.07	85.23 ± 4.91
Mean Rank	3.44	4.89	3.33	2.33	1.22

JointECOC approach demonstrates the overall best performance against other competitive methods.⁵

5.2 Application to Image Analysis

To test JointECOC in applications to image analysis, we carried out experiments on a satellite image data set (satimage), which also belongs to the UCI Machine Learning Repository database. The satimage data set includes 6435 instances within 7 classes and each instance has 36 attributes. The classification results obtained by JointECOC and compared methods are shown in the 10th row of Table I. As we can see, JointECOC outperforms other state-of-the-art methods in this image analysis problem.

We show the mean rank of each ECOC design on all the used data sets in the last row of Table I. The rankings are obtained as follows. We first estimate each particular ranking r_i^j for each problem i and each ECOC coding design j , and then we compute the mean ranking R for each coding strategy as $R_j = \frac{1}{N^*} \sum_i r_i^j$, where $N^* = 9$ is the total number of problems. As can be seen from Table I, JointECOC attains the best position.

6 Conclusion

In this paper, we developed a novel model which is capable of learning the ECOC matrix as well as dichotomizers simultaneously from data. We formulated our learning model as a sequence of concave-convex programming problems and proposed an efficient alternative minimization algorithm to solve it. Extensive experiments over nine real data sets demonstrated that, in most cases, our model outperforms other state-of-the-art coding designs in multi-class classification.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under grant no. 60825301.

⁵ Results for one-versus-one, one-versus-all, DECOC and ECOC-ONE are taken from [8] due to the same experimental setup.

References

1. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. *J. Artif. Intell. Res (JAIR)* 2, 263–286 (1995)
2. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research* 1, 113–141 (2000)
3. Utschick, W., Weichselberger, W.: Stochastic organization of output codes in multiclass learning problems. *Neural Computation* 13(5), 1065–1102 (2001)
4. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. *Machine Learning* 47(2-3), 201–233 (2002)
5. Pujol, O., Radeva, P., Vitrià, J.: Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(6), 1007–1012 (2006)
6. Pujol, O., Escalera, S., Radeva, P.: An incremental node embedding technique for error correcting output codes. *Pattern Recognition* 41(2), 713–725 (2008)
7. Escalera, S., Pujol, O., Radeva, P.: Re-coding ecocs without re-training. *Pattern Recognition Letters* 31(7), 555–562 (2010)
8. Escalera, S., Pujol, O., Radeva, P.: On the decoding process in ternary error-correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(1), 120–134 (2010)
9. Yuille, A.L., Rangarajan, A.: The concave-convex procedure. *Neural Computation* 15(4), 915–936 (2003)
10. Singh-Miller, N., Collins, M.: Learning label embeddings for nearest-neighbor multi-class classification with an application to speech recognition. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C.K.I., Culotta, A. (eds.) *Advances in Neural Information Processing Systems*, vol. 22, pp. 1678–1686 (2009)
11. Horst, R., Thoai, N.V.: DC programming: Overview. *Journal of Optimization Theory and Applications* 103(1), 1–43 (1999)

Wavelet Entropy Measure Based on Matching Pursuit Decomposition and Its Analysis to Heartbeat Intervals

Fausto Lucena^{1,*}, Andre Cavalcante¹, Yoshinori Takeuchi¹,
Allan Kardec Barros², and Noboru Ohnishi¹

¹ Nagoya University, Department of Media Science, Nagoya, Aichi, 464-8603, JPN

² Universidade Federal do Maranhão (UFMA), PIB, São Luís, MA, S/N, BRA

Abstract. Any natural or biological signal can be seen as a linear combination of meaningful and non-meaningful structures. According to the theory of multiresolution wavelet expansions, one can quantify the degree of information those structures using entropy and then select the most meaningful ones. Herein we propose to use adaptive time and frequency transform (ATFT) to measure wavelet entropy, where one line of approach to ATFT is to use a matching pursuit (MP) framework. The proposed method is tested on a set of heartbeat intervals whose population is composed of healthy and pathological subjects. Our results show that wavelet entropy measure based on MP decomposition can capture significant differences between the analyzed cardiac states that are intrinsically related to the structure of the signal.

Keywords: Wavelet entropy, matching pursuit, congestive heart failure, and heart rate variability.

1 Introduction

Choosing what structures should be discarded or maintained during the analysis of a signal is a standard problem in signal processing. To do so, one must comprehend the nature of the signal to infer about the relevance of the structures underlying its pattern. A common strategy to solve this problem has been to exclude those structures that do not add any meaningful information to the signal. Where, the trivial solution employs filter approaches to perform such task. The nuisance is that selecting an adequate filter is not an easy task and inaccurate filters tend to yield spurious representations. An alternative way, however, has been to decompose the signal into several layers according to multiresolution wavelet expansions [1]. It allows one to interpret the signal as a linear combination of a set of wavelet coefficients, similarly to local residual errors between consecutive signal approximations at each layer [2]. The advantage is the possibility of measuring (entropy) information from the wavelet coefficients, which helps to select, discriminate, and organize the most meaningful structures [3]. The drawback is that one needs to select in advance a wavelet package that will decompose the signal through a known time and frequency trade-off.

Perhaps, an optimal solution should consider an adaptive time and frequency decomposition in which the number of layers or “levels” depends on the nature of the signal

* Corresponding author. Email: lucena@ohnishi.m.is.nagoya-u.ac.jp

itself [4], such as described by matching pursuit (MP) algorithms [5]. But, could MP algorithms be used to compute wavelet entropy in a similar fashion that multiresolution wavelet decomposition?

To answer this question, we compare the mathematical framework of wavelet entropy with the adaptive MP decomposition. As a result, we propose that wavelet entropy could be used in a hybrid design with the MP algorithm. Therefore we carry out some experiments with physiological signals to determine how meaningful the values yielded by wavelet entropy derived from MP algorithms are. Specifically, we computed the wavelet entropy (according to the proposed method) using heart rate variability (HRV) of a group composed of 40 healthy and 29 congestive heart failure (CHF) subjects – a cardiopathology associated with the decrease of cardiac output. Our results suggests that the joint wavelet entropy and MP decomposition could be a useful asset to analyze signals whose structures present different degrees of complexity like biological and natural signals.

This paper is organized in seven sections. Sections 2 and 3 give the details about HRV and its dataset. Section 4 gives an overview of the MP decomposition in terms of autonomic cardiac regulation. Section 5 describes the wavelet entropy and draw the connection with the MP algorithm. At last, Results and Discussion are presented on Sections 6 and 7.

2 Heart Rate Variability

The heart rate variability is a straightforward data to access the neuroregulatory control of the heart by deriving discrete event series from electrocardiogram (ECG) signals. The advantages of analyzing the autonomous nervous system (ANS) using HRV are related

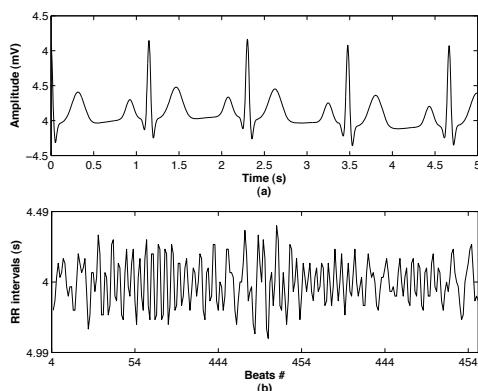


Fig. 1. Illustrative example of RR intervals derived from a synthetic electrocardiogram (ECG). (a) Five seconds of a synthetic ECG with mean zero and standard deviation 0.025 mV. (b) 256 RR intervals derived from the temporal difference between consecutive R-peak waves. The RR intervals were optimized to have a mean of 60 bpm. The most prominent waves represent the R-peaks, such as the one localized at 3.5 seconds.

to the computational simplicity and noninvasive aspects. Fundamentally, it is calculated by extracting the time that marks the precise fiducial time point in the R-waves of the ECG. The successive differences between those points are called RR intervals (see Fig. 1). HRV is obtained from the inverse of n RR intervals in seconds as $\text{HRV} = [\text{RR}_1/60, \text{RR}_2/60, \dots, \text{RR}_n/60]$.

It was found that frequency analysis of short HRV intervals (\approx five minutes) carries important information about the ANS [6]. In fact, the spectral analysis of heartbeat intervals have allowed us to infer about degenerative processes (i.e., Parkinson's and Alzheimer's diseases) and also to draw some patterns involving cardiopathies (i.e., cardiac heart failure and sleep apnea). The ANS can be divided into sympathetic (SNS) and parasympathetic (PNS) nerves. As we shall discuss in the following, they are strongly correlated to the modulatory rhythm of the heart.

The spectral analysis of HRV signals have shown that there are three main frequency bands describing the modulatory control of the ANS. They are divided into high, low, and very-low frequency bands. High frequency, $\text{HF} = \{w | 0.15 < w \leq 0.40 \text{ Hz}\}$, reflects both respiratory sinus arrhythmia (RSA) and vagal modulation (PNS). Low frequency band, $\text{LF} = \{w | 0.03 < w \leq 0.15 \text{ Hz}\}$, represents sympathovagal rhythms. Very low frequency ($\text{VLF} = \{w | 0.00 \leq w \leq 0.03 \text{ Hz}\}$) band remains under ongoing study due to the absence of well-known physiological mechanisms and has been often avoided.

3 The Dataset

The CHF dataset is composed of 29 ECG long-recording signals (24 hours) acquired from patients without any control protocol, whose age ranges from 34 to 79 years old. CHF is basically classified by the New York Heart Association [7] into four different classes, each one expressing how the CHF is evolved in terms of physical activity. In class I, there are neither evident symptoms nor limitations of any kind of physical maneuvers, and the subjects are able to perform simple day-life activities. In class II, the subjects start to have mild indicators of a cardiac disease, such as small resistance to physical activity and difficulty in breathing. In Class III, the symptoms are worse, there are notable physical limitations. The subjects are unable to do less-than-simple physical activities without pain, for example, walk long distances or climb stairs. In class IV, the subjects are incapable of performing any kinds of activities and feel pain even in inactive states. These are bedridden patients.

Herein the database is composed of subjects selected from NYHA classes I, II, and III. The gender of 10 patients is specified (eight man and two woman), but unknown for the remaining. The normal sinus rhythm (NSR) dataset is used as a control group. It is composed of 40 ECG waveforms (two hours) recorded from healthy volunteers during supine resting while watching the movie Fantasia (Disney, 1940). This dataset was divided into two groups: young (21-34 years old) and elderly (68-85 years old). Each group contains the same amount of man and woman. Both CHF and NSR datasets were, receptively, digitalized at 128 Hz and 250 Hz. The beats from each ECG were carefully cataloged through unsupervised systems followed by visual inspection of experts. Both databases were obtained from the Physionet archieves [8].

4 The Matching Pursuit Algorithm

Several models of autonomic cardiac regulation are either based on the analysis of input-output relationship [9][10][11] or the idea of selective frequency extraction [12]. Altogether, they often explore the standard frequency division suggested to analyze the HRV signals [13]. A simple way to accomplish this task is to use the Fourier transform or autoregressive methods (AR). A drawback, however, is that Fourier and AR methods are not robust to nonstationarity. An alternative way has been to use time and frequency transformations to overcome nonstationarity. Essentially, one can relaxe the nonstationarity problem by selecting a function that decomposes a signal into a sequence of basis using adaptive time-frequency transform (ATFT) algorithms. This approach is accomplished by scaling, translating, and modulating versions of the basis function, such that they represent the decomposed signal with a well-defined time and frequency distribution. For instance, ATFT algorithms have drawn a lot of attention in pattern classification [14] and signal compression due to its capacity of reducing a higher dimension space to a few number of parameters. One of the most used ATFT algorithms exploits a matching pursuit decomposition [54]. The MP framework represents a signal $x(t)$ as a linear combination of N basis functions $\phi(t)$ drawn from an overcomplete dictionary $\Phi = [\phi_1, \dots, \phi_M]$ where $M \gg N$, or alternatively

$$x(t) \approx \sum_{n=1}^N c_n \phi_n(t) \quad (1)$$

in which $\phi(t)$ can be Gabor functions described as

$$\phi(t) = Ae^{-\pi(\frac{t-u}{s})^2} \cos(w(t-u) + \varphi), \quad (2)$$

where c_n means modulatory coefficients, s scale, w frequency modulation, u translation, φ phase, and A a normalization factor, such that $\|\phi(t)\| = 1$. Using Gabor functions have several advantages. One may recall that Gabor functions have a compact time-frequency localization and can yield a large variety of shapes.

The MP decomposes $x(t)$ by finding the best orthogonal projections amongst a set of basis functions from a dictionary Φ that matches the structure of $x(t)$. It results in a finite number of basis functions organized in decreasing order of energy. The standard MP algorithm is described as follows

Step 1. Initialize $n = 1$.

Step 2. Define a dictionary Φ of kernels $\phi_j(t)$.

Step 3. Compute $|\langle R^{n-1}x(t), \phi_j(t) \rangle|$ for all $\phi_j(t) \in \Phi$.

Step 4. Find $\phi_n^* = \underset{\phi_j(t)}{\operatorname{argmax}} |\langle R^{n-1}x(t), \phi_j(t) \rangle|$.

Step 5. Compute the new residual,

$$R^n x(t) = x(t) - \sum_{i=1}^n \langle R^{n-1}x(t), \phi_i^*(t) \rangle \phi_i^*(t).$$

Step 6. Remove ϕ_n^* from Φ .

Step 7. If $n = m$ or $\epsilon \leq \text{threshold}$, **stop**;

Otherwise set $n = n + 1$ and go to **Step 3**.

where $R^0x(t) = x(t)$, m is a given iteration number, and ϵ is the energy of residual $R^n x(t)$. If the dictionary is a complete representation of the signal, then

$$x(t) = \sum_{n=1}^{\infty} \langle R^{n-1}x(t), \phi_n(t) \rangle \phi_n(t). \quad (3)$$

One of the intrinsic proprieties of MP algorithm is regarded to how the signal is decomposed [15]. That is, because not all the signals are composed of well-defined (meaningful coherent) components, the MP tends to decompose first coherent underlying structures. And then, break random spike-like noise (non-meaningful) structures into a set of basis functions whose time and frequency distribution are less compact than coherent ones.

5 Wavelet Entropy

Wavelet entropy is a measure based on entropy obtained from wavelet coefficients $C(k)$ at sample time $k = 1, 2, \dots, N$ to quantify the degree of complexity of a signal [2]. This processes divides the desired signal in j levels using multiresolution wavelet expansions [1]. At each level, the energy of the signal $E(k)$ can be expressed as

$$E_j(k) = \sum_{j=1}^J |C_j(k)|^2, \quad (4)$$

with total energy given by

$$E_T = \sum_{j=1}^J \sum_{k=1}^N |C_j(k)|^2 = \sum_j E_j. \quad (5)$$

Note that dividing the energy at level j by its total energy is equivalent to define a probability distribution $p_j = E_j/E_T$ in which $\sum_j p_j = 1$. Using the definition of entropy given by Shannon [16], wavelet entropy H_w is defined as

$$H_w(p) = - \sum_{j=1}^J p_j \log_2(p_j). \quad (6)$$

We have adapted the concept of wavelet entropy measure to the adaptive time-frequency decomposition of the matching pursuit algorithm. That is, at each n -iteration one of the basis function or kernel composing the dictionary decomposes the signal with an energy E_n . If the dictionary is complete, then $E_T = \sum_n E_n$ similarly to (4) and (5).

6 Results

We applied the MP algorithm to decompose HRV intervals derived from CHF patients and NSR volunteers into set of basis functions. Those function are further quantified

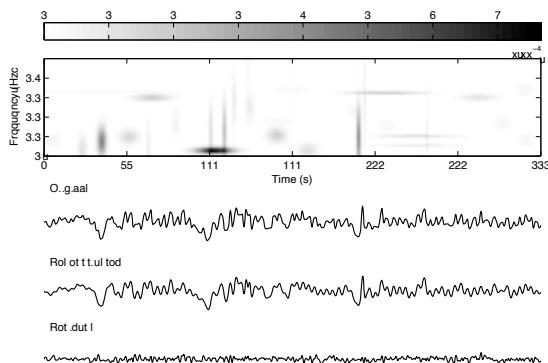


Fig. 2. Matching pursuit decomposition of a HRV signal derived from a healthy volunteer. (*Upper part*) Joint time and frequency plane whose energy levels (“shadow circles”) represent the individual contribution of the basis functions. (*Lower part*) The time series showing the decomposed (original) signal, its reconstructed version using 30 basis functions, and their residual difference.

using wavelet entropy. Each HRV intervals is comprised of 1024 samples and sampled at 4Hz. The decomposition was constraint to cover 95% of the energy of the signal or the process was terminated after consecutive 100 iterations.

Figure 2 illustrates an example of MP decomposition using NSR HRV waveforms and its time-frequency representation. In this example, the original signal was reconstructed using a limited number ($n = 30$) of basis functions for easy visualization. Although the difference between the original signal and its reconstructed version has a prominent structure (see the residual signal), the linear combination of the selected basis functions (reconstructed signal) represents fairly well the original signal. Furthermore, the energy concentration (“shadow circles”) on the joint time and frequency plane illustrates the individual contribution of each basis function. It is evident from the time and frequency distribution that each basis function represents the energy of the original signal fragmented in small parts.

To compute the wavelet entropy, we first reconstruct the signal using 100 basis functions or the its entire set (when the decomposition corresponds to 99.5% of the signal). For each basis basis function and its corresponding reconstructed signal, the wavelet entropy was computed according to 6. Figure 3 illustrates the cumulative sum of the probability distribution (p_j) for HRV signals (NSR and CHF) where $j = 1, 2, \dots, 100$. Observe that the cumulative probability does not follow a uniform distribution. It is worth to mention that the total probability is constraint to have unitary value, meaning that each basis function represents a fixed probability of the reconstructed signal. It is also possible to notice that the cumulative curve for both NSR and CHF signals are very similar. Consequently, it may yield equivalent (wavelet) entropy values as a possible consequence of the chaotic behavior underlying the genesis of HRV signals. But, according to previous studies [17], CHF is characterized by a decrease of energy in HF and increase in VLF and LF band. It suggests that one must analyze the HRV signal using its frequency bands instead the entire frequency distribution. It can be performed by using the central frequency of the basis functions, forming a linear combinations of

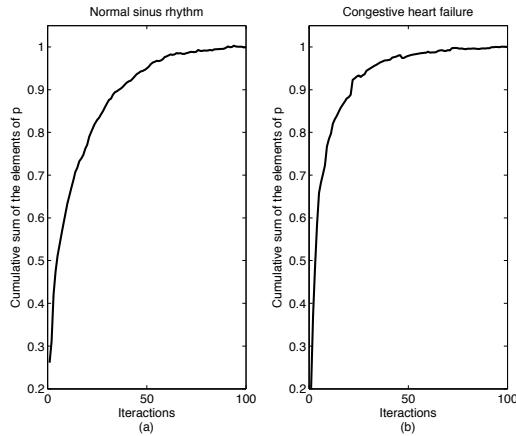


Fig. 3. Cumulative sum of the probability distribution (p_j) for HRV signals (healthy and cardiac pathology) used to compute the wavelet entropy. The probability distribution is obtained through dividing the energy of the basis function used to decompose an analyzed signal by the total energy of the reconstructed signal. Both decompositions are based on 100 basis functions after the same number of iterations.

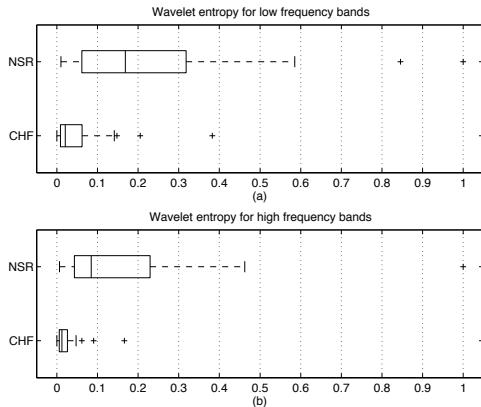


Fig. 4. Box-and-whisker plot of normalized wavelet entropy based on MP decomposition computed for NSR and CHF heartbeat intervals. The wavelet entropy was obtained from the basis functions whose center frequencies lies either on LH or HF bands. The central mark represents the median for 40 volunteers (NSR) and 29 subjects (CHF), the edges of the box are the 25th and 75th percentiles. While, outliers are plotted individually (+), the whiskers are considered not outliers.

the structures underlying LF and HF bands. Figure 4 shows the box-and-whisker plot of the wavelet entropy from the basis functions in terms of its frequency division. It shows that low and high frequency components quantified by wavelet entropy have notable differences. This result is coherent with Costa *et al.* (2002) whose study shown that CHF has lower entropy values than NSR [18].

7 Discussion

A possible argument to explain why the values obtained from wavelet entropy for low and high frequencies yield discriminant features is related to the adaptive MP decomposition. That is, wavelet entropy can only have high entropy values if the probability values are also composed of high values. We have explained that the MP algorithm works by selecting a basis function by projecting it onto an analyzed signal, such that it captures the maximum amount of energy of the signal through the basis function. Thus, one can argue that high frequency structures are more likely to be first decomposed than components with low frequency structures. In average, however, it does not preclude the proposed method, because the probability values are almost the same to both CHF and NSR, as shown in Fig. 3.

Our analysis suggest that wavelet entropy based on MP decomposition is a much more flexible way of capturing the underlying information in segregate frequency bands when compared to the previous method [2]. For instance, once the signal is decomposed, it is possible to reconstruct the signal using the center frequencies of the basis functions to select or to organize the frequency band of interest according to the values given by the wavelet entropy.

The main advantage of the proposed method lies on the adaptive time and frequency transform unlike standard wavelet entropy where *a priori* wavelet package needs to be selected. Further improvement on this work shall emphasize on quantifying the information loss due to the 0.5% energy of the signal (or more) that is not used to estimate the wavelet entropy. And, on using different basis functions to compute wavelet entropy.

References

1. Mallat, S.: A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.* 2, 674–693 (1989)
2. Blanco, S., Figliola, A., Quiroga, R.Q., Rosso, O.A., Serrano, E.: Time-frequency analysis of electroencephalogram series. iii. wavelet packets and information cost function. *Phys. Rev. E* 57, 932–940 (1998)
3. Quiroga, R.Q., Rosso, O.A., Başar, E., Schürmann, M.: Wavelet entropy in event-related potentials: a new method shows ordering of eeg oscillations. *Biol Cybern* 84(4), 291–299 (2001)
4. Mallat, S., Zhang, Z.: Adaptive time-frequency transform. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 241–244 (1993)
5. Mallat, S.G., Zhang, Z.: Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing* 41(12), 3397–3415 (1993)
6. Akselrod, S., Gordon, D., Ubel, F.A., Shannon, D.C., Barger, A.C., Cohen, R.J.: Power spectrum analysis of heart rate fluctuation: A quantitative probe of beat-to-beat cardiovascular control. *Science* 213, 220–222 (1981)
7. The Criteria Committee of the New York Heart Association. In: Nomenclature and Criteria for Diagnosis of Diseases of the Heart and Great Vessels. 9 (edn.), Boston, Mass, pp. 253–256 (1994)
8. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* 101(23), 215–220 (2000)

9. Berger, R.D., Saul, J.P., Cohen, R.J.: Assessment of atonomic response by broad-band respiration. *IEEE Trans. Biomed. Eng.* 36(11), 1061–1065 (1989)
10. Baselli, G., Cerutti, S., Civardi, S., Malliani, A., Pagani, M.: Cardiovascular variability signals: Towards the identification of a closed-loop model of the neural control mechanisms. *IEEE Trans. Biomed. Eng.* 35(12), 1033–1046 (1988)
11. Chon, K.H., Mullen, T.J., Cohen, R.J.: A dual-input nonlinear system analysis of autonomic modulation of the heart rate. *IEEE Trans. Biomed. Eng.* 43(5), 530–544 (1996)
12. Vetter, R., Celka, P., Vesin, J.M., Thonet, G., Pruvot, E., Fromer, M., Scherrer, U., Bernardi, L.: Subband modeling of the human cardiovascular system: New insights into cardiovascular regulation. *Annals of Biomedical Engineering* 26, 293–307 (1998)
13. Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology: Heart rate variability: Standards of measurement, physiological interpretation, and clinical use. *Circulation* 93, 1043–1065 (1996)
14. Akay, M., Mulder, E.: Examining fetal heart-rate variability using matching pursuit. *IEEE Engineering in Medicine and Biology* 15, 64–67 (1996)
15. Umapathy, K., Krishnan, S., Parsa, V., Jamieson, D.G.: Discrimination of pathological voices using a time-frequency approach. *IEEE Trans Biomed Eng* 52(3), 421–430 (2005)
16. Thomas, C.: Elements of Information Theory. Wiley-Interscience, Hoboken (2006)
17. Hadase, M., Azuma, A., Zen, K., Asada, S., Kawasaki, T., Kamitani, T., Kawasaki, S., Sugihara, H., Matsubara, H.: Very low frequency power of heart rate variability is a powerful predictor of clinical prognosis in patients with congestive heart failure. *Circ J.* 68(4), 343–347 (2004)
18. Costa, M., Goldberger, A.L., Peng, C.K.: Multiscale entropy analysis of complex physiologic time series. *Phys. Rev. Lett.* 89(6), 068102 (2002)

Application Rough Sets Theory to Ordinal Scale Data for Discovering Knowledge

Shu-Hsien Liao¹, Yin-Ju Chen², and Pei-Hui Chu³

¹ Department of Management Sciences and Decision Making, Tamkang University,
No. 151, Yingjuan Road, Danshuei Jen, Taipei 251, Taiwan, ROC

² Department of Management Sciences, Tamkang University,
No. 151, Yingjuan Road, Danshuei Jen, Taipei 251, Taiwan, ROC

³ Department of Information Management, National Taipei College of Business,
No.321, Sec. 1, Jinan Rd., Zhongzheng District, Taipei 10051, Taiwan, ROC
michael@mail.tku.edu.tw, s5515124@ms18.hinet.net,
mailsphi@gmail.com

Abstract. Rough set theory has been applied in many areas such as knowledge discovery and has the ability to deal with incomplete, imprecise or inconsistent information. The traditional association rule which should be fixed in order to avoid both that only trivial rules are retained and also that interesting rules are not discarded. In this paper, the new data mining techniques applied to ordinal scale data, which has the ability to handle the uncertainty in the classing process. The aim of the research is to provide a new association rule concept, which is using ordinal scale data.

Keywords: Knowledge discovery, Rough set, Data mining, Association rule.

1 Introduction

Rough set theory has been successfully applied in selecting attributes to improve the effectiveness in deriving decision trees/rules for decisions and classification problems. When decisions involve ordinal classes, the rough set reduction process should try to preserve the order relation generated by the decision classes [6]. And rough set theory, it has found practical applications in many areas such as knowledge discovery, multi-attribute choices, machine learning, approximate classification and data mining [7]. The previous research in mining association rules has two deficiencies. First, it pays no attention to finding rules from ordinal data. Second, it pays no attention to finding rules from imprecise data [4]. Therefore, this study improved [4], and then proposed the concept of algorithm which combines with the rough set theory, so that it can more effectively solve the problem of uncertainty information in ordinal scale data.

The remainder of this paper is organized as follows. Section 2 reviews relevant literature correlate with the research and problem statement. Section 3 Mathematical models for new algorithm. Closing remarks and future work are presented in Sect. 4.

2 Literature Review and Problem Statement

While there have been recent advances in algorithms for clustering categorical data, some are unable to handle uncertainty in the clustering process while others have stability

issues. This research proposes a new algorithm for clustering categorical data, termed Min-Min-Roughness, based on rough set theory, which has the ability to handle the uncertainty in the clustering process [1]. Of the various data mining algorithms, this analysis uses the rough set algorithm due to its ability to deal with incomplete, imprecise or inconsistent information, which is typical in credit assessment analyses [2, 5].

Furthermore, in this research, we incorporate association rules with Rough sets, and promote a novel point of view in applications. In fact, there is no rule for the choice of the “right” connective, so this choice is always arbitrary to some extent.

3 Incorporation of Rough Set for Classification Processing

The traditional association rule which pays no attention to finding rules from ordinal data. Furthermore, in this research, we incorporate association rules with Rough sets, and promote a novel point of view in ordinal scale data applications. The data processing of ordinal scale data is described as below.

3.1 First: Data Processing

Definition 1. Transform the questionnaire answers into information system $IS = (U, A)$, where $U = \{x_1, x_2, \dots, x_n\}$ is a finite set of objects, $A = \{a_1, a_2, \dots, a_m\}$ is a finite set of general attributes/criteria. $f_a = U \times A \rightarrow V_a$ called the information function, V_a is the domain of the attribute/ criterion a , and f_a is a ordinal function set such that $f(x, a) \in V_a$ for each $x_i \in U$.

Example 1. According to Table 1, ranking the non-alcoholic beverages brand recall from the first to eighth by x_1 named Tea, Packaged-waters, Sports, Juice, Soda, Others, Coffee and Energy.

Table 1. Information system

U	A Ordinal Scale Data Sets							
	Tea a_1	Soda a_2	Coffee a_3	Juice a_4	Sports a_5	Packaged-waters a_6	Energy a_7	Others a_8
x_1	1	5	7	4	3	2	8	6
x_2	1	4	3	7	6	5	4	2
x_3	1	7	2	4	6	5	3	8
x_4	1	2	3	5	7	6	4	8
x_5	1	3	6	6	5	4	8	7

Then: $f_{a_1} = \{1\}$ $f_{a_3} = \{2, 3, 6, 7\}$ $f_{a_5} = \{3, 5, 6, 7\}$ $f_{a_7} = \{3, 4, 8\}$
 $f_{a_2} = \{2, 3, 4, 5, 7\}$ $f_{a_4} = \{4, 5, 6, 7\}$ $f_{a_6} = \{2, 4, 5, 6\}$ $f_{a_8} = \{2, 6, 7\}$

Definition 2. According to specific universe of discourse classification, a similarity relation of the general attributes $a \in A$, denoted by $\mathcal{U}/_A$. All of the similarity relation, denoted by $K = (U, R_1, R_2 \cdots R_{m-1})$.

$$U|A = \{\{x_i\}_A | x_i \in U\}$$

Example 2.

$$R(a_3) = \frac{U}{a_3} = \{\{x_1\}, \{x_2, x_4\}, \{x_3\}, \{x_5\}\} \quad R(a_6) = \frac{U}{a_6} = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}\}$$

$$R(a_5) = \frac{U}{a_5} = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}\} \quad R(a_7) = \frac{U}{a_7} = \{\{x_1, x_5\}, \{x_2, x_4\}, \{x_3\}\}$$

Definition 3. The Information system is an ordinal scale data, therefore between the two attributes will have the ordinal response, where B is called the relation between a_i under $\mathcal{U}/_a$ condition.

$$D_a^+ = \left\{ x_i \left| \frac{U}{a}, V_{f_{a_i}} > V_{f_{a_j}} \right. \right\} \quad D_a^- = \left\{ x_i \left| \frac{U}{a}, V_{f_{a_i}} < V_{f_{a_j}} \right. \right\} \quad D_a^0 = \left\{ x_i \left| \frac{U}{a}, V_{f_{a_i}} = V_{f_{a_j}} \right. \right\}$$

When the two random objects both belong to the same fundamental set, $\forall : D_a^+ \equiv D_a^<$ or $\forall : D_a^- \equiv D_a^>$, is mean a core attribute value of ordinal scale data between a_i and a_j . As $V_{f_{a_i}} = V_{f_{a_j}}$, then will be ignored. And $ind(B) = [f_a]_{ind(B)} = \bigcap_{B \in U} [\mathcal{U}/_a]_U$.

Example 3. According to the similarity relation, and then finding that $R(a_5) = \mathcal{U}/_{a_5} = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}\}$ and $R(a_6) = \mathcal{U}/_{a_6} = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}\}$ are both belong to the same fundamental set, and the ranking of $V_{f_{a_5}}$ is place front $V_{f_{a_6}}$, that denoted $D_a^<$, as shown in table2. In other words, Sports and Packaged-waters are both the core attribute value of ordinal scale data of non-alcoholic beverages, and Sports always places after Packaged-waters.

$$ind(B) = [\text{Sports, Packaged - waters}]$$

Table 2. The core attribute value of ordinal scale data of non-alcoholic beverages

R	f_{a_5}	f_{a_6}	D_a
$\{x_1\}$	3	2	D_a^+
$\{x_2, x_3\}$	6	5	D_a^+
$\{x_4\}$	7	6	D_a^+
$\{x_5\}$	5	4	D_a^+

3.2 Second: Generated Rough Associational Rule

Definition 1. The first step in this study, we have found the core attribute value of ordinal scale data, hence in the step, the object is to generated rough associational rule. To consider other attributes into and the core attribute of ordinal scale data as the highest decision-making attributes is hereby to establish the decision table, ease to generate rules, shown as Table 3.

$DT = (U, Q)$, where $U = \{x_1, x_2, \dots, x_n\}$ is a finite set of objects, Q is usually divides into two parts, $G = \{g_1, g_2, \dots, g_m\}$ is a finite set of general attributes/criteria, $D = \{d_1, d_2, \dots, d_l\}$ is a set of decision attributes. $f_g = U \times G \rightarrow V_g$ called the information function, V_g is the domain of the attribute/ criterion g , and f_g is a total function such that $f(x, g) \in V_g$ for each $g \in Q$; $x \in U$. $f_d = U \times D \rightarrow V_d$ called the sorting decision-making information function, V_d is the domain of the decision attributes / criterion d , and f_d is a total function such that $f(x, d) \in V_d$ for each $d \in Q$; $x \in U$.

Table 3. Decision-making table of the drinking habits of "non-alcoholic beverages"

U	Q	General attributes				Decision attributes	
		Product Features g_1	Product Information Source g_2	Consumer Behavior g_3	Channels g_4	Rank	Products
x_1	Price	Seen on shelves	purchase by promotions	Convenience Stores	3	Sports	
x_2	Price	Advertising	purchase by promotions	Hypermarkets	6	Sports	
x_3	Brand	Seen on shelves	will not purchase by promotions	Convenience Stores	6	Sports	
x_4	Brand	Seen on shelves	will not purchase by promotions	Convenience Stores	7	Sports	
x_5	Price	Advertising	purchase by promotions	Hypermarkets	5	Sports	

Then: $f_{g_1} = \{\text{Price, Brand}\}$ $f_{g_2} = \{\text{Seen on shelves, Advertising}\}$
 $f_{g_3} = \{\text{purchase by promotions, will not purchase by promotions}\}$
 $f_{g_4} = \{\text{Convenience Stores, Hypermarkets}\}$

Definition 2. According to specific universe of discourse classification, a similarity relation of the general attributes, denoted by \mathcal{U}_G . All of the similarity relation, denoted by $K = (U, R_1, R_2 \dots R_{m-1})$.

$$U|G = \{\{x_i\}_G | x_i \in U\}$$

Example 2.

$$\begin{array}{ll} R_1 = \frac{U}{g_1} = \{\{x_1, x_2, x_5\}, \{x_3, x_4\}\} & R_6 = \frac{U}{g_2 g_4} = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\} \\ \vdots & \vdots \\ R_5 = \frac{U}{g_1 g_3} = \{\{x_1, x_2, x_5\}, \{x_3, x_4\}\} & R_{m-1} = \frac{U}{G} = \{\{x_1\}, \{x_2, x_5\}, \{x_3, x_4\}\} \end{array}$$

Definition 3. According to the similarity relation, and then finding the reduct and core. The attribute g which were ignored form G and the set G will not affect, thereby g is the unnecessary attribute, we can reduct it. $R \subseteq G$ and $\forall_g \in R$. A similarity relation of the general attributes from decision table, denoted by $ind(G)$. If $ind(G) = ind(G - g_1)$ then g_1 is the reduct attribute, and if $ind(G) \neq ind(G - g_1)$ then g_1 is the core attribute.

$$\text{Example: } U|ind(G) = \{\{x_1\}, \{x_2, x_5\}, \{x_3, x_4\}\}$$

$$U|ind(G - g_1) = U|(\{g_2, g_3, g_4\}) = \{\{x_1\}, \{x_2, x_5\}, \{x_3, x_4\}\} = U|ind(G)$$

$$U|ind(G - g_1 g_3) = U|(\{g_2, g_4\}) = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\} \neq U|ind(G)$$

When considers g_1 alone, g_1 is the reduct attribute, but when considers g_1 and g_3 simultaneously, g_1 and g_3 are the core attributes. A similarity relation and the relational attribute value as shown in Table 4.

Definition 4. The lower approximation, denoted as $\underline{G}(X)$, is defined as the union of all these elementary sets which are contained in $[x_i]_G$. More formally:

$$\underline{G}(X) = \bigcup \left\{ [x_i]_G \in \frac{U}{G} \mid [x_i]_G \subseteq X \right\}$$

The upper approximation, denoted as $\overline{G}(X)$, is the union of these elementary sets, which have a non-empty intersection with $[x_i]_G$. More formally:

$$\overline{G}(X) = \bigcup \left\{ [x_i]_G \subseteq \frac{U}{G} \mid [x_i]_G \cap X \neq \emptyset \right\}$$

The difference: $Bn_G(X) = \overline{G}(X) - \underline{G}(X)$ is called a boundary of $[x_i]_G$.

Example 4. $\{x_1, x_2, x_4\}$ are those customers that we are interested in, thereby $\underline{G}(X) = \{x_1\}$, $\overline{G}(X) = \{x_1, x_2, x_3, x_4, x_5\}$ and $Bn_G(X) = \{x_2, x_3, x_4, x_5\}$.

Definition 5. Using the traditional association rule to calculate the value of Support and Confidence, the formula is shown as follows:

$$Sup(ind(B)) = \left| \left[ind(B) | \underline{G}(X) \subseteq \overline{G}(X) \right] \right| = \left| \frac{ind(B) | \underline{G}(X)}{\overline{G}(X)} \right|$$

$$Conf(ind(B) \rightarrow d_{g_m}) = \left| \left[ind(B) \cap d_{g_m} | Sup(ind(B)) \right] \right| = \left| \frac{Sup(ind(B) \cap d_{g_m})}{Sup(ind(B))} \right|$$

Definition 6. Rough set-based association rules.

$$\frac{\{x_1\}}{g_1 g_3} : g_{1_1} \cap g_{3_1} \Rightarrow d_{d_1}^1 = 4 \quad \frac{\{x_1\}}{g_1 g_2 g_3 g_4} : g_{1_1} \cap g_{2_1} \cap g_{3_1} \cap g_{4_1} \Rightarrow d_{d_1}^1 = 4$$

Table 4. A similarity relation and the relational attribute value

R	Product Features g_1	Product Information Source g_2	Consumer Behavior g_3	Channels g_4	Decision attributes D (Sports)
$\{x_1\}$	Price	Seen on shelves	Purchase by promotions	Convenience Stores	Third $d_{a_5}^1 = 3$
$\{x_2, x_5\}$	Price	Advertising	Purchase by promotions	Hypermarkets	Sixth $d_{a_5}^2 = 6$
					Fifth $d_{a_5}^3 = 5$
$\{x_3, x_4\}$	Brand	Seen on shelves	will not purchase by promotions	Convenience Stores	Sixth $d_{a_5}^3 = 6$
					Seventh $d_{a_5}^4 = 7$
$\{x_1, x_2, x_5\}$	Price	Seen on shelves		Convenience Stores	Third $d_{a_5}^1 = 3$
		Advertising	Purchase by promotions	Hypermarkets	Sixth $d_{a_5}^2 = 6$
		Advertising		Hypermarkets	Fifth $d_{a_5}^5 = 5$
$\{x_1, x_3, x_4\}$	Price		Purchase by promotions		Third $d_{a_5}^1 = 3$
		Brand	will not purchase by promotions	Convenience Stores	Sixth $d_{a_5}^3 = 6$
			will not purchase by promotions		Seventh $d_{a_5}^4 = 7$

Algorithm-Step1

Input:

Information System (IS);

Output:

{Core Attributes};

Method:

1. Begin

2. $IS = (U, A)$;

3. $x_1, x_2, \dots, x_n \in U$; /* where x_1, x_2, \dots, x_n are the objects of set U */

```

4.    $a_1, a_2, \dots, a_m \in A$ ; /* where  $a_1, a_2, \dots, a_m$  are the elements of
      set  $A$  */
5.   For each  $a_m$  do;
6.     compute  $f(x, a)$ ; /* compute the information function
      in IS as described in definition1*/
7.     compute  $R(a_m)$ ; /* compute the similarity relation
      in IS as described in definition2*/
8.     compute  $D_a$ ; /* compute the  $V_a$  as condition
      attributes in  $R(a_m)$  as described in definition3*/
9.   Endfor;
10. Output {Core Attributes};
11. End;

```

Algorithm-Step2

Input:
 Decision Table (DT);
 Output:
 {Classification Rules};
 Method:
 1. Begin
 2. $DT = (U, Q)$;
 3. $x_1, x_2, \dots, x_n \in U$; /* where x_1, x_2, \dots, x_n are the objects of
 set U */
 4. $Q = (G, D)$;
 5. $g_1, g_2, \dots, g_m \in G$; /* where g_1, g_2, \dots, g_m are the elements
 of set G */
 6. $d_1, d_2, \dots, d_l \in D$; /* where d_1, d_2, \dots, d_l are the "core
 attributes" generated in Step1*/
 7. For each d_l do;
 8. compute $f(x, g)$; /* compute the information function
 in DT as described in definition1*/
 9. compute R_m ; /* compute the similarity relation in
 DT as described in definition2*/
 10. compute $ind(G)$; /* compute the relative reduct of DT
 as described in definition3*/
 11. compute $ind(G - g_m)$; /* compute the relative reduct
 of the elements for element m as described
 in definition3*/
 12. compute $\underline{G}(X)$; /* compute the lower-approximation of
 DT as described in definition4*/
 13. compute $\bar{G}(X)$; /* compute the upper-approximation of
 DT as described in definition4*/
 14. compute $Bn_G(X)$; /* compute the bound of DT as
 described in definition4*/

```

15. compute  $Sup(ind(B))$ ; /* compute the support as
   described in definition5*/
16. compute  $conf(ind(B) \rightarrow d_{g_m})$ ; /* compute the confidence
   as described in definition5*/
17. Endfor;
18. Output {Classification Rules};
19. End;

```

4 Conclusion and Future Works

The ordinal attributes which commonly occur in decision making problems, therefore in the research, we provide a new association rule concept, which is using ordinal scale data. Market segmentation is defined as a marketing technique that targets a group of customers with specific characteristics, and pursues the growth opportunities of further market. Every decision algorithm reveals some well-known probabilistic properties; in particular it satisfies the total probability theorem and Bayes' theorem. These properties give a new method of drawing conclusions from data, without referring to prior and posterior probabilities, inherently associated with Bayesian reasoning [3]. So, in the future, we try to incorporate the new association rules with Bayesian network, and promote a novel point of view in applications. The traditional association rules, the user must be trial and error for the association rules issued by explanatory power. The new association rule algorithm which try to combination with rough set theory to provide a more easily explained rules for user. For the convenience of users, to design a expert support system will help to improve the efficiency of the user.

Acknowledgements. This research was funded by the National Science Council, Taiwan, Republic of China, under contract No. NSC 98-2410-H-032-038-MY2.

References

1. Parmar, D., Wu, T., Blackhurst, J.: MMR: An algorithm for clustering categorical data using Rough Set Theory. *Data & Knowledge Engineering* 63, 879–893 (2007)
2. Liu, G., Zhu, Y.: Credit Assessment of Contractors: A Rough Set Method. *Tsinghua Science & Technology* 11, 357–363 (2006)
3. Pawlak, Z.: Rough sets, decision algorithms and Bayes' theorem. *European Journal of Operational Research* 136, 181–189 (2002)
4. Chen, Y.-L., Weng, C.-H.: Mining association rules from imprecise ordinal data. *Fuzzy Sets and Systems* 159, 460–474 (2008)
5. Jiayuan, G., Chankong, V.: Rough set-based approach to rule generation and rule induction. *International Journal of General Systems* 31, 601–617 (2002)
6. John, W.T., Lee, D.S., Yeung, E.C.C.: Tsang Rough sets and ordinal reducts. *Soft Computing - A Fusion of Foundations, Methodologies and Applications* 10, 27–33 (2006)
7. Khoo, L.P., Tor, S.B., Li, J.R.: A Rough Set Approach to the Ordering of Basic Events in a Fault Tree for Fault Diagnosis. *The International Journal of Advanced Manufacturing Technology*, 769–774 (2005)

Dynamic Population Variation Genetic Programming with Kalman Operator for Power System Load Modeling

Yanyun Tao, Minglu Li, and Jian Cao

School of electronic information and electrical engineering,
Shanghai JiaoTong University, Dongchuan road 800,
200240, Shanghai, China
tyy8219@163.com, tyy8219@sjtu.edu.cn

Abstract. According to the high accuracy of load model in power system, a novel dynamic population variation genetic programming with Kalman operator for load model in power system is proposed. First, an evolution load model called initial model in power system evolved by dynamic variation population genetic programming is obtained which has higher accuracy than traditional models. Second, parameters in initial model are optimized by Kalman operator for higher accuracy and an optimization model is obtained. Experiments are used to illustrate that evolved model has higher accuracy 4.6~48% than traditional models and It is also proved the performance of evolved model is prior to RBF network. Furthermore, the optimization model has higher accuracy 7.69~81.3% than evolved model.

Keywords: load modeling, genetic programming, kalman algorithm, dynamic population variation, optimization model.

1 Introduction

Load modeling in power system is of vital importance to power flow calculation, voltage stability, long-term dynamic processing analysis and short-term forecasting [1]. First, the unbalance between the power generation and the power demand reduces the systems security margins and increases the risks to safe operation. Second, the constant model is usually used as the power system loads, which is inadequate for some studies like power system dynamic studies and voltage collapse studies [2]. In order to reduce risks and satisfy power system studies, system studies have to be developed with better models for the systems components including better load models [3].

There are many static load model applied in power system such as polynomial model, power function model, exponential model and hybrid method[4]. In recent years, the intelligence algorithm such as Genetic Algorithm(GA)、Fuzzy approach and Neural Network have been applied in load modeling of power system[5,6]. The interpolation and extrapolation in power system dynamic load modeling was introduced in [7] and it pointed out that Artificial Neural Network (ANN) models have greater nonlinear interpolation and extrapolation than the DE models. A hybrid technique of the fuzzy system and neural network was proposed for approximating continuous

function and several fuzzy controllers implement one of the typical neural networks was pointed out [8]. Baranyi proposed a novel interpolation methodology based on the interpolation of relations instead of interpolating α -cut distances and all the algorithms extended to multivariable cases to improve the fuzzy interpolation [9]. A load forecasting model based on three layers neural networks was proposed in [10] and this technique can also be applied in load modeling. A methodology based on the Recurrent RBF Neural Network (RRBFNN) on modeling load from field measurements, which consists of recurrent network (RNN) and radial basic function (RBF) network and uses the ability of RNN for learning time series and the property of RBF with self-structuring and fast convergence [11].

Load modeling herein has been carried out based on the method that measures the voltage and frequency sensitivity of the active and reactive powers at substations [12]. Once the structure of the load model is determined, the method for improving the accuracy of model by optimizing parameters is feasible. According to this consideration, a hybrid method which combines dynamic population variation genetic programming (DPVGP) with Kalman operator (KO) is proposed. The proposed method works with two stages, evolving the initial model using DPVGP and optimizing the model using KO. The initial model evolved by DPVGP will have a higher accuracy than conventional model, which can excellently represent the characteristic of power system load model. It was proved that the KO can perform well in parameters optimization [13,14] and the optimization load model for power system load can be obtained after the processing of KO. One can use optimization model calculated by proposed methodology in order to determine, for example, the active and the reactive power injections of a load bus.

2 The Proposed Methodology for Load Modeling

By far, all the conventional load models applied in power system can not describe perfectly the power system because of its complicated time variation. The system models evolved by Genetic Programming (GP) are the complicated non-linear system that can't be obtained by human brain and the factor of time and state could not be taken into consideration. A novel dynamic population variation genetic programming (DPVGP) with Kalman operator (KO) is proposed for load modeling in power system, which works with two stages.

Different approaches called DPVGP have been proposed in many papers, However , the pivot function used in these methods are respectively subtraction(SUB), division(DIV), and gradient(GRAD), In our study, we uses the exponent function (EXP) as the pivot function of DPVGP. The new definition of the stagnation phase and novel measure characteristic in DPVGP are given here. Kalman is an exist method, the hybrid of the DPVGP and Kalman has not been proposed and reported.

In the first stage, the initial model evolved by genetic programming is the non-mechanism model which is similar as a black box. This evolutionary model can satisfy the input/output combination. However, the genetic programming uses float constant which only are constrained in a fixed interval. Its search space is limited due to

the float constant and the accuracy of the model evolved by GP system can be improved by optimizing the parameters. In the second stage, the parameters of the evolved load model are used as the state vector and optimized by Kalman filtering algorithm which is defined as Kalman operator.

Lemma 1. The search space of the new methodology for system modeling is larger than the GP system. There is the case : $m > n$, where m and n respectively represent the number of the element in $V^4(DPVGP-KO)$ and $V^4(GP)$, $V(\bullet)$ indicates the search space for the solution of the algorithm.

Proof. The search space of GP system depends on the interval of the float random constant (r), the number of the function set (f), the maximum depth of tree (h) and the maximum generation (g). It can be considered as a linear vector space: $V^4(GP) = \{\alpha_1, \alpha_2 \dots \alpha_k\}, \alpha_i = (r, h, f, g)^T$. Let k is the number of the element in $V^4(DPVGP)$. Although the size of the population in $DPVGP$ changes at each generation, $DPVGP$ has the same number of elements in search space as GP and there is the case: $k = n$. When the Kalman operator is added to optimize the parameters of the evolutionary model, the interval r' in $(r', h, f, g)^T$ is larger than r ($r' > r$). The search space $V^4(DPVGP-KO)$ is considerably expanded and the number m of the elements in $V^4(DPVGP-KO)$ is larger than n of $V^4(GP)$.

2.1 Genetic Programming

The population of the programs in GP system consists of Lisp tree and its search space relies on the terminal set (TS) and function set (FS). In power system, the elements such as Voltage (U), frequency (f) can be used to compose the terminals set of GP system. For instance, $T=\{U, f, Uf, R, 1, 2, 3\}$, where R is the random constant of float type, $R \in [-5, 5]$. “1”, “2” and “3” in TS is the integer constant. Traditional mathematic functions compose the function set of the GP. For example, $F=\{+, -, \times, /\}$. The active power (P) and inactive power (Q) are the outputs of the load model in power system.

GP inherited from Genetic algorithm and has the same evolution mechanism which includes selection operator, reproduction operator, crossover operator (P_c) and mutation operator (P_m). Selection operator applied in GP system uses the roulette wheel combined with elite strategy. This approach has been proved effective in many areas. Crossover operator uses two-point crossover. Assumed that there are two individuals, $Tree_1(x_1, x_2, \dots x_n)$ and $Tree_2(x_1, x_2, \dots x_m)$, $n \neq m$, the depth of these two trees are different and the crossover points are selected by equal probability. Let i and j are the crossover points, the new individuals produced after crossover are $newTree_1(x_1, x_2, \dots x_i, x_{j+1} \dots x_m)$ and $newTree_2(x_1, x_2, \dots x_j, x_{i+1} \dots x_n)$. Mutation operator uses multi-points mutation. We assume that a individual with tree structure is $Tree_1(x_1, x_2, \dots x_n)$ and the mutation points are selected by equal probability. If i, j and k are set as the mutation points, the mode of the point mutation can be expressed as: $x_2 \rightarrow c_2$, then the new individual after mutation is $newTree_1(x_1, x_2, \dots c_i, c_j, \dots c_k, \dots x_n)$.

Fitness function has efficiency of the population evolution. The least square method (LSM) is applied in design of fitness function. , the fitness function $f(X, Y, \varphi)$ can be defined as (1):

$$f(X, Y, \varphi) = M / \sum_{i=1}^m \varepsilon^2 = \frac{M}{(Y - X\varphi)^T (Y - X\varphi)} \quad (1)$$

Where Y is m dimension observation vector: $Y = [y_1, y_2, \dots, y_m]^T$, which represents the outputs data observed for m times. X is an $m \times n$ matrix that indicates the sample data of n independent items for m times:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

x_{ij} means the sample data of the i -th independent items at j -th time. φ is a n dimension vector, which means the parameters of the independent items $\varphi = [\varphi_1, \varphi_2, \dots, \varphi_n]$. $\varepsilon = Y - X\varphi$, which means the error between the measured data and the calculation data by load evolution model. M is a constant and can be set to 1000. Larger the fitness value $f(X, Y, \varphi)$ is, the higher the accuracy of the model is.

2.2 Dynamic Population Variation

First, a stagnation phase in GP system should been defined for dynamic population variation controller (DPVC). Some researches illustrate that the stagnation stage should depend on its best fitness or the mean of high performing individuals [15, 16]. In our study, the stagnation phase in GP system depend on both the best performing individual and the average performing of high individuals. When the high performing individuals and the best individual during the run at generation g both don't have any progress, the state of GP system can be considered to be entering into the stagnation phase.

The general delta function is the between the characteristic measure in previous generation and that in next generation, which is used as the assessment on the stagnation phase in GP system. According to definition in paper [16], the general delta function Δ can be expressed by (2) as follow:

$$\Delta_g = \phi_g - \phi_{g-1} \quad (2)$$

In our study, according to the new stagnation phase definition proposed above, the new characteristic measure can be defined as (3):

$$\phi_g = \alpha \cdot \sum_{HP} \frac{f_g}{S_{HP}(g)} + \beta \cdot f_g^{best} \quad (3)$$

Where f_g^{best} represents the fitness of the best individual and ϕ_g indicates a new type of characteristic measure which represents the weight sum of high performing mean and best performing individuals at generation g . α and β in (3) represents the weights for the average fitness of the high performance individuals and the fitness of the best individual respectively. In this paper, we set α to 0.6 and β is set to 0.4.

$$Pivot = \frac{1}{T} \cdot \sum_{g=j}^{j+T-1} \Delta_{g-1} \cdot (1 - e^{-\frac{|\Delta_{g-1}|}{\theta \cdot \Delta_g}}) \quad (4)$$

Where T represents the circle time of the population variation and it is set to 1 herein. The parameter θ indicates the constant used to adjust the magnitude of the variation. It has been proved that an appropriate population variation scheme can benefit to the evolution for low computational effort. In our approach, the population-generation profile $S(g)$ for exponent function (EXP) at any generation g is defined by (5).

$$\Delta S(g) = (-1)^n \cdot S(g) \frac{|\Delta_g|}{(f_{g-1}^{best} + f_g^{best})} \quad (5)$$

Where $n=1$, if $\Delta g > Pivot$; $n=2$, if $\Delta g \leqslant Pivot$. We must give a constraint to $\Delta S(g)$ for a normal run of GP, which is denoted as the expression: $\Delta S(g) = 0$, if $S(g) - \Delta S(g) \leq 1$.

The scheme for population increment or reduce can be described as follow: Sort the individuals in population at g generation according to their fitness at descending order. $f_1 > f_2 > \dots > f_N$. If the individuals need to be removed according to pivot, then $\Delta S(g)$ individuals in the rear of the population will be eliminated. Else $\Delta S(g)$ high performing individuals selected do the mutation and will be inserted into the population at $g+1$ generation.

2.3 Kalman Operator

Assumed that there is a linear discrete time system as follow:

$$\begin{cases} X(k+1) = A(k) \cdot X(k) + B(k) \cdot \varepsilon(k) \\ Y(k) = H(k) \cdot X(k) + \eta(k) \end{cases} \quad (6)$$

Where X is an n dimension state vector, $A(k)$ is the $n \times n$ state transition matrix, $\varepsilon(k)$ is a p dimension disturbance vector, $B(k)$ is $n \times p$ input disturbance transition matrix, $Y(k)$ is m dimension observation data vector, $H(k)$ is $m \times n$ measurement transition matrix, $\eta(k)$ is m dimension observation error vector. It is assumed that input noise $\varepsilon(k)$ and measurement noise $\eta(k)$ are uncorrelated and zero-mean independent white noises, of which the characteristic are as follow:

$$\begin{aligned}
E[\varepsilon(k)] &= O; E[\eta(k)] = O; \\
E[\varepsilon(k)\mu(j)^T] &= O; \\
E[\varepsilon(k)\varepsilon(j)^T] &= Q\delta(k-j); \\
E[\eta(k)\eta(j)^T] &= R\delta(k-j);
\end{aligned} \tag{7}$$

The initial state $X(0)$ is n dimension random vector and its statistical characteristics are shown as $E[X(0)] = x_0$ and $E[(X(0) - x_0)(X(0) - x_0)^T] = P(0)$.

The initial state $X(0)$ has no relationship with $\varepsilon(k)$ and $\eta(k)$. So as for any k there are the cases $E[X(0)\varepsilon(k)^T] = 0$ and $E[X(0)\eta(k)^T] = 0$.

Let $\hat{X}(k+1|k)$ is the pre-estimation value for $X(k+1)$ at time $k+1$. It is known that $\hat{X}(k)$ is the optimal value for $X(k)$. When the measurement value at time $k+1$ has not been obtained, the pre-estimation value can be calculated by (8):

$$\hat{X}(k+1|k) = A(k+1)\hat{X}(k) \tag{8}$$

The estimation value of $Y(k+1)$ can be calculated by (9):

$$\hat{Y}(k+1) = H(k+1) \cdot \hat{X}(k+1|k) \tag{9}$$

These two formulas remove the effect of the uncertain guise white noise. $\hat{X}(k+1)$ is the pre-estimation value for $X(k+1)$ and covariance matrix of its error is expressed as follow:

$$P(k+1) = E[(X(k+1) - \hat{X}(k+1))(X(k+1) - \hat{X}(k+1))^T] \tag{10}$$

There must be error in pre-estimation value. The formula: $Y(k+1) - H(k) \cdot \hat{X}(k+1)$ is used to amend the estimation value of $\hat{X}(k+1)$ for reducing the error between estimation and measurement data. Therefore, the amend formula can be expressed as (11):

$$\hat{X}(k+1) = \hat{X}(k+1|k) + K(k+1)[Y(k+1) - H(k) \cdot \hat{X}(k+1|k)] \tag{11}$$

Where K is the Kalman gain. The Kalman gain must make the mean square deviation of the error minimized., it means that the trace of the $n \times n$ $P(k+1)$ is minimum. ($Tr(P) = \min$). According to the minimum the mean square deviation of the error, the Kalman gain can be calculated by (12):

$$K(k+1) = P(k+1|k)H^T(k+1)[H(k+1)P(k+1)H^T(k+1) + R(k+1)]^{-1} \tag{12}$$

Where $P(k+1|k)$ is the pre-estimation value of the square deviation matrix.

The power system load evolutionary model can be mapping into a polynomial model which can be described as follow:

$$y(k) = a_1 u_1(k) + a_2 u_2(k) + \cdots + a_n u_n(k)$$

Where $u_i(k)$ and $y(k)$ are respectively the input sequence and outputs. The measurement matrix $H(k) = [u_1(k), u_2(k), \dots, u_n(k)]$ of Kalman filtering can composed by $u_i(k)$ and the parameters can be treated as states.

$$\begin{cases} x_1(k) = a_1 \\ x_2(k) = a_2 \\ \dots \\ x_n(k) = a_n \end{cases} \Rightarrow \begin{cases} x_1(k+1) = a_1 + w_1 \\ x_2(k+1) = a_2 + w_2 \\ \dots \\ x_n(k+1) = a_n + w_n \end{cases} \quad (13)$$

Where $\{w_i\}$ are the gaussian white noises, which are independent with the state vector: $x^T(k) = \{a_1, a_2, \dots, a_n\}$. Let $A(k) = 1$ and $B(k) = 1$, the state equations for the model can be defined as (14):

$$\begin{aligned} X(k+1) &= X(k) + \varepsilon(k) \\ Y(k) &= H(k) \cdot X(k) \end{aligned} \quad (14)$$

The set of recurrence formula in Kalman filtering is as follow:

$$\begin{aligned} \hat{X}(k+1|k) &= A(k+1)\hat{X}(k) \\ \hat{X}(k+1) &= \hat{X}(k+1|k) + K(k+1)[Y(k+1) - H(k) \cdot \hat{X}(k+1|k)] \\ P(k+1|k) &= A(k+1|k)P(k)A^T(k+1|k) + Q(k) \\ K(k+1) &= P(k+1|k)H^T(k+1)[H(k+1)P(k+1)H^T(k+1) + R(k+1)]^{-1} \\ P(k+1) &= [I - K(k+1)H(k+1)]P(k+1|k) \end{aligned} \quad (15)$$

The $\hat{X}(k+1)$ and $P(k+1)$ calculated by these formula above are the results we need.

3 Experiments

The experiments in this paper are used for the comparison among the convention approaches, RBF network and the proposed approach for the power system load modeling.

The experiment as a proof of new methodology uses the power system load data measured from the 35kv and 10kv main transformers of TanShi substation in China. These data was separated morning and evening monitor data. The standard measured data are used to for load modeling testing.

The main evaluation index for the methodology is mean error (*ME*) that indicates the distance between measured data from actual transformers and calculated data by obtained model, which can be defined as follow:

$$ME = \frac{1}{N} \sum_{i=1}^N |y_i - y'_i| \quad (16)$$

Where *N* is the number of sample data. y_i represents the *i*-th measured data from actual transformers and y'_i represents the *i*-th calculated data by obtained model. The small *ME* of the load model represents the high accuracy of the model and the power system load model is sufficiently closer to the actual data.

Table 1. Main transformer 35Kv active power model (*P*)

35Kv <i>P</i>	Monitor in morning	ME	Monitor in evening	ME
power function	$P = 0.998U^{0.690}$	0.02515	$P = 0.992U^{1.038}$	0.00859
polynomial	$P = 0.674U + 0.325$	0.0237	$P = 1.0319U - 0.040$	0.00859
Evolutionary model	$P = (U^2 + 1)/(U + 1)$	0.01406	$P = \frac{2U^4 - 2U^2}{U^2 - U + 5.5084} + 1$	0.008116

Table 2. Main transformer 35Kv inactive power model (*Q*)

35Kv <i>Q</i>	Monitor in morning	ME	Monitor in evening	ME
power function	$Q = 1.015U^{4.122}$	0.0778	$Q = 1.014U^{6.294}$	0.02676
polynomial	$Q = 4.392U - 3.351$	0.09935	$Q = 8.621U - 7.640$	0.01979
Evolutionary model	$Q = U^2(U^3 - 0.9176U + 1)$	0.06136	$Q = 4U^2 - 3$	0.01705

Table 3. Main transformer 10Kv active power model (*P*)

10Kv <i>P</i>	Monitor in morning	ME	Monitor in evening	ME
power function	$P = 0.983 \cdot U^{0.27}$	0.00969	$P = 0.993 \cdot U^{1.183}$	0.00893
polynomial	$P = 0.264U + 0.719$	0.01002	$P = 1.182U - 0.189$	0.00891
Evolutionary model	$P = 1 + (\frac{U}{3} + \frac{1}{3} - \frac{1}{U}) / [3U(9 + 3U - \frac{3}{U} + \frac{6U^2 - 3}{3U^2 - 2U^3 + U})) - 3]$	0.007906	$P = (3U^6 - 3U^7 + U) / (U^2 - U) + 1$	0.008139

Table 4. Main transformer 10Kv inactive power model (Q)

10Kv Q	Monitor in morning	ME	Monitor in evening	ME
power function	$Q = 1.004U^{1.991}$	0.014663	$Q = 0.997U^5$	0.0392
polynomial	$Q = 1.9995U - 0.99$	0.01429	$Q = 6.325U - 5.325$	0.0258
Evolutionary model	$Q = U^2 - 9(U - 1)^3$	0.01364	$Q = 0.0636U^2 + 2U$	0.0191

The results of the experiments for 35kv active and reactive power model are given in Table1 and Table2 and those for 10Kv in Table3 and Table4. Tables (1-4) illustrate that the accuracy of power system load models evolved by DPVGP was higher 4.6~48% than polynomial model and power function model. In these evolutionary experiments, 30 independent runs have used for obtaining the best result. Eight initial models obtained by DPVGP all have improvement in accuracy since its structure is more complicated than conventional methods. The best result calculated by DPVGP is the reactive power load model of 10Kv monitor in evening, which has raised the accuracy 48% approximately. While the worst model is the active power load model of 35Kv monitor in evening that only raise the accuracy 4.6%.

Table 5. Comparison between RBF and proposed method for inactive power model (Q)

Approaches	10Kv Q	10Kv Q	35Kv Q	35Kv Q
	Monitor in morning ME	Monitor in evening ME	Monitor in morning ME	Monitor in evening ME
RBF network	0.1227	0.2745	0.2180	0.5593
Evolutionary model	0.01364	0.06136	0.01364	0.01705

Comparison between RBF network and proposed method for inactive power model has been done in this experiment and the results in Table.5 show that the performance of the RBF network is poorer than proposed method for inactive power modeling. The performance of the RBF network for inactive power model strongly relies on the training data. Only when the feature of the training data is close to the actual measure data, the model of RBF network based on these training data can present a high performance. However, the proposed method can perform well without any prior knowledge.

The applications of the Kalman filter have proved that the stability of Kalman filter in theory can't guarantee the convergence of the filter in practice. The efficiency of the filter in practice can't be guaranteed because there are model error and computation error in system [17]. The divergence of algorithm also happens in our experiments. Therefore, the parameters in Kalman such as the initial value of error covariance matrix P , gaussian noise parameter R and Q are adjusted for achieving satisfying results. The traces of error covariance matrix for all the active or reactive power load models

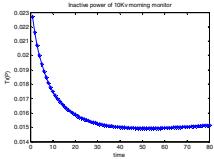
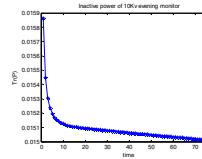
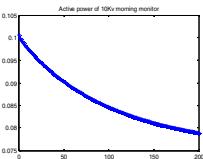
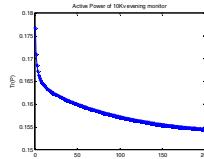


Fig. 1. The trace of error covariance matrix for active power model

Fig. 2. The trace of error covariance matrix for inactive power model

Table 6. The optimization model after processing of KO

	Active power model	ME	Inactive power model	ME
35kV P morning	$P = \frac{0.96376U^2 + 1.0522}{U + 1}$	0.01205	35kV Q morning	$Q = 0.9116U^5 - 0.9184U^3 + 1.0874U^2$
10kV P morning	$P = 3.1107 - (350.08U + 443.17 + \frac{133.45}{U}) / [3U(9 + 3(U - \frac{3}{U} + \frac{6U^2 - 3}{3U^2 - 2U^3 + U})) - 3]$	0.007298	10kV Q morning	$Q = -40.068U^3 + 120.68U^2 - 118.8U + 39.194$
35kV P evening	$P = \frac{2.0046U^4 - 1.9894U^2}{U^2 - U + 5.5084} + 0.9965$	0.008136	35kV Q evening	$Q = 4.1802U^2 - 3.2182$
10kV P evening	$P = -709.31U^8 + 307.3U^9 + 349.71U^7 + 870.81U^3 - 1076.3U^2 + 258.8$	0.003509	10kV Q evening	$Q = 11.429U^2 - 17.956U + 7.527$

calculated by KO are shown in Fig1 and Fig2. In this case, the errors between actual and calculation value of optimization load model were sufficiently close to zero.

After the process of KO, the optimization load models and their *ME* are given by Table6. It shows that the accuracy of models processed by KO are improved 7.69~81.3% except the load model of 35Kv evening monitor. The best result calculated by KO is the reactive power load model of 10Kv monitor in evening, which has raised the accuracy 81.3% approximately. While the worst model is the active power load model of 10Kv monitor in morning that only raises the accuracy 7.69%. The results give the proof that the KO added into methodology has played an important role in system optimization modeling. Due to effect of KO, the probability of resulting in better model in terms of *ME* is greatly increased.

5 Conclusion

This work dealt with the problem of load modeling in electric power systems using DPVGP-KO. The results demonstrated the efficiency of DPVGP-KO. The load model

evolved by DPVGP has higher accuracy than conventional model such as power function model and polynomial model. KO is introduced and used to achieve more accurate load model by optimizing parameters of initial model obtained by DPVGP. The experiments show that the proposed approach for load modeling are valid, since the obtained active and reactive power optimization model residuals are very close to zero.

Acknowledgments. Thanks to the support of Shanghai key scientific and technological projects (0951150603) and the help of the Professor Minglu Li and Jian Cao.

References

1. Ju, P., Ma, D.Q.: Load modeling in power system. Water-conservancy hydropower Press, Beijing (1995)
2. Awang, B.J.: The Instability Effect of Constant Power Loads. In: 2004 National Power & Energy Conference (PECon), pp. 175–179. IEEE Press, New York (2004)
3. Zarco, P., Exposito, A.G.: Power system parameter estimation: a survey. *IEEE Trans Power Syst.* 15, 16–21 (2000)
4. Coker, M.L., Kgasoane, H.: Load modeling, *IEEE Africon* (1999)
5. Mastorocostas, P.A., Theocharis, J.B., Kiartzis, S.J.: A hybrid fuzzy modeling method for short-time load forecasting. *Mathematic and Computers in simulation* 51(2), 221–232 (2000)
6. Wu, Q.H., Wen, J.Y., Nuttall, K.I.: Power System Load Modeling by Evolutionary Computation Based on System Measurements. *Electric Machines & Power Systems* 31(5), 423–439 (2003)
7. IEEE Task Force on load representation for dynamic performance, Load representation for dynamic performance analysis. *IEEE Trans Power Syst* 8(2), 472–482 (1993)
8. Kóczy, L.T., Tikk, D., Gedeon, T.D.: On Functional Equivalence of Certain Fuzzy Controllers and RBF Type Approximation Schemes. *International Journal of Fuzzy Systems* 2(3), 164–175 (2000)
9. Baranyi, P., Kóczy, L.T., Gedeon, T.D.: A Generalized Concept for Fuzzy Rule Interpolation. *IEEE Transactions on Fuzzy Systems* 12(6), 820–837 (2004)
10. Yang, Z.G., Che, Y.B., Cheng, K.W.E.: Genetic Algorithm-Based RBF Neural Network for Load Forecasting Model. In: 2007 IEEE Power Engineering Society General Meeting, pp. 1–6. IEEE Press, New York (2007)
11. Wang, Z.Q., Chen, X.Q., Deng, C.H.: The Power System Load Modeling Based on Recurrent RBF Neural Network. In: The 8th International Power Engineering Conference (IPEC 2007), pp. 910–915. IEEE Press, New York (2007)
12. Lia, T.M.M., Alexandre, A.M.: Load modeling at electric power distribution substations using dynamic load parameters estimation. *Electrical Power and Energy Systems* 26, 805–811 (2004)
13. Gastaldi, M., Lamedica, R., Nardecchia, A.: Short-term forecasting of municipal load through a Kalman filtering based approach. In: 2004 IEEE PES Power Systems Conference and Exposition, pp. 1453–1458. IEEE Press, New York (2004)
14. Al-Hamadi, H.M., Soliman, S.A.: Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model. *Electric Power Systems Research* 68(1), 47–59 (2004)

15. Tomassini, M., Vanneschi, L., Cuendet, J.: A new technique for dynamic size populations in genetic programming. In: 2004 IEEE Congress on Evolutionary Computation (CEC 2004), pp. 486–493. IEEE Press, New York (2004)
16. Kouchakpour, P., Zakhnich, A., Branul, T.: Dynamic population variation in genetic programming. *Information Sciences* 179, 1078–1091 (2009)
17. Quigley, A.L.C.: An approach to the control of divergence in Kalman filter algorithm. *Int.J.Cont.* 17, 741–746 (1993)

A Robust Iris Segmentation with Fuzzy Supports

C.C. Teo¹, H.F. Neo¹, G.K.O. Michael¹, C. Tee¹, and K.S. Sim²

¹ FIST

² FET, Multimedia University, Jalan Ayer Keroh Lama, 75450 Malacca, Malaysia
`{ccteo,hfneo,michael.goh,tee.connie,kssim}@mmu.edu.my`

Abstract. Today, iris recognition is reported as one of the most reliable biometric approaches. With the strength of contactless, the hygienic issue is therefore minimized and the possibility of disease infection through the device as a medium is low. In this paper, a MMU2 iris database with such consideration is created for this study. Moreover, the proposed iris segmentation method has shown its robustness with intelligent fuzzy supports. Furthermore, it has been tested with 18414 iris images across different databases available in the public without changing any threshold values and parameters. The experiment results show a total of 17915 or 97.30% of correct iris segmentation.

Keywords: Iris Segmentation, MMU2, Fuzzy, Pupil, Biometrics.

1 Introduction

Human iris is stable throughout one's life [1]. It is the chaotic pattern of furrows, freckles, coronas, stripes and crypts that interweaving on the plane between the pupil and sclera. Be exceptionally rich in natural uniqueness, iris recognition gains a competitive advantage over other biometrics classes. This uniqueness is one of the virtues of iris recognition because no two irises are alike and the chance of finding two people with identical iris pattern is almost zero [2, 3]. Nevertheless, robust iris recognition is still a challenging because the overall performance could be encumbered with poor image quality and the inaccuracy of iris segmentation. Normally, the poor quality of image could be filtered and compensated through good hardware design and convention image preprocessing techniques. However, the success of iris segmentation still requires recondite principles and knowledge to differentiate between those useful region of interest and noises. The noises include partially occluded eyelids, eyelashes and shadow and refraction spots [4].

During the late 20th century, a scrupulous study of iris segmentation for supporting the further recognition works has been initiated by Daugman [5]. His fundamental approach was to apply active contour and integro-differential operators to detect the associate region of interest. On the other hand, Tisse et al. [2] and Wildes [6] used Hough transform, integro-differential operators and gradient-based edge detector to detect local boundaries of iris respectively. Although these methods are said to be effective, but they are time consuming and could be easily affected by the local variation [4]. After that, the research of iris segmentation and its following works have been progressed by a group of researchers from [15] with disinterested sharing of CASIA iris databases version 1.0. With these iris images, Cui et al. [1] made use of the low frequency information from wavelet

transform for pupil segmentation and localized the iris with integro-differential operator. Furthermore, Ma et al. [8] approximated the center of pupil and iris region before executing edge detection and Hough transform. Alternatively, some of the researchers have paid their attention to process non-cooperative iris images. Proenca et al. [9] claimed that many of the iris recognition approaches obtain almost optimal results because of the favorable conditions of iris images with few image noise factors. In addition, UBIRIS [10] and CASIA iris database V-3.0 [7] with opposition condition have been collected. For non ideal iris images, Proenca et al. [10] improved the robustness of local edge segmentation using fuzzy k-mean clustering algorithm. Soon, He et al. [4] proposed a refection removal method on CASIA version 3.0 iris images follow with Adaboost-cascade and pulling and pushing methods for iris detection. To improve He et al. [4] works, Carlos et al. [11] used the same pulling and pushing methods but include an additional active contour. In the latest report, Jonathon [12] found that it is possible to detect and identify pupil using modify Haar wavelet. For all of the related works, no one has tested the effectiveness of their algorithm in large and different dataset that available in the public.

In this paper, a new iris segmentation method is presented for large number of iris images across different databases. They are CASIA iris database (V-1.0) [15], CASIA iris database (V-3.0) [7], MMU1iris database [13] and proposed MMU2 iris database [14]. The rest of this paper is structured as follows: Section 2 is devoted to propose iris database in our experiments. Section 3 explains the wavelet transformation which makes all iris images are fair to be compared. Section 4 contains the overall process of proposed iris segmentation method. Experimental results are presented in Section 5. Finally, the conclusion is reported in the last section.

2 Iris Database

MMU2 iris database [14] consists of 995 iris images. The iris images are collected using Panasonic BM-ET100US Authenticam using visible light and its operating range is even farther with a distance of 47-53 cm away from the user. The captured iris images have reflected spot at the pupil and noises such as hairs, eyelashes, eyelids and eyebrows. These iris images are contributed by 100 volunteers with different age and nationality. They come from Asia, Middle East, Africa and Europe. Each of them contributes 5 iris images for each eye. There are 5 left eye iris images which are excluded from the database due to cataract disease. In addition, more images with eyeglasses and contact lenses are included. Table 1 shows the profile of the MMU2 iris database.

Table 1. Profile of the MMU2 iris database

Gender	Male	66%
	Female	34%
Age	<25	66%
	25-40	33%
	>40	1%
Environment	Indoor	84%
	Outdoor	16%

For Malaysians, people from different races are included (Malay, Chinese, Indian)

* Other countries include: Algeria, Bangladesh, China, Yemen, Sudan, Iran, South Africa, India, Somalia, Japan, Thailand, Myanmar, France, Germany, Libya, Jordan and Nigeria

3 Wavelet Transform

Wavelet transform is the best scale analysis tool especially to the signal with finite energy. A particular image is transformed to multiresolution representation, which enables one to efficiently compute a small-sized feature representation

The wavelet decomposition of a signal $f(x)$ can be obtained by convolution of signal with a family of real orthonormal basis, $\psi_{a,b}(x)$ [16]:

$$(W_\psi f(x))(a,b) = a^{-\frac{1}{2}} \int_{\mathfrak{R}} f(x) \psi\left(\frac{x-b}{a}\right) dx, f(x) \in L^2(\mathfrak{R}) \quad (1)$$

where $a, b \in \mathfrak{R}$ and $a \neq 0$ are the dilation parameter and the translation parameter respectively.

Each iris images in CASIA-IrisV3-Lamp [7] have the resolution of 640 x 480. It is impractical to compute and expensive to store. Therefore, one-level decomposition has been applied to all the images inside CASIA-IrisV3-Lamp so that their resolution will reduce to 320 and 240.

4 Iris Segmentation

4.1 Detection of Inner Iris Boundary

Today, Fuzzy logics are applied in many applications such as remote sensing image segmentation [17], color image enhancement [18] and image segmentation [19]. With Fuzzy logics, the machines could mimic human to make decision from the approximate information [18]. The mathematics of fuzzy set theory [20] describes an alternative value in the interval between '0' and '1'. They are represented with commonsense linguistic labels such that $\Omega = \{\text{small, large, slow, fast, strong, weak and others}\}$. Fuzzy sets represent A in universe may be represented as a set of ordered pairs such that

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (2)$$

where x is a linguistic variable in universe if $\mu_A(x) > 0$. This variable is described by $T(X)$ with membership function $\mu(x)$.

$$T(x) = \left\{ T_x^1, T_x^2, T_x^3, \dots, T_x^m \right\} \quad (3)$$

$$\mu(x) = \left\{ \mu_x^1, \mu_x^2, \mu_x^3, \dots, \mu_x^m \right\} \quad (4)$$

In natural, human pupil is a dark circular region. To the best of image's condition, this information could help machine to detect the pupil region easily. However, sometimes the pupil is partially mixed and influenced by the noises such as refection spots, eyelashes, eyelids and hairs. It creates the uncertainties area in a particular iris image. Therefore, a set of fuzzy rules to handle such uncertainties is proposed in this study. The steps are further explained as following:

Step 1: Scan the global image to find out the darkest pixel (τ). This pixel has the highest chance to be one of the members of pupil. However, it still could be one of the members from eyelashes, eyebrows or even hairs.

$$\tau = \arg \min(\sum_x \sum_y f(x, y)) \quad (5)$$

Step 2: Increase of the brightness level of τ by adding a cut off interval with predefined value (δ). This is to increase the portability of τ across different iris databases.

$$\tau = \tau + \delta \quad (6)$$

Step 3: $\forall \sum_x \sum_y f(x, y)$, identify all other dark pixels when their values are

fall below τ . At the same time, increase the counter (α) whenever a dark pixel is detected and set this pixels to 0 (extreme dark). After this step, dark image-region will be clustered and this region can be considered as a pupil region. Exceptionally, it could be the region of eyelashes, eyebrows and hairs.

$$f(x, y) = \begin{cases} f(x, y) & \text{if } f(x, y) > \tau \\ 0 & \text{if } f(x, y) \leq \tau \end{cases} \quad (7)$$

Step 4: Perform a technical rejection of iris image whenever the estimated location of pupil falls outside the predefined area. It may mean that there are high chances that the iris portion is excluded totally or partially from the image during iris image acquisition.

Step 5: Consider a single-input and single-output system on a particular iris image set A. Let $x = (x_1, x_2, x_3, \dots, x_w)^T$ represent the input vectors and let $z = (z_1, z_2, z_3, \dots, z_w)^T$ be the output vectors. Then, a linguistic variable of x is declared to indicate the darkness's strength level of every column of the image. In addition, $T(x)$ refers to the corresponding term sets {low, medium or high}. On the other hand, variable z is set to represent the intensity and therefore $T(z)$ includes a term set {dark, medium, bright}

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (8)$$

$$T(x) = \left\{ T_x^1, T_x^2, T_x^3, \dots, T_x^w \right\} \quad (9)$$

Step 6: Calculate the total number of dark pixels with the intensity of '0' for each columns (λ_{col}). Next, determine the membership function $\mu(x_c)$ by divide with the total number of dark pixels (α) in the global image.

$$\lambda_{col} = \begin{cases} \lambda_{col} + 1 & \text{if } f(x, y) = 0 \\ \lambda_{col} & \text{otherwise} \end{cases} \quad (10)$$

$$\mu(x_c) = \left\{ \frac{\lambda_{col}}{\alpha}, c = 1, 2, 3, \dots, w \right\} \quad (11)$$

where w is image's width and (λ_{col}) is a local counter to measure darkness's strength level for a particular column.

Step 7: Repeat step 5 and step 6 with a minor modification according to Equation 12 to 14 and measure darkness's strength level for every rows.

$$T(y) = \left\{ T_y^1, T_y^2, T_y^3, \dots, T_y^h \right\} \quad (12)$$

$$\lambda_{row} = \begin{cases} \lambda_{row} + 1 & \text{if } f(x,y) = 0 \\ \lambda_{row} & \text{otherwise} \end{cases} \quad (13)$$

$$\mu(y_r) = \left\{ \frac{\lambda_{row}}{\alpha}, r = 1, 2, 3, \dots, h \right\} \quad (14)$$

where h is image's height and (λ_{row}) is a local counter to measure darkness's strength level for a particular row. Next, determine the membership function $\mu(y_r)$ by divide with the total number of dark pixels (α) in the global image.

Step 8: Categorize the degree of membership $\mu(x_c)$ and $\mu(y_r)$ according to different darkness's strength level. Next, construct a single *if-then* rule as following:

- R1: IF darkness's strength level is high, THEN intensity level is dark*.
- R2: IF darkness's strength level is medium, THEN intensity level is medium*.
- R3: IF darkness's strength level is low, THEN intensity level is bright*.

Step 9: With the reference of fuzzy output vectors, an initial search is preferably executed from the bottom to the top. If the search encounters dark* row continuously, the first encountered bright* row will be treated as bottom segmentation line. As the search continues and bypass with the medium* row, the first encountered bright* row again is treated as top segmentation line. Repeat the same step for columns until it reaches convergence. Figure 1 shows the example of automatic segmentation for pupil region.

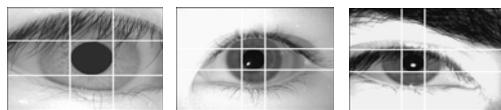


Fig. 1. Automatic segmentation for pupil region with complex noises

4.2 Detection of Outer Iris Boundary

Since the information of the pupil region is known, the radius and center (c) of the pupil (r) can be estimated easily. Refer to a mixed joint density in Equation 15, our

outer iris detection starts from the boundary of pupil and performs searching until the convergence of principal circular contrast.

$$\omega = \arg \max \left(\int_{R=c+r}^{\infty} \sum_{\theta=0}^{2\pi} ((f_{R+1}(\theta, R+1) - f_R(\theta, R)) dR \right) \quad (15)$$

where ω is the estimated outer boundary of iris, c is the estimated center of pupil and r is the radius of pupil.

5 Experiment Results

For experiment setup, a prototype system was developed in Microsoft Visual Basic .Net, and installed on a 1.80GHz Intel machine with 512MB of RAM. The proposed iris preprocessing method is tested on different iris databases [7, 13, 14, 15] with a total number of 18414 iris images and our method proves its robustness to detect inner and outer boundaries of the iris. Table 2 shows the localization results for different iris databases with the comparison with the inner localization using Hough Transform method. The proposed method works well for CASIA V-1.0 iris images, where generally the size of pupil is relatively larger with no reflected spot. The results of the implementation of the combination of the proposed pupil detection method and the mixed joint density function are stable and accurate across different iris databases. Figure 2 and Figure 3 presents some correct and incorrect results of iris localization for CASIA (V-1.0), MMU1 and MMU2 iris images respectively.

Table 2. Iris localization results for different iris databases

	CASIA V-1.0	MMU1	MMU2
Inner Localization (Proposed Method)	100.00 %	99.56	98.40%
Inner Localization (Hough Transform)	99.47%	96.4%	97.39%
Outer Localization (Proposed Method)	100.00%	99.33%	99.30%

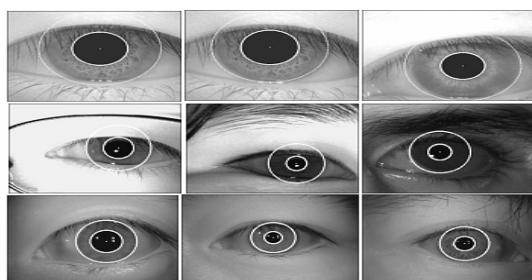


Fig. 2. Correct results of iris localization

In addition, the proposed algorithms also test with the latest version of CASIA-IrisV3-Lamp (V-3.0) iris database. In the experiments, 251 poor quality iris images are filtered by automatic technical rejection scheme. For the rest of the images, the proposed algorithm shows 98.56% or 15732 perfect iris segmentation. Figure 4 presents the incorrect results of iris localization for CASIA-IrisV3-Lamp (V-3.0).

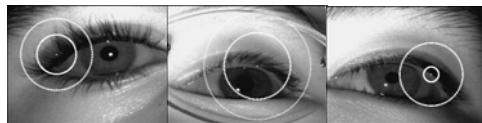


Fig. 3. Incorrect results of iris localization in MMU2 iris database

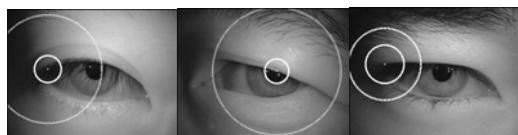


Fig. 4. Incorrect results of iris localization for CASIA-IrisV3-Lamp (V-3.0)

6 Conclusion

In conclusion, the proposed iris segmentation method shows an encouraging result with 18414 iris images across different iris databases [7, 13, 14, 15] are used in the experiments. Furthermore, none of the parameters from the algorithm have been changed when different databases are tested. Thus, potentially, a many-to-one platform for the iris recognition server can be setup to receive iris images from different type of cameras. On the other hand, a real-time iris images were collected for MMU2 iris database using portable web camera. Since the iris images were taken in a long distance under different kinds of real-time condition, the preprocessing of these iris images is more challenging where it is convenient to be used by the user but it creates more noises in the image at the same time. Lastly, the proposed iris segmentation method with Fuzzy logics provides a good foundation towards developing an accurate iris recognition system.

Acknowledgments. The authors wish to thank Institute of Automation, Chinese Academy of Sciences for providing CASIA iris databases for web access [7, 15].

References

1. Cui, J., Wang, Y., Tan, T., Ma, L., Sun, Z.: A Fast and Robust Iris Localization Method Based on Texture Segmentation. In: SPIE Defense and Security Symposium (2004)
2. Tisse, C., Martin, L., Torres, L., Robert, M.: Person Identification Technique using Human Iris Recognition. In: Proc. Vision Interface, pp. 294–299 (2002)
3. Boles, W., Boashash, B.: A Human Identification Technique Using Images of the Iris and Wavelet Transform. IEEE Trans. Signal Processing 46(4), 1185–1188 (1998)

4. He, Z., Tan, T., Sun, Z., Qiu, X.: Toward Accurate and Fast Iris Segmentation for Iris Biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 31(9), 1670–1684 (2009)
5. Daugman, J.: How Iris Recognition Works. *IEEE Trans. Circuits and Systems for Video Technology* 14(1), 21–30 (2004)
6. Wildes, R.: Iris Recognition: An Emerging Biometric Technology. *Proc. of IEEE* 85(9), 1348–1363 (1997)
7. CASIA Iris Image Database V-3.0: (2010),
<http://www.cbsr.ia.ac.cn/IrisDatabase.htm>
8. Ma, L., Tan, T., Wang, Y., Zhang, D.: Efficient Iris Recognition by characterizing Key Local Variations. *IEEE Trans. IP* 13(6), 739–750 (2004)
9. Proenca, H., Alexandre, L.A.: UBIRIS: A noisy iris image database. *Proc. ICAIP* 1, 970–977 (2005)
10. Proenca, H., Alexandre, L.A.: Iris Segmentation Methodology for Non-Cooperative Recognition. *IEE Proc. VISIP* 153, 199–205 (2006)
11. Carlos, A.C.M.B., Tsang, I.R., George, D.C.C.: A Combined Pulling & Pushing and Active Contour Method for Pupil Segmentation. In: *Proc. ICASSP*, pp. 850–853 (2010)
12. Jonathan, M.S.: A New Method of Pupil Identification. *IEEE Potential*, 15–20 (2010)
13. MMU1 Iris Database (2010),
<http://pesona.mmu.edu.my/~ccteo/MMUirisdatabase.rar>
14. MMU2 Iris Database (2010),
<http://pesona.mmu.edu.my/~ccteo/MMU2irisdatabase.rar>
15. CASIA Iris Image Database, V-1.0 (2010), <http://www.sinobiometrics.com>
16. Mallat, S.G.: A Theory For Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Trans. Pattern Analy. Mach. Intell.* 11(7), 674–693 (1989)
17. Lizarazo, I., Elsner, P.: Fuzzy regions for handling uncertainty in remote sensing image segmentation. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) *ICCSA 2008, Part I. LNCS*, vol. 5072, pp. 724–739. Springer, Heidelberg (2008)
18. Milindkumar, V.S., Ladha, S.A., Prashant, R.D.: Fuzzy system for color image enhancement. *World Academy of Science, Engineering and Technology*, 311–316 (2008)
19. Fan, J., Yau David, K.Y., Elmagamid, A.K., Aref, W.G.: Automatic image segmentation by integrating color edge extraction and seeded region growing. *IEEE Trans. Image Processing* 10, 1454–1466 (2001)
20. Zadeh, L.: Fuzzy Sets. *Information and control* (8), 338–353 (1965)

An Adaptive Local Search Based Genetic Algorithm for Solving Multi-objective Facility Layout Problem

Kazi Shah Nawaz Ripon, Kyrre Glette, Mats Høvin, and Jim Torresen

Department of Informatics, University of Oslo, Norway
`{ksripon, kyrrehg, matsh, jimtoer}@ifi.uio.no`

Abstract. Due to the combinatorial nature of the facility layout problem (FLP), several heuristic and meta-heuristic approaches have been developed to obtain good rather than optimal solutions. Unfortunately, most of these approaches are predominantly on a single objective. However, the real-world FLPs are multi-objective by nature and only very recently have meta-heuristics been designed and used in multi-objective FLP. These most often use the weighted sum method to combine the different objectives and thus, inherit the well-known problems of this method. This paper presents an adaptive local search based genetic algorithm (GA) for solving the multi-objective FLP that presents the layouts as a set of Pareto-optimal solutions optimizing both quantitative and qualitative objectives simultaneously. Unlike the conventional local search, the proposed adaptive local search scheme automatically determines whether local search is used in a GA loop or not. The results obtained show that the proposed algorithm outperforms the other competing algorithms and can find near-optimal and non-dominated solutions by optimizing multiple criteria simultaneously.

Keywords: Adaptive local search, Multi-objective facility layout problem, Pareto-optimal solution, Multi-objective evolutionary optimization.

1 Introduction

The objective of the FLP is the optimum arrangement of facilities/departments on a factory floor, such that the costs associated with projected interactions between these facilities can be optimized. These costs may reflect minimizing the total cost of transporting materials (material handling cost) or maximizing the adjacency requirement between the facilities. In essence, FLP can be considered as a searching or optimization problem, where the goal is to find the best possible layout. In fact, the problem falls into the category of multi-objective optimization problems (MOOPs). On one hand, many researchers handle the problem as one of optimizing product flow by minimizing the total material handling costs. On the other hand, it can also be considered as a design problem by considering qualitative information on how different activities, like safety, management convenience, and assistant power supply are related from the viewpoint of adjacency.

Facility layout planning plays an important role in the manufacturing process and has a serious impact on a company's profitability. A good layout will help any company to improve its business performance and can reduce up to 50% of the total operating expenses [1]. Although the FLP is an inherently MOOP, it has traditionally

been solved considering only one objective, either qualitative or quantitative feature of the layout. Quantitative approaches involve primarily the minimization of material handling (MH) costs between various departments. Qualitative goals aiming at placing departments that utilize common materials, personnel, or utilities adjacent to one another, while separating departments for the reasons of safety, noise, or cleanliness. In general, minimization of the total MH costs is often used as the optimization criterion in FLP. However, the closeness rating, hazardous movement or safety, and the like are also important criteria in FLP. Based on the principle of MOOP, obtaining an optimal solution that satisfies all the objectives is almost impossible. It is mainly for the conflicting nature of objectives, where improving one objective may only be achieved when worsening another objective. Surprisingly, however, there is a little attention to the study of multi-objective FLP.

Although dealing with multiple objectives has received attention over the last few years [2], [3], these approaches are still considered limited, and mostly dominated by the unrealistic weighted sum method. In this method, multiple objectives are added up into a single scalar objective using weighted coefficients. However, there are several disadvantages of this technique [4]. Firstly, as the relative weights of the objectives are not exactly known in advance and cannot be pre-determined by the users, the objective function that has the largest variance value may dominate the multi-objective evaluation. As a result, inferior non-dominated solutions with poor diversity will be produced. Secondly, the user always has to specify the weight values for objectives and sometimes these will not have any relationship with the importance of the objectives. Thirdly, a single solution is obtained at one time. Also, the layout designer based on his/her past experience randomly selects the layout having multiple objectives. This restricts the designing process completely designer dependent and thus, the layout varies from designer to designer. To overcome such difficulties, Pareto-optimality [5] has become an alternative to the classical weighted sum method. Previously, we proposed a Pareto-optimal based genetic approach for solving the multi-objective FLP [6]. Later we extend this approach by incorporating local search [7]. To our best knowledge, these are the first and only available approaches to find Pareto-optimal layouts for the multi-objective FLP.

Due to the combinatorial nature of the FLP, optimal algorithms are often found not to be suited for large FLP instances. Thus, the interest lies in the application of heuristic and meta-heuristic methods to solve large problems. Among these approaches, GA seems to become quite popular in solving FLP [3]. GA is an efficient, adaptive and robust search process for optimizing general combinatorial problems and therefore is a serious candidate for solving the FLP. However, there is also a limitation in applying GA to multi-objective FLP - GA can do global search in the entire space, but there is no way for exploring the search space within the convergence area generated by the GA loop. Therefore, it is sometimes insufficient in finding optimal solution for complex and large search space, which is very usual in real-world FLP. In certain cases, GA performs too slowly to be practical. A very successful way to improve the performance of GA is to hybridize it with local search techniques. Local search techniques are used to refine the solutions explored by GA by searching its vicinity for the fittest individuals and replacing it if a better one is found. However, hybridization with local search may degrade the global search ability of a multi-objective GA and require much computational time than conventional GA [8]. This is because most of the computation time is spent by local search.

This paper proposes an adaptive local search based GA for solving the multi-objective FLP to demonstrate the importance of finding a good balance between GA and local search technique. The proposed adaptive local search scheme automatically determines whether local search is used in a GA loop or not. The experimental results are compared to the results found by several existing and meta-heuristic approaches, and our previously proposed two Pareto-optimal based multi-objective evolutionary approaches [6], [7]. Computational experience shows that the proposed approach performs very well to improve the best fitness value, average number of generations, and convergence behaviors of average fitness values. Concurrently, it can find a set of Pareto-optimal layouts optimizing both quantitative and qualitative objectives simultaneously throughout the entire evolutionary process. The rest of the paper is organized as follows. In Section 2, the importance of Pareto-optimality in the FLP is introduced. Section 3 presents the related works. The justification of adaptive local search and the implementation details has been discussed in Section 4. Section 5 is to design the proposed approach. The proposed approach has been experimentally verified in Section 6. Finally, chapter 7 provides a conclusion of the paper.

2 Importance of Pareto-Optimality in FLP

Real-life scientific and engineering problems typically require the search for satisfactory solution for several objectives simultaneously. It is also common that conflicts exist among the objectives. In such problems, the absolute optimal solution is absent, and the designer must select a solution that offers the most profitable trade-off between the objectives instead. The objective of such MOOPs is to find a set of Pareto-optimal solutions, which are the solutions to which no other feasible solutions are superior in all objective functions. Historically, FLPs have been solved for either quantitative or qualitative goodness of the layout. In general, real-life FLPs are multi-objective by nature and they require the decision makers to consider a number of criteria involving quantitative and qualitative objectives before arriving at any conclusion. Thus, instead of offering a single solution, giving options and letting decision makers choose between them based on the current requirement is more realistic and appropriate.

3 Related Works

Several researches have been done in the FLP for the past few decades. These approaches can be divided into exact algorithms and heuristic algorithms. The exact methods such as the branch-and-bound and cutting plane algorithm have been successfully applied to FLP when the number of facilities is less than 15. However, problem of size larger than 15 cannot be solved optimally in reasonable time. Recently, meta-heuristic approaches such as simulated annealing (SA), GA, tabu search (TS) and ant colony optimization (ACO) have been successfully applied to solve large FLP. Among these approaches, GA has been shown to be effective in solving FLPs, especially in solving the large-scale problems [6]. However, research in this area typically focused on single objective. To date, there are only a few attempts to tackle the multi-objective FLP using GA. However, they used weighted sum method. Thus, they ignored the prospects of Pareto-optimal solutions in solving the multi-objective FLP. Interested readers should consult [3], [9] for a detailed review. In recent time,

various methods for hybridizing GA using conventional local search techniques have been suggested [10]. The combination of local search heuristics and GA is also a promising approach for finding near-optimal solutions to the FLPs [7], [11]. However, most of the local search techniques used in these approaches are applied to all individuals of the population within each generation of a GA without any generalization or analysis with respect to their convergence characteristics and reliabilities. For improving these limitations in the application of local search, a local search technique with adaptive scheme can be a better alternative.

4 Adaptive Local Search in Multi-objective FLP

Although GA has proven to be successful in producing quite good results when applied individually, near-optimal and top quality solutions seem to require the combined efforts of local search techniques and GA [10]. Local search algorithms are improvement heuristics that search in the neighborhood of the current solution for a better one until no further improvement can be made. If the search space is too large, GA has inherent difficulties to converge in the global optimum with an adequate precision. Local search techniques can remove this limitation by finding any local optimum with great precision, using information from the neighboring candidate solutions [12]. The synergy between both methods can therefore give rise to a family of hybrid algorithms, simultaneously global and precise. The GA globally explores the domain and finds a good set of initial estimates, while the local search further refines these solutions in order to locate the nearest, best solution. In order to apply local search, we have to specify an objective function to be optimized by the search. This specification is straightforward in a single objective optimization problem. Unfortunately, the situation in the MOOP is unclear since the number of such studies in the field has been small. A weighed sum of multiple objectives is often used for local search in hybrid multi-objective GA [8], which is not realistic. In this work, we extend the *1-opt* local search [12] for multi-objective FLP by incorporating domination strategy [5]. The general outline of this algorithm is given in Algorithm 1.

Algorithm 1. *1-opt* Local Search for MOFLP

```

set  $fit_1^1$  = MH cost of the current chromosome
set  $fit_2^1$  = CR score of the current chromosome
for  $i = 1$  to  $N$  do
repeat
    select another gene  $j$  randomly such that  $i \neq j$ 
    construct a new layout by swapping  $i$  and  $j$ 
    find  $fit_1^2$  = MH cost of the new layout
    find  $fit_2^2$  = CR cost of the new layout
    if the current layout is dominated by the new layout
        replace the current one with the new layout
    else
        continue with the current one
    end if
until any improvement in the current layout
end for

```

In most studies, local search is applied to the solutions of each generation of GA. As a result, the local search technique has to examine a large number of solutions for finding a locally optimum solution from each initial solution generated by genetic operations. It is nothing but mere waste of CPU time. For decreasing the computation time spent by local search, we implemented an adaptive local search technique that will only search around the convergence area produced by GA loop instead of applying to all individuals. This technique helps generating new individuals having certain high fitness values like the superior individuals generated by GA. The proposed adaptive local search scheme is based on similarity coefficient method (SCM) [10] to consider the similarity of individuals of a GA population. We modify the basic idea of SCM for multi-objective FLP environment. We can calculate the similarity coefficient SC_{pq} between two chromosomes p and q as follows:

$$SC_{pq} = \frac{\sum_{k=1}^n \partial(f_{pk}, f_{qk})}{n} \quad (1)$$

Where, k is the index of location in the layout, f_{pk} is the facility at location k in chromosome p , and n is the number of facilities.

$$\partial(f_{pk}, f_{qk}) = \begin{cases} 1, & \text{if } f_{pk} = f_{qk} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The average similarity coefficient (\bar{SC}) for all individuals can be expressed as follows:

$$\overline{SC} = \frac{\sum_{p=1}^{n-q} \sum_{q=p+1}^n SC_{pq}}{n} \quad (3)$$

Assuming that a pre-defined threshold value (β), the modified *l-opt* local search method described in Algorithm 1 will be automatically invoked in a GA loop by the following condition:

5 The Proposed Approach

Figure 1 presents the flowchart for a complete evolutionary cycle of the proposed multi-objective FLP using adapting local search. In this approach, we used the non-dominated sorting genetic algorithm 2 (NSGA 2) [13] as the multi-objective GA.

5.1 Chromosome Representation and Genetic Operators

In this study, the FLP has been presented as a quadratic assignment problem (QAP), which assigns n equal area facilities to the same number of locations. An integer string representation is used to represent the layouts. The solution is represented as a string of length n , where n is the number of facilities. The integers denote the facilities and their positions in the string denote the positions of the facilities in the layout. For crossover and mutation, we follow the concept described by Suresh *et. al.* [14] and swap mutation, respectively. Their implementation details can be found in [6].

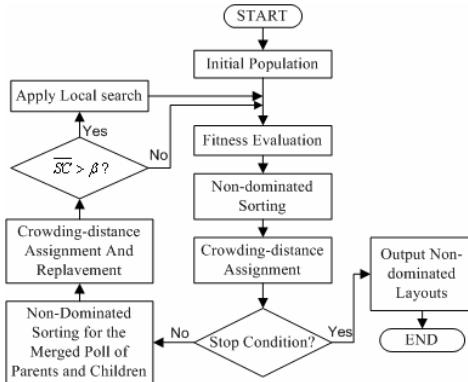


Fig. 1. Flowchart for the multi-objective FLP using adaptive local search

5.2 Fitness Function

We separately apply material handling (MH) costs and closeness relationship (CR) among various departments as quantitative and qualitative objective respectively. They can be expressed in the following mathematical model:

$$F_1 = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n f_{ik} d_{jl} X_{ij} X_{kl} \quad (5)$$

$$F_2 = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n W_{ijkl} X_{ij} X_{kl} \quad (6)$$

Subject to

$$\sum_{i=1}^n X_{ij} = 1, \quad j=1,2,\dots,n \quad (7)$$

$$\sum_{j=1}^n X_{ij} = 1, \quad i=1,2,\dots,n \quad (8)$$

$$X_{ij} = 0 \text{ or } 1 \quad (9)$$

$$W_{ijkl} = \begin{cases} r_{ik}, & \text{if locations } j \text{ and } l \text{ neighbours} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Where, i, k are facilities; j, l are locations in the layout; X_{ij} : 1 or 0 variable for locating i at j ; f_{ik} is the flow cost for unit distance from facility i to k ; d_{jl} is the distance from location j to l ; r_{ik} : CR value when departments i and k are neighbours with common boundary and n is the number of facilities in the layout. Here, the aim of the first fitness function is to minimize the MH costs (Eq.(5)). The second fitness function tries to maximize the adjacency requirement based on CR value (Eq.(6)).

6 Experimental Results and Analysis

The proposed algorithm is tested for benchmark instances taken from published literature [6]. The problems are composed of 6, 8, 12, 15, 20, 30, 42, 72 and 100 facilities.

We used the last digits to indicate the number of facilities in each problem. Very few benchmark problems are available for multi-objective FLP, particularly, in the case of CR score. Thus, we have ourselves created the test data sets for CR score where no such data exists. The experiments are conducted using 50 chromosomes and 40 generations for problems with up to 15 facilities; and 100 chromosomes and 80 generations for problems with more than 15 facilities. However, for justifying the convergence behaviour of the proposed adaptive local search, later we run the three multi-objective FLP approaches less than the mentioned generations. The probabilities of crossover, mutation and the pre-defined coefficient (β) are 0.9, 0.3, and 0.9 (90%), respectively. We use traditional tournament selection with tournament size of 2. Each benchmark problem is tested 30 times with different seeds. Then each of the final generations is combined and non-dominated sorting is performed to constitute the final non-dominated solutions.

Table 1. Comparison with existing algorithms for MH cost

Problem (naug n)	H63	H63-66	CRAFT	Biased Sampling	FLAC	DISCON	FATE	TAA	GESA	HU & Wang	Without LS	With LS	Best Known	Proposed Adaptive LS
6	43 (44.2)	43 (44.2)	43 (44.2)	43 (44.2)	43 (43)	43 (47.5)	50.6 (50.6)	43 (43)	43 (43)	43 (43)	43 (43)	43 (43)	43 (43)	43 (43)
8	109 (114.4)	107 (110.2)	107 (113.4)	107 (107)	107 (107)	107 (118.8)	126.7 (126.7)	116 (116)	107 (107)	107 (107.8)	107 (107.75)	107 (107.34)	107 (107)	107 (107)
12	301 (317.4)	304 (310.2)	289 (296.2)	289 (293)	289 (289)	295 (322.2)	326.2 (326.2)	314 (314)	289 (289.36)	289 (290.6)	287 (292.57)	287 (290.87)	287 (289.97)	287 (289.97)
15	617 (632.6)	578 (600.2)	583 (600)	575 (580.2)	585 (585)	597 (630.8)	660.8 (660.8)	596 (596)	575 (575.18)	575 (576.4)	575 (677.01)	575 (601.16)	575 (588.7)	573 (588.7)
20	1384 (1400.4)	1319 (1345)	1324 (1339)	1304 (1313)	1303 (1303)	1376 (1416.4)	1436.3 (1436.3)	1414 (1414)	1285 (1287.38)	1285 (1290.5)	1286 (1290.6)	1286 (1289.08)	1285 (1288.13)	1285 (1288.13)
30	3244 (3267.2)	3161 (3206.8)	3148 (3189.6)	3093 (3189.6)	3079 (3079)	3330 (3436.4)	3390.6 (3390.6)	3326 (33269)	3062 (3079.32)	3064 (3075.1)	3062 (3081.02)	3062 (3081.07)	3059 (3079.05)	3059 (3079.05)

Since almost all FLP algorithms try to optimize single criteria only (mainly minimizing the MH cost), first we compare the MH costs obtained by our approach with the existing single objective approaches. The values provided in Table 1 show the MH cost for the best layouts by some existing algorithms. From this table, it can be easily found that performance of the proposed approach is superior or equivalent to other approaches. Most importantly, the incorporation of adaptive local search helps the algorithm to achieve the new best solutions for *naug15* and *naug30*, and also to reduce the gaps between the best and the average MH costs.

Table 2 shows the performance statistics of the proposed adaptive local search based multi-objective FLP approach with the Pareto-optimal based multi-objective approach with conventional local search [7] and without local search [6] in the context of MH cost and CR score. Note that, for both single and multiple objectives, we used the same results obtained by our approach. The results shown in the table indicate that the adaptive local search based approach clearly outperforms the others.

Indeed, it achieves the new best MH cost for *naug20*, *naug30*, *sko42*, *sko72*, and *will100*. Also, in case of CR score, it finds better result for *ds6*, *ct9*, *naug20*, *naug30*, *sko72*, and *will100*. Furthermore, the average values for both objectives considerably improve. This can be more justified by Fig. 2, where the convergence behavior of the proposed and previous methods over generations for both objectives is depicted. From the figures, it can be found that the incorporation of adaptive local search reduces the gaps between the best and average values than that of the competing approaches.

Table 2. Comparison with exiting MOFLP approaches

Pr.	MH Cost		CR Score		Time (sec)	Optimal solutions after 50% of the generations
	Best	Avg	Best	Avg		
ds6	Adaptive LS	92	92	56	52.16	0.128
	With LS	92	94.96	48	44.24	0.139
	Without LS	96	96.8	48	43.40	0.128
ds8	Adaptive LS	179	200.04	82	76.34	0.208
	With LS	179	199.649	82	75.26	0.291
	Without LS	179	209.84	82	70.3	0.2
singh6	Adaptive LS	94	95.25	56	52.96	0.132
	With LS	94	96.66	56	51.305	0.138
	Without LS	94	98.28	48	40.48	0.132
singh8	Adaptive LS	179	187.6	82	73.1	0.192
	With LS	179	191.013	82	72.99	0.22
	Without LS	179	199.84	82	73.1	0.19
ct9	Adaptive LS	4818	4819.6	92	80.14	0.273
	With LS	4818	4820.03	90	77.09	0.315
	Without LS	4818	4822.9	90	74.79	0.27
naug20	Adaptive LS	1285	1288.13	202	187.25	15.814
	With LS	1286	1289.08	198	176.4	24.714
	Without LS	1286	1290.6	186	172.56	13.997
naug30	Adaptive LS	3059	3079.05	332	298.75	63.933
	With LS	3062	3081.07	312	272.36	96.42
	Without LS	3062	3081.02	292	254.05	56.025
sko42	Adaptive LS	15484	15863.42	372	346.12	235.287
	With LS	15642	16095.3	380	328.24	308.918
	Without LS	15796	16876.56	370	325.13	207.13
sko72	Adaptive LS	65002	65946.2	604	576.5	1750.993
	With LS	65544	66086.3	592	518.15	1942.56
	Without LS	66034	67658.33	602	536.8	1605.96
Will100	Adaptive LS	271862	276954.52	1128	1017.72	3960.21
	With LS	273862	279073.12	1109	979.25	4864.65
	Without LS	273988	280126.62	1084	977.42	3804.84

The required time for a complete evolutionary cycle mentioned in Table 2 also shows that the proposed method is able to optimize both MH cost (minimize) and CR score (maximize) from the first to the last generation faster than the others. From the table, we can find that the adaptive local search based approach takes less time than the conventional local search based approach, and the difference is very significant. For obvious reason, the proposed approach takes slightly more time than the approach

without local search. Despite that it is very important to note that the performance of our proposed approach with adaptive local search is much better than the other two approaches. In fact, after performing half of the scheduled generations, the proposed approach finds the best values for both objectives in case of more than 50% of the populations for all test problems. Where as, at this point the performances of the other two approaches are not satisfactory enough. They can find the best values only for the problems with 6 facilities and the number is also small. Table 2 also summarizes the percentages of the optimal solutions obtained by each algorithm.

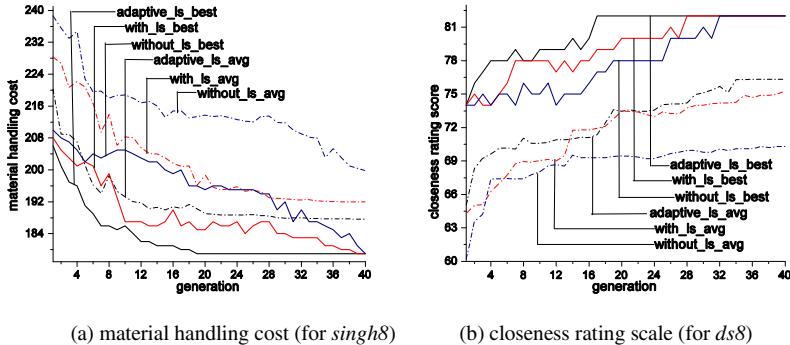


Fig. 2. Two objectives over generations

As mentioned earlier, we run the proposed algorithm for 60% of the scheduled generations for all test problems to test their convergence behaviors. The experimental results suggest that after 50% of the scheduled generations, it starts finding the known best values and it almost converges for around 60% of the total population. Fig. 2 also shows this tendency. For this reason, the required time for the proposed approach will be less than the values mentioned in Table 2. However, for fair comparison, we mentioned the time for the same number of generations for all approaches. Thus, the proposed adaptive local search appears to be highly effective, and the additional coding effort and time required in comparison to the approach without local search is definitely justified.

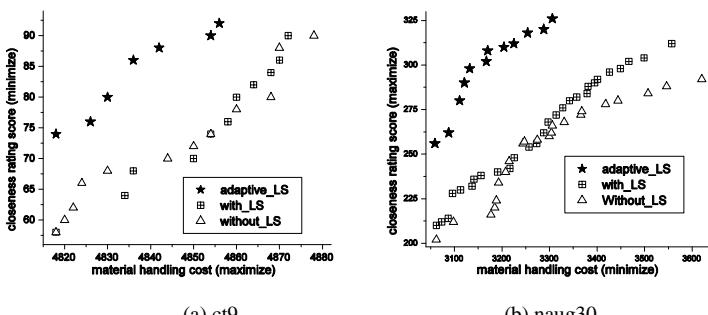


Fig. 3. Final Pareto-optimal front

MOEAs do not try to find one optimal solution but all the trade-off solutions, and deal with two goals: convergence and diversity. Non-dominated solutions of the final generation obtained by each alternative are shown in Fig. 3 to illustrate the convergence and diversity of the layouts. It is worthwhile to mention that in all cases, most of the solutions of the final population are Pareto-optimal. In the figures, the occurrences of the same non-dominated solutions are plotted only once. From these figures, it can be observed that the final solutions produced by the proposed method are more spread and well converged than previous approaches. And for this reason, it is capable of finding extreme solutions.

7 Conclusion

In this paper, we propose an adaptive local search based evolutionary approach for solving the multi-objective FLP, which is very rare in the literature. The proposed approach is composed of the multi-objective GA, the improved *l-opt* local search method and the modified adaptive local search scheme using SCM. Unlike conventional local search method, the adaptive local search scheme automatically determines whether local search is used in a GA loop or not. A comparative analysis clearly show that the introduction of adaptive local search helps well to improve the best fitness value, average number of generations, and convergence behaviors of fitness values. In addition, it can find the near-optimal and non-dominated layout solutions, which are also the best-known results especially when applied to larger instances ($n > 15$). Furthermore, it is capable of finding a set of Pareto-optimal layout solutions that optimizes both MH cost and CR score simultaneously throughout the entire evolutionary process and provides a wide range of alternative layout choices for the designers.

References

1. Tompkins, A.: Facilities Planning, 3rd edn. John Wiley & Sons, New York (2003)
2. Singh, S.P., Singh, V.K.: An Improved Heuristic Approach for Multi-Objective Facility Layout Problem. Int. J. Prod. Res. 48(4), 1171–1194 (2010)
3. Drira, A., Pierreval, H., Hajri-Gabouj, S.: Facility Layout Problems: A survey. Annual Rev. Con. 31, 255–267 (2007)
4. Grosan, C., Abraham, A., Tigan, S., Chang, T.G.: How to Solve a Multicriterion Problem for Which Pareto Dominance Relationship Cannot Be Applied? A Case Study from Medicine. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4253, pp. 1128–1135. Springer, Heidelberg (2006)
5. Pareto, V.: Cours D'Economie Politique, Rouge, Lausanne, Switzerland (1896)
6. Ripon, K.S.N., Glette, K., Mirmotahari, O., Høvin, M., Tørresen, J.: Pareto Optimal Based Evolutionary Approach for Solving Multi-Objective Facility Layout Problem. In: Chan, J.H. (ed.) ICONIP 2009, Part II. LNCS, vol. 5864, pp. 159–168. Springer, Heidelberg (2009)
7. Ripon, K.S.N., Glette, K., Høvin, M., Tørresen, J.: Multi-Objective Evolutionary Approach for Solving Facility Layout Problem using Local Search. In: 25th ACM Symposium On Applied Computing, Switzerland, pp. 1155–1156 (2010)

8. Ishibuchi, H., Yoshida, T.: Implementation of Local Search in Hybrid Multi-Objective Genetic Algorithms: A Case Study on Flowshop Scheduling. In: 4th Asia-Pacific Conference on Simulated Evolution and Learning, Singapore, vol. 1, pp. 193–197 (2002)
9. Singh, S.P., Sharma, R.R.K.: A Review of Different Approaches to the Facility Layout Problems. *Int. J. Adv. Manuf. Technol.* 30, 425–433 (2006)
10. Yun, Y., Moon, C., Kim, D.: Hybrid Genetic Algorithm with Adaptive Local Search Scheme for Solving Multistage-Based Supply Chain Problems. *Comp. & Ind. Eng.* 56, 821–838 (2009)
11. Ye, M., Zhouz, G.: A Local Genetic Approach to Multi-Objective Facility Layout Problems with Fixed Aisles. *Int. J. Prod. Res.* 45(22), 5243–5264 (2007)
12. Merz, P., Freisleben, B.: Greedy and Local Search Heuristics for Unconstrained Binary Quadratic Programming. *J. of Heur.* 8, 197–213 (2002)
13. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evo. Comp.* 6(2), 182–197 (2002)
14. Suresh, G., Vinod, V.V., Sahu, S.: A Genetic Algorithm for Facility Layout. *Int. J. Prod. Res.* 33(12), 3411–3423 (1995)

Non-uniform Layered Clustering for Ensemble Classifier Generation and Optimality

Ashfaqur Rahman¹, Brijesh Verma¹, and Xin Yao²

¹ CINS, CQUni, Australia

{a.rahman, b.verma}@cqu.edu.au

² CERCIA, University of Birmingham, United Kingdom

x.yao@cs.bham.ac.uk

Abstract. In this paper we present an approach to generate ensemble of classifiers using non-uniform layered clustering. In the proposed approach the dataset is partitioned into variable number of clusters at different layers. A set of base classifiers is trained on the clusters at different layers. The decision on a pattern at each layer is obtained from the classifier trained on the nearest cluster and the decisions from the different layers are fused using majority voting to obtain the final verdict. The proposed approach provides a mechanism to obtain the optimal number of layers and clusters using a Genetic Algorithm. Clustering identifies difficult-to-classify patterns and layered non-uniform clustering approach brings in diversity among the base classifiers at different layers. The proposed method performs relatively better than the other state-of-art ensemble classifier generation methods as evidenced from the experimental results.

Keywords: ensemble classifier, genetic algorithm, optimal clustering.

1 Introduction

An ensemble classifier refers to a collection of base classifiers. The base classifiers learn the decision boundaries on the training patterns. Their decisions on a pattern are combined to reach the final classification verdict. The training process in an ensemble classifier aims to produce the base classifiers in way so that they differ from each other in terms of the errors they make on identical patterns. This phenomenon is known as diversity [1]–[3]. The fusion methods on the other hand explore ways to merge the decisions from the base classifiers into a final verdict.

Bootstrap aggregating or *bagging* [4] is one of the earliest ensemble classifier generation methods. Diversity in bagging is achieved by training the base classifiers on different randomly drawn (with replacement) subsets of the training data. Bagging and its variants [5][6] provide a mechanism to achieve diversity but does not mention any mechanism to identify difficult-to-classify overlapping patterns that leaves space for improvement. *Boosting* [7][8] achieves diversity by creating data subsets by resampling the training examples, however, by providing the training examples that are badly classified for the consecutive classifier in the next iteration. The subsets in boosting, however, not necessarily contain examples that are difficult to classify when combined together. A number of variants of boosting can be observed in the literature including Learn++-NC [9] and heterogeneous boosting [10].

A data set can contain multiple clusters within. Some clusters are easy to classify and some are not. A clear identification of these difficult-to-classify clusters can better guide the learning process. This philosophy is adopted by the *clustered ensembles* [11]–[13] that identify multiple clusters in classified data and train base classifiers on them. A pattern is classified by finding the appropriate cluster and using the corresponding base classifier to predict the class. A pattern in clustered ensembles can belong to one cluster only and as a result the decision can be obtained from a single classifier leading to lack of diversity.

We aim to address this issue by using overlapping clustering. In order to achieve diversity the data set can be independently partitioned n times into variable number of clusters and identical patterns will belong to n alternate clusters. Let n layers to refer to n alternative clustering of the data set. The decision provided by the base classifiers trained on the n non-uniform clusters at n layers can be fused to obtain the final verdict on the pattern. With clustering we can generate the base classifiers and with layers we can achieve the diversity. Note that the number of layers at which maximum diversity is achieved depends on the characteristics of the data set and needs to be optimized. We have adopted a Genetic Algorithm based approach for identifying the optimal number of layers.

The research presented in this paper is based on the above philosophy and aims to: (i) develop a novel method for generating ensemble of classifiers using non-uniform cluster layers and optimizing the number of layers, (ii) investigate the impact of number of layers on classification accuracy, and (iii) obtain a comparative analysis on how well the proposed approach performs compared to the commonly used approaches for ensemble classifier generation.

2 Non-uniform Layered Cluster Oriented Ensemble Classifier

The overall architecture of the proposed ensemble classifier is presented in Fig 1. Let f refers to the ensemble classifier. Given a pattern \mathbf{x} , the ensemble classifier maps the pattern to a class c as

$$f(\mathbf{x}) = c \quad (1)$$

where $c \in \{1, 2, \dots, N_c\}$ and N_c is the number of classes in the data set. Let the data set is partitioned into L different layers. The decision of the ensemble classifier is composed of the discrete decisions produced at the L layers. Representing the decision at layer l as $f_c(\mathbf{x})$, $f(\mathbf{x})$ can be expressed as

$$f(\mathbf{x}) = \text{mode}(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_L(\mathbf{x})) \quad (2)$$

where *mode* is the statistical mode function representing the majority voting decision fusion function.

Now assume that K_l refers to the number of clusters at layer l . The k -th cluster in layer l is referred to as $C_{K_l,k}$ and the corresponding cluster centre as $\theta_{K_l,k}$. The cluster id η for a pattern \mathbf{x} at layer l is obtained based on the Euclidian distance ξ as

$$\eta_{x,l} = \operatorname{argmin}_k \xi(x, \theta_{K_l,k}) \quad (3)$$

where $1 \leq k \leq K_l$.

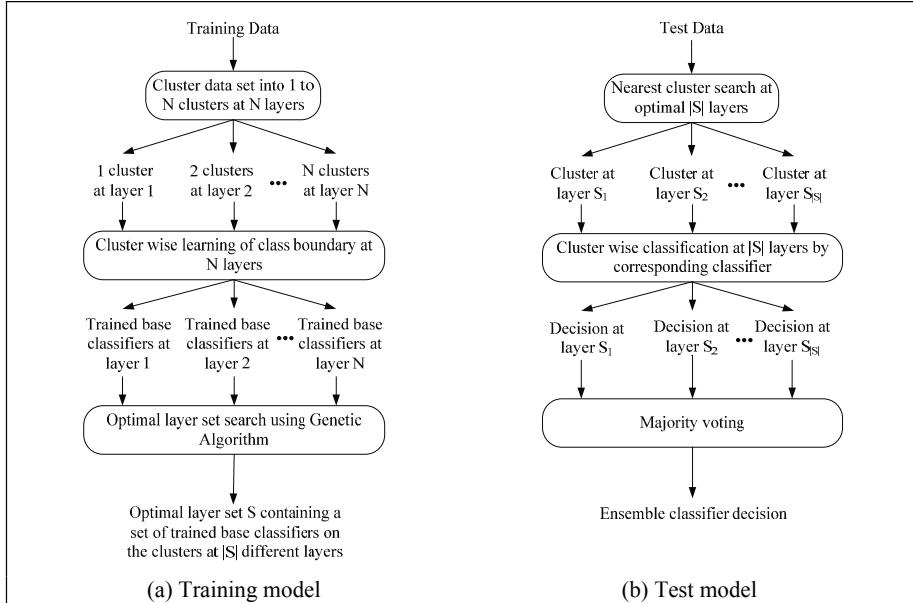


Fig. 1. Architecture of the proposed ensemble classifier

Let $\varphi_{K_l,k}$ be the base classifier trained on the corresponding cluster $C_{K_l,k}$. The pattern x belongs to cluster $C_{K_l,\eta_{x,l}}$ at layer l and the corresponding base classifier $\varphi_{K_l,\eta_{x,l}}$ gives a class decision c_l at layer l as

$$\varphi_{K_l,\eta_{x,l}}(x) = c_l, \quad (4)$$

where $c_l \in \{1, 2, \dots, N_c\}$ and N_c is the number of classes in the data set. A pattern can belong to one cluster only at a layer and thus we can express $f_l(x)$ as

$$f_l(x) = \varphi_{K_l,\eta_{x,l}}(x). \quad (5)$$

Based on (1) to (5) the ensemble classifier mapping process can be summarized as

$$\begin{aligned} f(x) &= \text{mode}(f_1(x), f_2(x), \dots, f_L(x)) \\ &= \text{mode}(\varphi_{K_1,\eta_{x,1}}(x), \varphi_{K_2,\eta_{x,2}}(x), \dots, \varphi_{K_L,\eta_{x,L}}(x)) \\ &= \text{mode}(c_1, c_2, \dots, c_L) \\ &= c \end{aligned} \quad (6)$$

As the numbers of clusters are variable at different layers, the proposed model is called Non-Uniform Layered Cluster Oriented Ensemble Classifier (NULCOEC). The performance of NULCOEC depends on a number of parameters including number of layers L and number of clusters K_l at layer l where $1 \leq l \leq L$. The technique used in this paper for optimizing L and $\{K_l\}$ is presented next.

The aim of designing an ensemble classifier is to obtain better classification performance while achieving diversity. This is why we used the optimality criteria of

accuracy \times diversity while searching for the optimal L and $\{K_l\}$. Accuracy is computed as the average across the positive diagonal of the confusion matrix obtained by the ensemble classifier on the training set. We have computed diversity of the ensemble classifier using Kohavi–Wolpert (KW) variance [2]. Given a set of $|\Gamma|$ examples $\{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_{|\Gamma|}, t_{|\Gamma|})\}$ in the training set, KW variance over L layers is computed as

$$KW = \frac{1}{|\Gamma| \times L} \sum_{j=1}^{|\Gamma|} D_l(\mathbf{x}_j) \times (L - D_l(\mathbf{x}_j)) \quad (7)$$

where L is the number of layers, and D_l is computed as

$$D_l(\mathbf{x}_j) = \begin{cases} 1 & \text{if } \mathbf{x}_j \text{ classified correctly} \\ 0 & \text{if } \mathbf{x}_j \text{ classified incorrectly} \end{cases} \quad (8)$$

Before applying any search algorithm it is required to setup a search space. We used hierarchical clustering algorithm for this purpose. The problem of applying k -means clustering algorithm is that the cluster contents can change for the same k depending on the initialization of the cluster centres [14]. The search space creation algorithm is presented in Fig 2. Note that the search space means a set of trained base classifiers on clusters at different layers. Each level in the tree represents a possible contender layer in the ensemble classifier. At level $|\Gamma|$ each example in the training set is in a separate cluster whereas at level one all the examples in the training set are in a single cluster. The search algorithm will look for a set of levels from this tree so that their combination maximizes the optimization criteria.

Algorithm. Create Search Space

Input: Training set with $|\Gamma|$ examples

Output: Search Space Tree

- Step 1. $l=|\Gamma|$
- Step 2. Put each example in the training set in separate clusters at level l . Each cluster contains one pattern only and the class name is memorized.
- Step 3. $l=l-1$
- Step 4. Merge two closest neighbour clusters at level l . If the newly created cluster contains patterns of a unique class memorize the class label, otherwise train a classifier.
- Step 5. If $l>1$ GOTO Step 3

Fig. 2. Algorithm for creating the search space

We have used Genetic Algorithm (GA) for searching the optimal L and $\{K_l\}$. The chromosomes have a length of $|\Gamma|$ and each gene in the chromosome represents a possible level in the search tree. GA starts with a large set (population) of randomly generated chromosomes. Each chromosome in the current population is decoded to obtain the levels. Classification of the training data is performed by using the corresponding base classifiers in these levels and combining their decisions. The fitness of

the chromosome is set to accuracy \times diversity. A global variable stores the chromosome with maximum fitness. A set of fit chromosomes are then used to fill in a mating pool. The crossover operator randomly selects two parent chromosomes from the mating pool and interchanges a portion of the gene stream between them to generate the offspring. A mutation operator is then applied on the offspring where every gene (bit) in a chromosome is altered with a certain probability. These newly generated chromosomes constitute the population for the next generation. GA stops when there is no improvement of fitness for a consecutive number of generations.

3 Results and Discussion

We have conducted a number of experiments on benchmark data sets as used in contemporary research works from the UCI Machine Learning Repository [15]. A summary of the data sets is presented in Table 1. In all experiments 50% of the data set was used for training and the remaining 50% for the testing. The genetic algorithm was applied with the following set of parameters: (i) population size = 50, (ii) mutation probability 0.001, and (iii) termination try count = 10. The classification results for bagging and boosting are obtained using WEKA [16] on identical training and test sets. SVM with radial basis kernel was used as the base classifier and the libsvm library [17] was used. All the experiments on the proposed method were conducted on MATLAB 7.5.0.

Table 1. Data sets used in the experiments

Dataset	# instances	# attributes	# classes
Glass	214	10	7
Ionosphere	351	33	2
Breast Cancer	699	9	2
Sonar	208	60	2
Thyroid	215	5	3
Wine	178	13	3

3.1 Classification with Non-uniform Layered Clustering

Fig 3 presents the diversity achieved on different data sets in Table 1. Each data set is partitioned into one to 50 clusters at 50 layers. Their decisions from the different layers are fused using majority voting. It can be observed that the diversity increases as the number of layers go up. However the maximum diversity is achieved at different number of layers for different data sets and once the maximum is achieved the diversity falls with increasing number of layers. The diversity falls once the maximum is reached for *Glass*, *Sonar*, *Thyroid*, and *Wine* data sets. This indicates the necessity for adopting some way of optimizing the number of layers.

Fig 4 presents the training set classification accuracies obtained on the data sets using identical experimental procedure. Note that the best classification accuracy is obtained at more than one layer. However the maximum accuracy is achieved at different number of layers for different data sets and thus requires some optimization process. Also note from Fig 3 and Fig 4 that the maximum accuracy may not always correspond to maximum diversity. For example, all the layers on a pattern provide the correct

decision. In this case the classification accuracy is at its maximum whereas diversity is zero. We are thus motivated to obtain an optimal parameter setting based on the criteria accuracy \times diversity to keep a balance between accuracy and diversity.

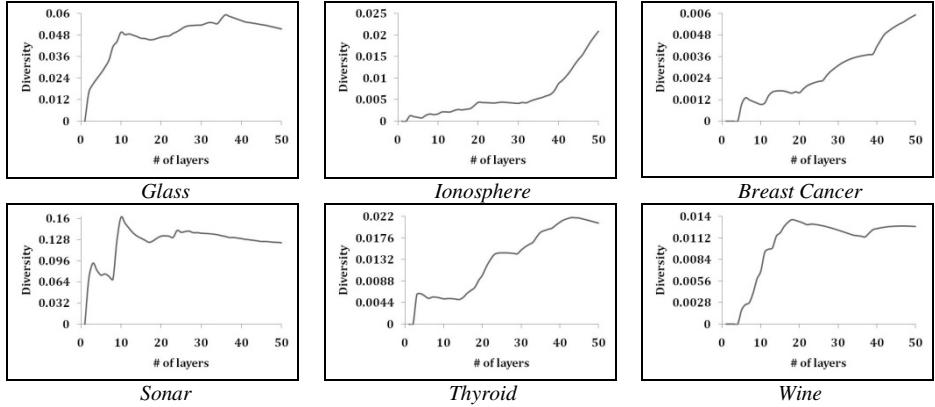


Fig. 3. Diversity as the number of layers changes

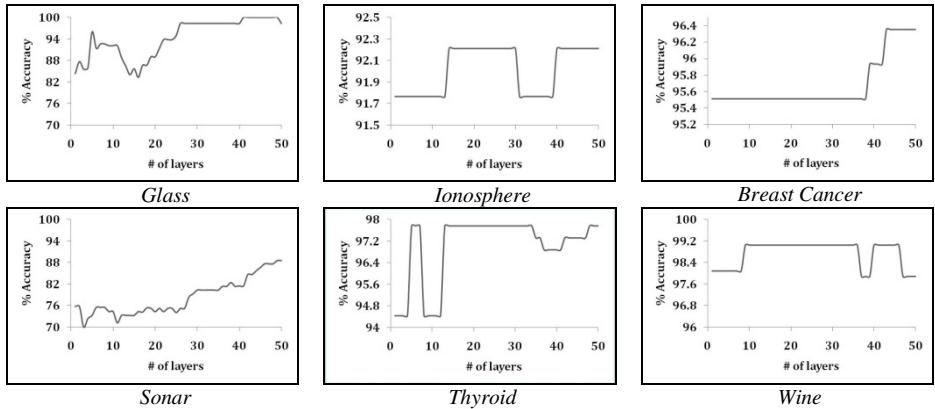


Fig. 4. Classification accuracy as the number of layers changes

3.2 Classification at Optimal Layers and Clusters

This section presents the results on the optimality of layers and clusters. The results presented in Section 3.1 are different from the one presented in this section. While computing the optimal number of clusters, note that the length of the GA chromosome is equal to the number of examples in the training set. All the experiments in Section 3.1 are done with a maximum of 50 clusters for all data sets. Moreover, the GA chromosome can be composed of a random sequence of zeros and ones. Thus L layers in this section do not necessarily imply the partitioning of a data set into $1, 2, \dots, L$ consecutive layers as in the previous section. The results in this section are more robust as it considers combinations of layers from non-consecutive levels in the search tree.

Table 2 presents the optimal number of layers and clusters and the corresponding diversity and accuracy obtained using diversity×accuracy optimization criteria. It is worth noting the variability of number of layers for each data set at which maximum performance is achieved. In general the learning algorithms of unique classifiers consider the entire data set as a single cluster. Note that the optimal performance of *Glass* and *Sonar* data set is obtained without considering the classifier trained at layer one. Overall the proposed optimality criterion keeps a balance between accuracy and diversity.

Table 2. Diversity and accuracy at optimal number of layers and clusters with accuracy×diversity optimality criteria

Data Set	Diversity	Training Accuracy	Test Accuracy	No. of layers	Min clusters	Max clusters
<i>Glass</i>	0.0565	98.33	91.39	29	2	69
<i>Ionosphere</i>	0.0470	95.83	88.17	54	1	172
<i>Breast Cancer</i>	0.0130	98.90	96.37	111	1	315
<i>Sonar</i>	0.1328	100	82.78	30	6	103
<i>Thyroid</i>	0.0242	97.33	92.44	23	1	84
<i>Wine</i>	0.0151	99.05	97.69	24	1	83

3.3 Comparative Analysis

Table 3 provides a comparative analysis between the proposed and existing ensemble classifier generation methods based on classification performance on the test sets. NULCOEC using accuracy×diversity optimization criteria performs 16.20% better than bagging and 3.55% better than boosting. The better performance of the proposed technique can be attributed to two reasons: (i) identification of regions in the data set that are difficult to classify and thus deployment of experts to precisely learn that region, and (ii) use of non-uniform layered clustering to achieve diversity.

Table 3. A comparative analysis of test set accuracies between the proposed and existing ensemble classifier generation methods

Data Set	Bagging	Boosting (AdaBoost M1)	NULCOEC
<i>Glass</i>	60.87	87.51	91.39
<i>Ionosphere</i>	80.33	80.24	88.17
<i>Breast Cancer</i>	96.37	97.62	96.37
<i>Sonar</i>	76.91	80.74	82.78
<i>Thyroid</i>	72.22	88	92.44
<i>Wine</i>	96.11	97.04	97.69

4 Conclusions

In this paper we have presented a novel approach towards generating ensemble of classifiers using non-uniform clustering at multiple layers. Partitioning a data set into variable number of clusters ensures that patterns belong to different clusters at different layers and as a result cluster wise learning at layers leads to diversity. There is no

straightforward correlation between number of layers and optimal classification accuracy. The proposed approach thus provides a GA based algorithm to obtain optimal number of layers and clusters for each data set using accuracy×diversity optimization criteria. NULCEOC performs 16.20% better than bagging and 3.55% better than boosting in terms of average classification accuracy.

An interesting observation is that the number of patterns in each cluster is very small when partitioned into a large number of clusters. This is sometimes insufficient for accurate learning of the base classifiers. Moreover at some clusters the number of patterns for a particular class is relatively small leading to data imbalance. This results in poor classification of the patterns in that class. In future we intend to investigate these issues in order to improve the overall classification performance of NULCOEC.

References

1. Tang, E.K., Suganthan, P.N., Yao, X.: An analysis of diversity measures. *Machine Learning* 65, 247–271 (2006)
2. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning* 51(2), 181–207 (2003)
3. Scherbart, A., Nattkemper, T.W.: The diversity of regression ensembles combining bagging and random subspace method. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008. LNCS, vol. 5507, pp. 911–918. Springer, Heidelberg (2009)
4. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
5. Chen, L., Kamel, M.S.: A generalized adaptive ensemble generation and aggregation approach for multiple classifiers systems. *Pattern Recognition* 42, 629–644 (2009)
6. Nanni, L., Lumini, A.: Fuzzy bagging: a novel ensemble of classifiers. *Pattern Recognition* 39, 488–490 (2006)
7. Schapire, R.E.: The strength of weak learnability. *Machine Learning* 5(2), 197–227 (1990)
8. Freund, Y., Schapire, R.E.: Decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119–139 (1997)
9. Muhlbaier, M.D., Topalis, A., Polikar, R.: Learn++. NC: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes. *IEEE Trans. on Neural Networks* 20(1), 152–168 (2009)
10. Nascimento, D.S.C., Coelho, A.L.V.: Ensembling heterogeneous learning models with boosting. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009. LNCS, vol. 5863, pp. 512–519. Springer, Heidelberg (2009)
11. Rokach, L., Maimon, O., Lavi, I.: Space decomposition in data mining: a clustering approach. In: Int. Symp. on Methodologies for Intelligent Systems, Maebashi, Japan, pp. 24–31 (2003)
12. Kuncheva, L.I.: Switching between selection and fusion in combining classifiers: an experiment. *IEEE Trans on Systems, Man and Cybernetics* 32(2), 146–156 (2002)
13. Nasierding, G., Tsoumakas, G., Kouzani, A.Z.: Clustering based multi-label classification for image annotation and retrieval. In: IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 4514–4519 (2009)
14. Mulder, W.D., Schliebs, S., Boel, R., Kuiper, M.: Initialization dependence of clustering algorithms. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008. LNCS, vol. 5507, pp. 615–622. Springer, Heidelberg (2009)
15. UCI Machine Learning Database, <http://archive.ics.uci.edu/ml/>
16. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *SIGKDD Explorations* 11(1) (2009)
17. LIBSVM, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Membership Enhancement with Exponential Fuzzy Clustering for Collaborative Filtering

Kiatichai Treerattanapitak and Chuleerat Jaruskulchai

Department of Computer Science, Kasetsart University
50 Phahon Yothin Rd, Chatuchak Bangkok Thailand
kiat@smartzap.com, fscichj@ku.ac.th

Abstract. In Recommendation System, Collaborative Filtering by Clustering is a technique to predict interesting items from users with similar preferences. However, misleading prediction could be taken place by items with very rare ratings. These missing data could be considered as noise and influence the cluster's centroid by shifting its position. To overcome this issue, we proposed a new novel fuzzy algorithm that formulated objective function with Exponential equation (XFCM) in order to enhance ability to assign degree of membership. XFCM embeds noise filtering and produces membership for noisy data differently to other Fuzzy Clustering. Thus the centroid is robust in the noisy environment. The experiments on Collaborative Filtering dataset show that centroid produced by XFCM is robust by the improvement of prediction accuracy 6.12% over Fuzzy C-Mean (FCM) and 9.14% over Entropy based Fuzzy C-Mean (FCME).

Keywords: Exponential, Collaborative Filtering, Noisem Clustering.

1 Introduction

Collaborative Filtering (CF) is a technique implemented in Recommendation System. It plays important role behind the successful of many well-known commercial websites such as Amazon.com. The main goal of CF is to predict user interested from peer's opinions and offer proper products that match particular users interested. There are several techniques have been used to produce output in CF but one of the most popular technique is Memory based method. Either by User oriented [1] or Item oriented [2], they are easy to implement. In the process of prediction, similar objects to the target objects are used as equivalent to a group of peer's opinions. Objects are similar if similarity calculates (usually mathematical calculation in term of Residual Sum Square) to the target is above the threshold. Unknown ratings then calculate based on similarity weight between the target and all similar objects.

However because of some specific natures of the data, they lead to the low accuracy of the prediction. First, the users do not always give the rating to all products; they rate only the products they previously used. Then, the formation of user's profile of their rating contains a lot of missing value thus the users-items matrix is very sparse. Second, the users and items matrix is always large. The formation of user's profile contains very high dimensions vector thus users-items matrix is very huge.

Similarity calculation on high dimensions and huge matrix produces high level of dispersion hence the prediction is poor. In addition, the new items or new users that introduce to the system, they both contain a lot of missing value even more than usual. In this case, the prediction calculation is impossible when there is no similar object or no neighbor to compute. However this problem will not impact to clustering based method because the prediction is calculated toward cluster's centroid which contains no missing value.

In Data Mining, Clustering is an Unsupervised Learning with objective to separate unlabelled data into finite and discrete set. In general, each data is formulated into a vector by its associated features. The vectors are then used to distinguish by the similarity calculation in term of mathematics. There are several clustering techniques but Fuzzy C-Means is much more appropriate to CF problem because one object or product in E-Commerce could be classified into multiple categories. There are several studies that implement clustering algorithm on CF domain such as Kim modified K-Means algorithm for CF domain [3]. Treerattanapitak improved Item-based Collaborative Filtering by combining with Fuzzy C-Means [4] and extended by integrating FCM with Entropy Regularization [5]. Jinlong modified objective function to minimize Root Means Square Error (RMSE) instead of minimize the sum of distance as usual in Fuzzy C-Means [6]. However FCM is sensitive to noise whereby centroid is influenced to shift the position.

In this paper we proposed a new novel Exponential Fuzzy Clustering Algorithm (XFCM) with single parameter to robust the centroid in the noisy environment by enhancement the membership degree to deal with the mentioned issues. This paper organizes as follow. In Section 2, we review Fuzzy Clustering and address the centroid shift problem. In Section 3, we propose new Exponential Fuzzy Clustering with the outlier detection. In Section 4, we evaluate Exponential Fuzzy Clustering on the real dataset. In Section 5, the conclusion is drawn with our future works.

2 Fuzzy Clustering: A Revisited

The Fuzzy approach involves uncertainty i.e. the data could not belong to one cluster nevertheless all data belong to all clusters but in the different degree of membership. Fuzzy Clustering partitions data into k clusters by distance measurement between data (x_i) and the cluster's center or centroid (v_j) of the vector size M ($m=1..M$). Each data (x_i) belong to each cluster j at the different degree of membership (μ_{ij}) between [0,1] and sum of membership to all cluster is one. The distance function for similarity measurement ($d(x_i, v_j)$) is normally Euclidean distance function.

2.1 Fuzzy C-Means (FCM)

FCM tries to minimize the objective function (J_{FCM}) in (1) by iterative the procedure to partition.

$$J_{FCM} = \sum_{j=1}^k \sum_{i=1}^N \mu_{ij}^m d(x_i, v_j)^2, m \in (1, \infty), \sum_{j=1}^k \mu_{ij} = 1. \quad (1)$$

In order to minimize J_{FCM} , Lagrange Multiplier was used to find solution for μ_{ij} and v_i . They are computed accordingly to (2) and (3).

$$\mu_{ij} = \frac{1}{\sum_{u=1}^k \left(\frac{d(x_i, v_j)^2}{d(x_i, v_u)^2} \right)^{\frac{1}{m-1}}} \quad (2)$$

$$v_j = \frac{\sum_{i=1}^N (\mu_{ij}^m x_i)}{\sum_{i=1}^N \mu_{ij}^m}; 1 \leq j \leq k. \quad (3)$$

FCM terminates by validating objective function changed between iteration t and previous iteration $t-1$. The algorithm is terminated when $|J_{FCM}^t - J_{FCM}^{t-1}| < \epsilon$. where termination coefficient (ϵ) is a small value.

2.2 Fuzzy C-Means with Entropy Regularization (FCME)

Fuzzy C-Means with Entropy Regularization [7][8][20] is an improve version of FCM by employing Entropy term from Information Theory to the objective function as (4).

$$J_{FCME} = \sum_{j=1}^k \sum_{i=1}^N \mu_{ij} d(x_i, v_j)^2 + \lambda \sum_{j=1}^k \sum_{i=1}^N \mu_{ij} \log \mu_{ij}, \lambda > 0; \sum_{j=1}^k \mu_{ij} = 1 \quad (4)$$

Again, Lagrange Multiplier was used to find solution. μ_{ij} and v_i are displayed in (5) and (6).

$$\mu_{ij} = \frac{\exp(-\lambda \cdot d(x_i, v_j)^2)}{\sum_{u=1}^k \exp(-\lambda \cdot d(x_i, v_u)^2)} \quad (5)$$

$$v_j = \frac{\sum_{i=1}^N (\mu_{ij} x_i)}{\sum_{i=1}^N \mu_{ij}}; 1 \leq j \leq k \quad (6)$$

The algorithm tries to minimize the objective function by minimizing within cluster dispersion and maximizing the negative entropy. The λ parameter is used to control the effect of entropy term. When λ is very large, the algorithm assigns data to a single cluster which behaves similar to K-Means.

2.3 Curse of Noise

The noise vectors may contain high amplitude value or a lot of missing data in the vector's element. Even if at the low degree of membership, it could significant contribute to the centroid because of all data is taken into account in the updated centroid

equations (3) and (6). Therefore, the noisy data could have high level of degree of membership as ordinary data as in Fig. 1. There are several workarounds to deal with the noise as follow.

Detection approach. Noisy data usually locates at the boundary. There are several methods used to detect them. Statistical based technique [9] but this method does not work well in large dataset. Distance based technique [10][11] detect outlier based on top n points distance from the centroid. Deviation-based [12][13] identifies outliers by examining the main characteristics of objects in a group.

Clustering approach. It regards small clusters as outliers or identified outliers by removing clusters from the original dataset. Thus, clustering algorithms can detect outliers as a by-product of the clustering process [14][15].

Possibilistic approach. This approach breaks the normal probabilistic constraint that limit the degree of membership for one particular data to all clusters must be sum to one [16][17][18][19].

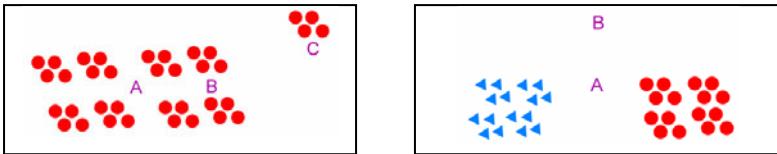


Fig. 1. Left picture displays the centroid drifts away from point A to B because there is a group of noisy data at point C. Right picture illustrates the degree of membership of point A and B is about 0.5 even if point A is a good member while point B is a poor member.

3 Exponential Fuzzy Clustering (XFCM)

Exponential Fuzzy Clustering is formulated in exponential to widen the gap between good and poor data. We formulate the objective function under some constraints then resolve it to find the optimum solution. There are two constraints to set the objective function J . The first condition, $\mu_{ij}=0$, data is not belong to the cluster then J must be 0. The second condition, $\mu_{ij}=1$, data is belong to the cluster then J must be $d(x_i, v_j)^2$. Exponential that satisfies both constraints is formulated as (7). We could find the solution by minimizing the objective function (7) subject to $\sum_{j=1}^k \mu_{ij} = 1$ with Lagrange Multipliers.

$$J_{XFCM} = \sum_{j=1}^k \sum_{i=1}^N \left(\frac{m^{\mu_{ij}} - 1}{m - 1} \right) d(x_i, v_j)^2, m \in (1, \infty), \sum_{j=1}^k \mu_{ij} = 1 \quad (7)$$

We first introduce the Lagrange Multiplier λ_i , $i=1, \dots, N$ to yield Lagrange function in (8)

$$L = \sum_{j=1}^k \sum_{i=1}^N \left(\frac{m^{\mu_{ij}} - 1}{m - 1} \right) d(x_i, v_j)^2 + \sum_{i=1}^N \lambda_i \left(\sum_{j=1}^k \mu_{ij} - 1 \right) \quad (8)$$

Differentiate (8) by μ_{ij} to get the optimality as (9) and Differentiate (8) again by v_j to find the optimality for the update of centroid, we have the solution as (10)

$$\mu_{ij} = \frac{1 + k \log_m \left[\frac{1}{d(x_i, v_j)^2} \right] - \sum_{u=1}^k \log_m \left[\frac{1}{d(x_i, v_u)^2} \right]}{k} = \frac{1 + \log_m \left[\frac{\prod_{u=1}^k d(x_i, v_u)^2}{(d(x_i, v_j)^2)^k} \right]}{k} \quad (9)$$

$$v_{ij} = \frac{\sum_{i=1}^N [(m^{\mu_{ij}} - 1)x_{ij}]}{\sum_{i=1}^N m^{\mu_{ij}} - 1} \quad (10)$$

3.1 Noise Filtering

The degree of membership calculated by (9) could go beyond 1 on the positive side or shift below 0 to the negative number side. The logarithm term in (9) could be negative if and only if

$$(d(x_i, v_j)^2)^k > \prod_{u=1}^k d(x_i, v_u)^2 \quad (11)$$

Especially, the data locates very far away from the cluster. The out of range to negative side for the membership degree emphasizes that data x_i is strongly should not be included as a member of the cluster j or it can be considered as noise. In the other hand, if the membership degree is beyond 1, it emphasizes that data x_i is strongly be the member of the cluster j . So the algorithm can filter the noise in case the degree of membership is negative number. However the degree of membership that produce from (9) by including out of range cases is usually not complement to Fuzzy approach in which $\mu_{ij} \in [0,1]$. Thus, a new condition (12) is introduced to verify the inclusive of data. In case the conflict with condition (12), data is regardless and the degree of membership will be assigned to zero in target cluster.

$$m \geq \frac{(d(x_i, v_j)^2)^k}{\prod_{j=1}^k d(x_i, v_j)^2} \quad (12)$$

In order to embed the noise filtering into algorithm, the clustering process for XFCM can be modified and illustrated in Fig. 2.

As aforementioned, the data sparsity is a fundamental problem for Collaborative Filtering. To fill the missing value during the initialize centroid, we use average rating of that item rather than average rating of that user because it return better result when we combine Clustering with Item based method.

STEP1 Predefined parameters: This step is to define the required parameters to process in the algorithm. They are; predefine the number of cluster (k), the fuzzifier parameter (m) and termination coefficient (ε) and initialize k centroids for each cluster j .

STEP2 Allocation the data: Compute membership degree of all data. If data does not satisfy condition (12), assign degree of membership to zero. Repeat this step until all data satisfy condition (12)

STEP3 Update Centroids: Once all data are allocated, the centroids get updated regarding to their membership.

STEP4 Validate the Termination Criteria: Repeat STEP2 until the desire condition is met then algorithm is terminated.

Fig. 2. Procedure of Exponential Fuzzy Clustering

3.2 Item Based Exponential Fuzzy Clustering

We combined Item based Collaborative Filtering with XFCM by a replacement of dissimilarity in XFCM with inverse of similarity calculation in Item based method as describe in [2]. With this method, the estimate rating in (13) performs calculation against centroid and time complexity would reduce from $O(N)$ to $O(k)$ where k is number of clusters.

$$P_{u,i} = \sum_{j=1}^k \mu_{ij} \cdot v_j \quad (13)$$

4 Experiments

We use the real dataset from the MovieLens website¹ to validate the performance of XFCM. The data is very sparse with 0.9396 sparsity level. Such level measures by 1-(nonzero entries/total entries). We separated 80% of dataset for training and 20% for the prediction test. We evaluated accuracy by measure Mean Absolute Error (MAE).

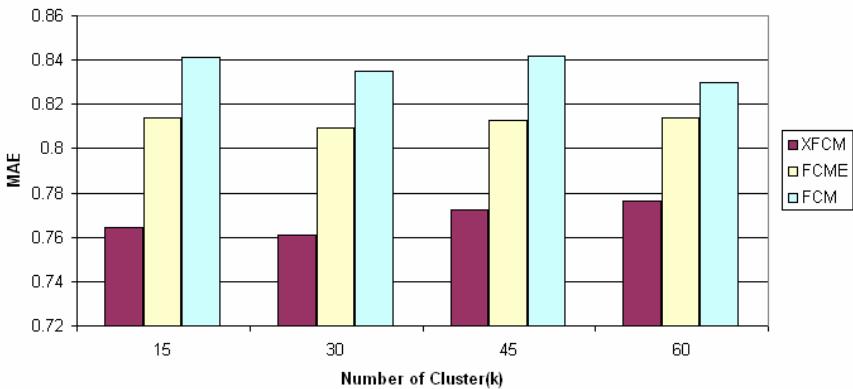


Fig. 3. Benchmarking Result. XFCM outperform FCM and FCME for MovieLens dataset. The improvement is 6.43-9.14% over FCM and 4.57-6.12% over FCME.

¹ <http://www.grouplens.org>

We estimated fuzzifier (m) by clustering data at different value of m . We found that the best result is at $m=10$. From our preliminary test, the changes of objective functions between next two iterations did not return the highest accuracy even with the very low ε . However the best accuracy could be selected from the first 10 iterations. We obtained the best result by benchmarking with FCM and FCME at variation of number of cluster k as illustration in Fig. 3.

We investigated the distribution of μ_{ij} from the best results generated by each clustering algorithms as illustration in Table 1. S.D. of μ_{ij} is in similar range while a large number of $\mu_{ij}=0$ is generated by XFCM. This means XFCM used much lower number of data to update the centroid (around 50%). Thus the centroid is not overfitted with the noise. Eventually, the positions of centroids do not shift away.

Table 1. Average S.D. of Membership Degree

	FCM	FCME	XFCM
S.D.	0.1029	0.1150	0.0911
Count No. of $\mu_{ij}=0$	0.02%	1.50%	50.98%

5 Conclusion

We addressed sensitivity of Fuzzy Clustering Centroid especially in Collaborative Filtering. We proposed a new novel algorithm to robust the centroid from the noise during the clustering process by enhancement the degree of membership. Our experiment indicated that the XFCM work very well in the noisy environment by outperformed FCM about 9.14% and FCME about 6.12%. As future work, we plan to study theoretical of XCM and integrate the possibilistic approach in order to create a more concrete in robustness algorithm.

References

1. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of Predictive Algorithms for Collaborative Filtering. In: 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52 (1998)
2. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: 10th ACM International World Wide Web Conference, pp. 285–295 (2001)
3. Kim, T.H., Park, S.I., Yang, S.B.: Improving Prediction Quality in Collaborative Filtering based on Clustering. In: IEEE/WIC/ACM International Conference, pp. 704–710 (2008)
4. Treerattnapitak, K., Juruskulchai, C.: Items Based Fuzzy C-Mean Clustering for Collaborative Filtering. Information Tech. J. 10, 30–34 (2009)
5. Treerattnapitak, K., Juruskulchai, C.: Entropy based Fuzzy C-Mean for Item-based Collaborative Filtering. In: 9th International Symposium on Communication and Information Technology, pp. 881–886 (2009)
6. Jinlong, W., Tiejun, L.: A modified fuzzy C-means algorithm for collaborative filtering. In: 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition, pp. 1–4 (2008)

7. Miyamoto, S., Mukaidono, M.: Fuzzy c - means as a regularization and maximum entropy approach. In: 7th International Fuzzy Systems Association World Congress, vol. 2, pp. 86–92 (1997)
8. Miyamoto, S., Suizu, D., Takata, O.: Methods of fuzzy c-means and possibilistic clustering using quadratic term. *Scientiae Mathematicae Japonicae*. l. 60(2), 217–233 (2004)
9. Hodge, V.J., Austin, J.: A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, 85–126 (2004)
10. Knorr, E., Ng, R.: Algorithms for mining distancebased outliers in large datasets. In: 24th Conference on VLDB, New York, pp. 392–403 (1998)
11. Ramaswamy, S., Rastogi, R., Kyuseok, S.: Efficient algorithms for mining outliers from large data sets. In: ACM SIGMOD International Conference on Management of Data, pp. 93–104 (2000)
12. Breunig, M.M., Kriegel, H.P., Ng, R.T.: LOF: Identifying density based local outliers. In: ACM Conference, pp. 93–104 (2000)
13. Escalante, H.J.: A Comparison of Outlier Detection Algorithms for Machine Learning. In: Programming and Computer Software, pp. 228–237 (2005)
14. Jiang, M.F., Tseng, S.S., Su, C.M.: Two-phase clustering process for outliers detection. *Pattern Recognition Letters* 22(6-7), 691–700 (2001)
15. He, A., Xu, X., Deng, S.: Discovering Cluster Based Local Outliers. *Pattern Recognition Letters* 24(9-10), 1641–1650 (2003)
16. Pal, N.R., Pal, K., Bezdek, J.C.: A Mixed C-Means Clustering Model. In: IEEE International Conference on Fuzzy Systems, pp. 11–21 (1997)
17. Pal, N.R., Pal, K., Keller, J.M., Bezdek, J.C.: A Possibilistic Fuzzy c-Means Clustering Algorithm. *IEEE Transactions On Fuzzy Systems* 13(4), 517–530 (1997)
18. Krishnapuram, R., Keller, J.: A possibilistic approach to clustering. *IEEE Trans. Fuzzy Syst.* 1(2), 98–110 (1993)
19. Mizutani, K., Miyamoto, S.: Possibilistic Approach to Kernel-Based Fuzzy c-Means Clustering with Entropy Regularization. In: Torra, V., Narukawa, Y., Miyamoto, S. (eds.) MDAI 2005. LNCS (LNAI), vol. 3558, pp. 144–155. Springer, Heidelberg (2005)
20. Miyamoto, S., Ichihashi, H., Honda, K.: Algorithms for Fuzzy Clustering. Springer, Heidelberg (2008)

Real-Valued Multimodal Fitness Landscape Characterization for Evolution

P. Caamaño, A. Prieto, J.A. Becerra, F. Bellas, and R.J. Duro

Integrated Group of Engineering Research,
Escuela Politécnica Superior, Campus de Esteiro, 15403 Ferrol, Spain
`{pcosbrino, aprieto, ronin, fran, richard}@udc.es`
<http://www.gii.udc.es>

Abstract. This paper deals with the characterization of the fitness landscape of multimodal functions and how it can be used to choose the most appropriate evolutionary algorithm for a given problem. An algorithm that obtains a general description of real valued multimodal fitness landscapes in terms of the relative number of optima, their sparseness, the size of their attraction basins and the evolution of this size when moving away from the global optimum is presented and used to characterize a set of well-known multimodal benchmark functions. To illustrate the relevance of the information obtained and its relationship to the performance of evolutionary algorithms over different fitness landscapes, two evolutionary algorithms, Differential Evolution and Covariance Matrix Adaptation, are compared over the same benchmark set showing their behavior depending on the multimodal features of each landscape.

1 Introduction

The application of evolutionary algorithms (EA) to real-valued problems is one of the most prolific topics in evolutionary computation due to the successful results obtained, mainly in high dimensional cases [1][2], where other approaches fail. The main drawback detected by the users is the uncertainty about which would be the most appropriate EA for a particular problem or fitness landscape. This usually implies either a trial and error stage until a proper algorithm is found or the use of suboptimal algorithms which may waste precious resources and time. Due to the usually high computational cost of EAs, much attention has been paid to solving this problem through the study and analysis of the features that characterize a landscape to determine its complexity prior to EA selection [3][4].

The multimodality of a fitness landscape is one of the main features that contribute to the increase in the difficulty of an optimization problem. It seems clear that a large number of local optima is a characteristic that complicates the search, even more as dimensionality grows. However, this feature is not so relevant on its own, and must be combined with the sizes of the attraction basins of the local optima or the distances between these local optima and the global optimum for the analysis to be meaningful. Here, we propose a real-valued fitness landscape analysis algorithm that provides such detailed information and

discuss its implications using two very successful EAs, Differential Evolution and Covariance Matrix Adaptation.

The rest of the paper is structured as follows: section 2 is devoted to the formal presentation of the algorithm and its application to the characterization of a multimodal benchmark function set. Section 3 presents the results obtained when applying the algorithm to the problem of selecting between Differential Evolution (DE) and Covariance Matrix Adaptation (CMA), showing how this detailed information is crucial in some cases. Finally, the main conclusions of this work are commented in Section 4.

2 Multimodal Fitness Landscape Characterization

In [5], the authors present a method to analyze multimodal fitness landscapes based on the attraction basins theory for binary valued functions. They suggest an algorithm that, given a landscape and a local search operator selected for the particular features of the EA under study, estimates the number of local optima and the attraction basins size with the objective of determining the complexity of such a landscape for the particular operator. Starting from the background ideas of this work, we have developed a more general analysis algorithm for real valued problems that uses a generic local search operator to estimate the local optima distribution with the objective of comparing different EAs in these terms.

First of all and for the sake of formalization, a fitness landscape is defined as a composition of three elements [6]:

- A set S of possible solutions, also known as the search space of the problem.
- A neighborhood function $\mathcal{V} : S \longrightarrow 2^{|S|}$, which assigns to each solution $s \in S$ a subset of neighbors $\mathcal{V}(s) \subset S$. The concept of neighborhood is defined from a local search operator, i. e., if we have an operator μ , the neighborhood of a solution s is the set $\mathcal{V}(s) = y \in S | y = \mu(s)$.
- A fitness function $f : S \longrightarrow \mathbb{R}$, that assigns the fitness value to each solution.

The degree of difficulty of a fitness landscape is linked, among other features, to the existence of local optima. Thus:

- A fitness landscape is *multimodal* when it has more than one local optimum.
- A solution $m \in S$ is a local optimum if, for every solution of its neighborhood, there is no solution with a better fitness value.
- Each local optimum of the fitness landscape is associated with an attraction basin $E(m)$.
- A solution s_0 belongs to the attraction basin of the local optimum m if, by applying operator μ a finite number of steps, the search process starting at s_0 ends up at m .

The distribution of local optima and the size of every attraction basin are important factors in the performance of any metaheuristics [7] and, therefore, of evolutionary algorithms.

The proposed algorithm applies a local search to several random points $x_i \in S$, so that by starting at each x_i the algorithm ends up near one of the local optima. Therefore, to apply the algorithm, we assume that a landscape L, defined by a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with n parameters, may be divided into M partitions or subspaces, each one of them is an attraction basin for one of the local optima of the landscape. The algorithm is described in the following pseudocode:

Multimodality Analysis Algorithm

```

for all  $x_i \in N$  do
    do
        Record, for all  $\mu$ -neighbors of  $x_i$ ,  $f(\mu_j(x_i))$ .
        Assign  $x_i = \mu_j(x_i)$  where  $f(\mu_j(x_i))$  reach the minimum value.
    while there is strictly positive improvement in the
         $\mu$ -neighbors
end for
Where:
    -  $\mu(x_i)$  is the application of the  $\mu$  operator on  $x_i$ .
    - The  $\mu$  operator is applied P times in each loop for each  $x_i$ .
```

As shown, it is run for N random points of the search space. The μ operator generates each random point within an area of radius r around point x_i . After the execution of the local search, we store the number of random points belonging to each attraction basin. As we are working in a continuous domain, we have considered that a solution s_i belongs to an attraction basin if the Euclidean distance between s_i and the local optimum of this basin is less than δ . The point with the best fitness value in each attraction basin is taken as its center and corresponds to a local optimum. Finally, the best value of all the local optima is taken as the global optimum.

The number of random points (N) could be set to a fixed number but we have implemented it dependent on the search process. That is, we start with only 1 point and add one more each step until a stopping criterion is met. In particular, we stop the search process when the statistical frequencies of all the local optima do not change more than a threshold, $1.00E-3$ in this case, from one execution to the next.

The information provided by the algorithm has been summarized into the following measures:

- Number of attraction basins: an estimation of the number of attraction basins, and thus, of local optima, is produced.
- Size of the attraction basins: it can be calculated as the statistical frequency with which the local search algorithm reaches its corresponding local optimum.
- Distance between attraction basins: with this measure we can estimate the distribution of the local optima. In particular, in our experiments we have measured:
 - Maximum distance between attraction basins.
 - Distance between the global optimum and the local optimum with the largest attraction basin.

3 Application

In this section we apply the previous algorithm to a real valued multimodal function benchmark set to characterize it using the measures stated above and then run two very relevant EAs over the benchmark set trying to understand the influence of these parameters on the performance of the algorithms. The benchmark set is made up of 23 multimodal functions with different topological features (see Table I). Functions $f_1 - f_9$ [8] are scalable (in this work the results are obtained for dimension 5) and functions $f_{10} - f_{23}$ [9] are non-scalable with low dimensions (from 2 to 6).

The results obtained by applying the characterization algorithm to the selected benchmark function set are presented in Table II. The columns of the table display, from left to right, the measures commented above: an estimation of the number of local minima (m), the size of the largest basin and the optimum attraction basin (largest and optimum respectively), the distance between the largest and the optimum attraction basins and the maximum distance between attraction basins for the functions that make up the selected benchmark set.

Table 1. Analysis of the set of multimodal benchmark functions

Function	m	Attr. Basin Size		Distance	
		Largest	Optimum	lar.-opt.	Maxima
f₁ Cosine	$\approx 5^n$	$2.60E-2$	$2.60E-2$	$0.00E+0$	$2.15E-1$
f₂ Rastrigin	$\approx 11^n$	$8.00E-3$	$8.00E-3$	$0.00E+0$	$4.86E-1$
f₃ Schwefel	$\approx 8^n$	$2.36E-1$	$2.96E-5$	$9.20E-1$	$9.20E-1$
f₄ Ackleys	$\approx 65^n$	$1.00E-3$	$1.00E-3$	$0.00E+0$	$5.00E-1$
f₅ Griewank	$\approx 4^n$	$1.00E-3$	$2.00E-4$	$3.97E-1$	$4.55E-1$
f₆ Levy	$\approx 5^n$	$5.77E-1$	$5.77E-1$	$0.00E+0$	$2.61E-1$
f₇ Penalized1	$\approx 5^n$	$1.28E-1$	$1.28E-1$	$0.00E+0$	$1.13E-1$
f₈ Penalized2	$\approx 5^n$	$1.74E-1$	$1.74E-1$	$0.00E+0$	$5.00E-2$
f₉ Rosenbrock	2	$3.76E-1$	$0.376E-1$	$0.00E+0$	$2.20E-1$
f₁₀ Deckers	3	$8.79E-1$	$8.79E-1$	$0.00E+0$	$2.91E-1$
f₁₁ Bohac1	13	$7.54E-1$	$7.54E-1$	$0.00E+0$	$9.00E-3$
f₁₂ Bohac2	10	$5.35E-1$	$5.35E-1$	$0.00E+0$	$9.00E-3$
f₁₃ Foxholes	25	$6.70E-2$	$6.70E-2$	$0.00E+0$	$4.87E-1$
f₁₄ Beale	4	$4.69E-1$	$4.69E-1$	$0.00E+0$	$5.91E-1$
f₁₅ Aluffi	10	$8.60E-1$	$8.60E-1$	$0.00E+0$	$7.00E-2$
f₁₆ Becker	4	$1.00E+0$	$1.00E+0$	$0.00E+0$	$0.00E+0$
f₁₇ Goldstein	6	$5.03E-1$	$5.03E-1$	$0.00E+0$	$4.24E-1$
f₁₈ Hartman3	3	$6.22E-1$	$6.22E-1$	$0.00E+0$	$4.66E-1$
f₁₉ Hartman6	2	$6.25E-1$	$2.74E-1$	$4.50E-1$	$4.50E-1$
f₂₀ Shekel5	5	$3.12E-1$	$2.61E-1$	$2.02E-1$	$3.89E-1$
f₂₁ Shekel7	7	$2.95E-1$	$2.95E-1$	$0.00E+0$	$3.77E-1$
f₂₂ Shekel10	10	$2.40E-1$	$2.40E-1$	$0.00E+0$	$3.81E-1$
f₂₃ SixHump	6	$2.65E-1$	$2.32E-1$	$1.14E-1$	$1.40E-1$

Table 2. Success Performance (SP) and Success Rate (SR) for multimodal functions

Function	Dim.	CMA		DE	
		SR	SP	SR	SP
f_1 Cosine	5	1.00	2636.60	1.00	2232.84
f_2 Rastrigin	5	0.08	621949.00	1.00	5142.76
f_3 Schwefel	5	0.08	608619.00	1.00	7483.64
f_4 Ackleys	5	1.00	1864.60	1.00	5171.72
f_5 Griewank	5	0.04	1232999.00	1.00	19992.92
f_6 Levy	5	1.00	815.00	1.00	2210.80
f_7 Penalized1	5	1.00	1484.20	1.00	2759.96
f_8 Penalized2	5	1.00	2074.60	1.00	2869.88
f_9 Rosenbrock	5	1.00	3683.80	1.00	16258.24
f_{10} Deckers	2	1.00	282.60	1.00	370.92
f_{11} Bohac1	2	1.00	711.00	1.00	1252.64
f_{12} Bohac2	2	1.00	704.60	1.00	1247.80
f_{13} Foxholes	2	1.00	10612.20	1.00	2334.20
f_{14} Beale	2	1.00	753.40	1.00	939.44
f_{15} Aluffi	2	1.00	513.40	1.00	835.48
f_{16} Becker	2	1.00	523.80	1.00	526.88
f_{17} Goldstein	2	1.00	602.60	1.00	1050.36
f_{18} Hartman3	3	1.00	514.20	1.00	1798.92
f_{19} Hartman6	6	0.96	6034.00	0.28	156288.86
f_{20} Shekel5	4	0.88	24175.36	1.00	8261.76
f_{21} Shekel7	4	1.00	5315.00	1.00	5406.04
f_{22} Shekel10	4	1.00	5375.00	1.00	6885.12
f_{23} SixHump	2	1.00	317.40	1.00	1898.46

To illustrate the practical relevance of the characterization algorithm, we have run two well-known evolutionary algorithms, Differential Evolution (DE) [10] and Covariance Matrix Adaptation (CMA) [11], over the same benchmark function set. These two algorithms operate quite differently, the DE is a better explorer and the CMA a really good exploiter, although its exploration capabilities are limited. We have implemented the canonical version of the DE algorithm, *DE/rand/1/bin* with the parameters recommended in [12] and the CMA version and parameters as presented in [13]. For each benchmark function, 25 independent runs were performed. The stopping criterion is based on the number of function evaluations as in [2], with a maximum number set to $10000n$, being n the dimensionality of the problem. The results obtained in terms of Success Performance (SP) and Success Rate (SR) [2] are displayed in Table 2. The SP indicates the average number of function evaluations the algorithm needs to reach the optima and is an indication of the convergence speed. The SR is the number of success runs measured as a fraction of unity, it is 1.00 if all the runs are solved.

The table clearly shows that both algorithms obtain very successful results over the benchmark set. The CMA solves all the runs in 18 of the 23 functions while the DE solves all the runs for 22 of the 23. This is basically due to the fact

that we are using low dimensions in the functions and these two algorithms are very powerful. Consequently, in order to determine how the features of the fitness landscape produced by the algorithm can be associated to the performance of each one of the EAs, the analysis must be carried out in terms of the functions where they fail and in terms of their relative convergence speed.

A closer look at the results of Table 1 shows that there are 18 functions where the attraction basin of the optimum is the largest one (simply by comparing the values in columns *largest* and *optimum*) making the exploration phase easy, that is, it is easy to have samples in a large attraction basin, and if the largest one corresponds to the optimum and is large in terms of percentage of the search space, the exploration power of the algorithms is not determinant, being the discriminant element their exploitation capabilities. This is what is extracted from Table 2, which shows that the DE outperforms the CMA only for the Cosine, Rastrigin and Foxholes functions, where the optimum attraction basin is very small, in particular, less than 10% of the whole landscape.

When the size of the optimum attraction basin is larger than 10% of the landscape (functions Penalized1, Penalized2, Levy, Rosenbrock, Deckers, Bohal1, Boha2, Beale, Aluffi, Becker, Goldstein, Hartman3, Shekel7 and Shekel10), the behavior of the CMA improves as the exploration phase becomes less significant. Once the CMA locates the optimum attraction basin, its exploitation behavior is faster than that of the DE, obtaining better convergence speed results.

The remaining functions (Schwefel, Griewank, Hartman6, Shekel5 and Six-Hump) share the feature that the optimum attraction basin is not the largest one and there are several attraction basins in the landscape that take up more space. The Hartman6 function must be analyzed in detail, as it is the only case where the DE clearly fails in some runs. This function presents two attraction basins and the size of the optimum one is less than half the size of the other one. The convergence speed of the CMA is 25 times faster than the DE. In this case, the optimum attraction basin is large enough to be easily reached by the CMA and the DE. The CMA quickly exploits it and reaches the optimum. However, for the DE, this function is quite ambiguous. Its exploitation characteristics are not fast and precise enough not to be misled in some of the runs towards the local optimum with largest attraction basin confirming the trend indicated above: when the attraction basin of the optimum is large enough for basically any exploration strategy to place samples there, the better exploiter is the algorithm of choice, in this case, the CMA algorithm.

With the aim of performing a better analysis of the remaining two parameters, that is, variation of size of the attraction basins with distance from the optimum and distance between attraction basins, we consider the remaining four functions. Figure 1 displays their attraction basin distribution. The x-coordinate of these graphs represent the distance between the attraction basins and the optimum attraction basin whereas the y-coordinate represents the size of the attraction basins in a logarithmic scale (measured as the number of points of the search space that belong to each attraction basin given as a fraction of unity). In the first two graphs, corresponding to the Schwefel and Griewank functions, the size

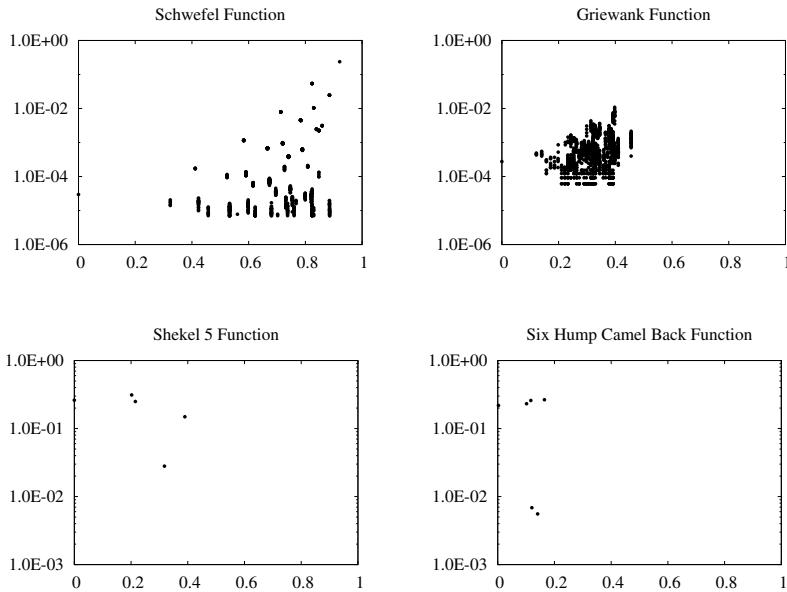


Fig. 1. Graphs of the distribution of the attraction basins for the scalable functions set. The X-axis represents the normalized distance to the optimum and the Y-axis represents the size of the attraction basins.

of the attraction basins increases when moving away from the optimum. This makes it more difficult to place samples in the optimum attraction basin and thus favors algorithms with better exploration characteristics. This is confirmed by the results obtained for the EAs, where the CMA was more than 70 times slower than the DE. The exploration strategy of the CMA tends towards large attraction basins, and in these cases, these attraction basins are far from the optimum.

The distance between attraction basins is also a relevant parameter as can be seen in the results for the Shekel5 and Sixhump functions which show in Fig. II a very similar size for all the attraction basins. However, in the Shekel5 function the distance between them is large, making it harder to "jump" from one to the next. This requires better exploration strategies (DE). On the other hand, the SixHump attraction basins are close, and even an underpowered exploration strategy allows jumping from one to the next, being again the exploitation of solutions the differentiating element in the performance of the algorithms, which favors the CMA due to its faster convergence.

4 Conclusions

In this paper we have presented an extension for real valued problems of the method proposed in [5] that allows us to characterize the topology of the fitness

landscapes for optimization problems in terms of attraction basins. By considering the number of optima, the size of their attraction basins and how their size evolves when moving away from the global optimum, we have been able to understand in detail the behavior of two very well known evolutionary algorithms, Differential Evolution and Covariance Matrix Adaptation, in multimodal terms. This analysis, through the consideration of the attraction basins of the optima, in particular their sparsity and size evolution, has helped to determine some relevant characteristics that make functions harder or easier to solve by each evolutionary algorithm and provides a way to estimate which would be more adequate for each type of function.

Acknowledgments. This work was partially funded by the Xunta de Galicia through the research project 09DPI012166PR.

References

1. Ashlock, D.: *Evol. Computation for Modeling and Optimization*, Springer, Heidelberg (2006)
2. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real parameter optimization, Nanyang Technological University, Tech. Rep (2005)
3. Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Eshelman, J. (ed.) *Proc. 6th Int. Conf. Genetic Algorithms*, pp. 184–192. Morgan Kaufmann, San Mateo (1995)
4. Manderick, B., de Weger, M., Spiessens, P.: The genetic algorithm and the structure of the fitness landscape. In: Belew, R.K., Booker, L.B. (eds.) *Proc. 4th Int. Conf. Genetic Algorithms*, pp. 143–150. Morgan Kaufmann, San Mateo (1991)
5. Garnier, J., Kallel, L.: How to detect all maxima of a function? In: *Proc. of the Second EVONET Summer School on Theoretical Aspects of Evolutionary Computing* (Anvers, 1999), pp. 343–370. Springer, Berlin (2001)
6. Stadler, P.F.: Towards a theory of landscapes. In: Lopez-Pena, R., Capovilla, R., Garcia-Pelayo, R., Waelbroeck, H., Zeturche, F. (eds.) *Complex Systems and Binary Networks*, vol. 461, pp. 77–163. Springer, Heidelberg (1995)
7. Garnier, J., Kallel, L.: Statistical distribution of the convergence time of evolutionary algorithms for long-path problems. *IEEE-EC* 4(1), 16 (2000)
8. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation* 3(2), 82–102 (1999)
9. Dixon, L.C.W., Szeg, G.P.: *The Global Optimization Problem: An Introduction. Towards Global Optimisation*. North-Holland, Amsterdam (1978)
10. Storn, R., Price, K.V.: Differential Evolution A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11, 341–359 (1997)
11. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 9, 159–195 (2001)
12. Ronkkonen, J., Kukkonen, S., Price, K.V.: Real-parameter optimization with differential evolution. In: *The 2005 IEEE Congress on, Evolutionary Computation*, vol. 1, 5, pp. 506–513 (2005)
13. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. In: *Congress on Evolutionary Computation 2005*, pp. 1769–1776 (2005)

Reranking for Stacking Ensemble Learning

Buzhou Tang, Qingcai Chen, Xuan Wang, and Xiaolong Wang

Harbin Institute of Technology Shenzhen Graduate School,
Computer Science Department, 518055 Shenzhen, China

{tangbuzhou, qingcai.chen}@gmail.com, {wangxuan, wangx1}@insun.hit.edu.cn

Abstract. Ensemble learning refers to the methods that combine multiple models to improve the performance. Ensemble methods, such as stacking, have been intensively studied, and can bring slight performance improvement. However, there is no guarantee that a stacking algorithm outperforms all base classifiers. In this paper, we propose a new stacking algorithm, where the predictive scores of each possible class label returned by the base classifiers are firstly collected by the meta-learner, and then all possible class labels are reranked according to the scores. This algorithm is able to find the best linear combination of the base classifiers on the training samples, which make sure it outperforms all base classifiers during training process. The experiments conducted on several public datasets show that the proposed algorithm outperforms the baseline algorithms and several state-of-the-art stacking algorithms.

Keywords: ensemble learning, stacking algorithm, meta-learner.

1 Introduction

Ensemble learning refers to the methods that combine multiple models to improve the performance. In other words, an ensemble is a technique for combining weak learners (also called as base learners) in order to form a strong learner. Therefore, an ensemble algorithm usually has a two-layer structure explicitly or implicitly. Many ensemble algorithms have been proposed. Among them, "stacking", which provides a meta-learner for combining base learners, is one of the most representative methods. The most important issues in stacking are the choice of the features and learning algorithm for the meta-learner.

The standard stacking algorithm is first introduced by Wolpert (1992), which takes the class labels predicted by the base classifiers as the input features of meta-learner [1]. Merz (1999) presents a stacking algorithm called SCANN that uses correspondence analysis to model the relationship between the training samples and their class labels predicted by the base classifiers [2]. It only selects the uncorrelated dimensions as the input features of meta-learner, and a nearest neighbor method is used as the meta-learner. Ting and Witten (1999) uses probability distributions of each possible class label returned by the base classifiers as input features of the meta-learner, and a multi-response linear regression (MLR) method is used as the meta-learner [3]. Todorovski and Džeroski (2000)

proposes a stacking algorithm called meta decision trees (MDTs) that replaces the output class labels in the leaves by the predictive probabilities returned by the base classifiers, and the meta-learner uses them as input features [4]. Seewald and Fürnkranz (2001) proposes a stacking algorithm called grading that built a meta-learner for each base classifier [5]. The task of each meta-learner is to determine whether the base classifier can be trusted. It uses both the features of the base classifier and the attribute indicating whether the class label predicted by the base classifier is correct. The final prediction of the ensemble system is determined by summing up the predictive probabilities of the base classifiers which are predicted to be correct. Based on the stacking with MLR, Seewald (2002) introduces an alternative stacking algorithm with MLR (called as StackingC (STC)), which reduces the number of probabilities returned by the base classifiers to overcome the weakness of stacking with MLR on multi-class problems [6]. Džeroski and Ženko (2004) proposes a stacking algorithm that uses multi-response model trees as the meta-learner [7]. Sill et al. (2009) presents a linear technique, Featured-Weighted Linear Stacking (FWLS), which incorporates meta-features with the predictive probabilities of the base classifiers for improvement on the Netflix Prize collaborative filtering dataset [8]. Sam Reid and Greg Grudic (2009) demonstrates the over-fitting problem of stacking algorithms and proposes the standard linear least squares regression regularized with an L_2 penalty (Ridge regression), an L_1 penalty (lasso regression) or a combination of the two (elastic net regression) to overcome the over-fitting [9]. Ledezma et al. (2010) uses genetic algorithms to search for good Stacking configurations automatically [10]. Although lots of stacking algorithms have been proposed to improve the performance on several problems, there is no guarantee that a stacking algorithm outperforms all base classifiers.

In this paper, we propose a new stacking algorithm, where the predictive scores of each possible class label returned by the base classifiers are firstly collected by the meta-learner, and then all possible class labels are reranked according to those scores. This algorithm is able to find the best linear combination of the base classifiers on the training samples, which make sure it outperforms all base classifiers during training process. Experiments conducted on several public datasets show that the proposed algorithm outperforms the baseline algorithms and several state-of-the-art stacking algorithms.

This paper is organized as follows. It starts with the introduction of reranking for stacking ensemble learning in Section 2. Section 3 then setups the experimental comparison of our algorithm with several state-of-the-art stacking algorithms. Section 4 presents and discusses the experimental results, and Section 5 draws a conclusion.

2 Reranking for Stacking Ensemble Learning

2.1 Framework of Stacking Algorithms

Let X and Y denotes the space of inputs and the set of class labels, respectively, assuming $Y = \{l_1, l_2, \dots, l_L\}$, where L is the number of class labels. Given a

training data set $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, where $x_i \in X$ and $y_i \in Y (i = 1, \dots, m)$. Suppose that there are K base machine learning classifiers: b_1, b_2, \dots, b_K with $b_i : X \rightarrow Y$, where $\forall i, b_i(x) = \arg \max_{l \in Y} s_{i,l}(x)$, $s_{i,l}(x)$

is the predictive score returned by the classifier b_i when the input x is labeled with l . The score $s_{i,l}(x)$ is usually the conditional probability $p_i(l|x)$ in statistical machine learning methods. Standard stacking algorithm seeks a meta-classifier $M_l : X'_l \rightarrow Y$, where $X'_l = \{(b_1(x), b_2(x), \dots, b_K(x))\}$, which takes the class labels predicted by the base classifiers as the input features. An alternative way, which takes the predictive scores returned by the base classifiers for each possible class as the input features, shows slightly better performance. It tries to find a meta-learner $M_s : X'_s \rightarrow Y$, where $X'_s = \{s_{11}(x), s_{12}(x), \dots, s_{1L}(x), \dots, s_{i1}, \dots, s_{ij}(x), \dots, (x)s_{KL}(x)\}$, $s_{ij}(x)$ is the predictive scores returned by the $i-th$ base classifier for the class label j for $1 \leq i \leq K$ and $1 \leq j \leq L$. We call the M_l "label-based" meta-learner and M_s the "score-based" meta-learner. The stacking ensemble algorithms using M_l are called as label-based stacking, while those using M_s are called as score-based stacking. The structures of the two basic stacking algorithms are shown in Fig. II. As shown in Fig. II, both stacking algorithms construct a meta-learner according to the output of base classifiers directly.

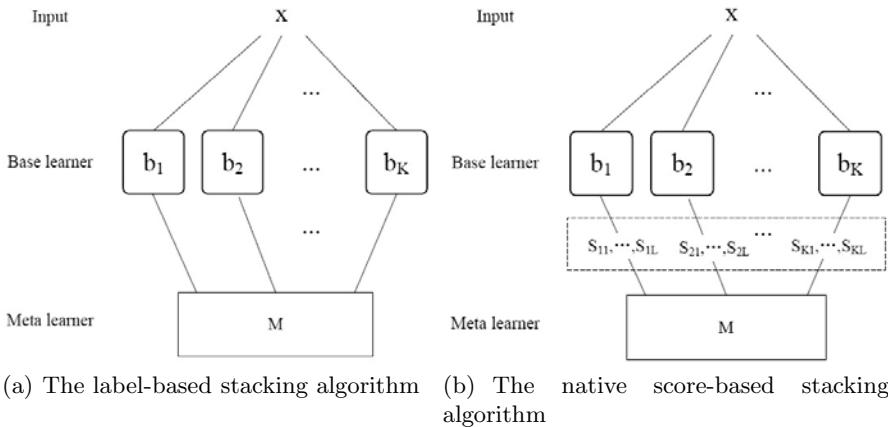


Fig. 1. Two types of basic stacking algorithms

2.2 Reranking for Stacking Ensemble Learning

In this paper, we propose a new stacking algorithm that builds a meta-learner to find a linear optimum combination of base classifiers on the training samples during training process. The meta-learner firstly collects the predictive scores returned by the base classifiers for each possible class label, and then reranks all possible class labels according to the scores. The key problem of this meta-learner is how to construct the input samples. It can be formulated as

follow: given an original sample (x, y) , get a predictive score vector of length $L: (s_{1,l}(x), s_{2,l}(x), \dots, s_{K,l}(x))$ for each possible class label l , construct ranked queries ($y > l_j$) for all $l_j \neq y$ if the original sample is a training sample. For example, there are 5 base classifiers and 3 possible class labels. Given an original training sample (x, y) (suppose that $y = l_1$), and the predictive scores returned by all base classifiers for each label are shown as Table 1.

Table 1. Example on predictive scores returned by $N = 5$ base classifiers

	b_1	b_2	b_3	b_4	b_5
l_1	0.8	0.2	0.2	0.1	0.5
l_2	0.1	0.7	0.2	0.4	0.3
l_3	0.1	0.1	0.6	0.5	0.2

We can get the following ranked queries from the given sample: ($l_1 > l_2$) and ($l_1 > l_3$), which can be formulated as:

$$\begin{array}{ccccccccc} 1 & 1 : 0.8 & 2 : 0.2 & 3 : 0.2 & 4 : 0.1 & 5 : 0.5 & \#l_1 \\ 0 & 1 : 0.1 & 2 : 0.7 & 3 : 0.2 & 4 : 0.4 & 5 : 0.3 & \#l_2 \\ 0 & 1 : 0.1 & 2 : 0.1 & 3 : 0.6 & 4 : 0.5 & 5 : 0.2 & \#l_3, \end{array}$$

where the i -th feature of each row denotes the output of the i -th base classifier, and its value denotes the predictive score returned by that classifier for the given class label followed by "#". The description of this algorithm is shown in Fig. 2.

The goal of our meta-learner is to find a parameter vector that makes every original training sample (x, y) satisfy the following constraints:

$$\forall l_j \neq y, w^T(s_{1,y}(x), s_{2,y}(x), \dots, s_{K,y}(x)) > w^T(s_{1,l_j}(x), s_{2,l_j}(x), \dots, s_{K,l_j}(x))$$

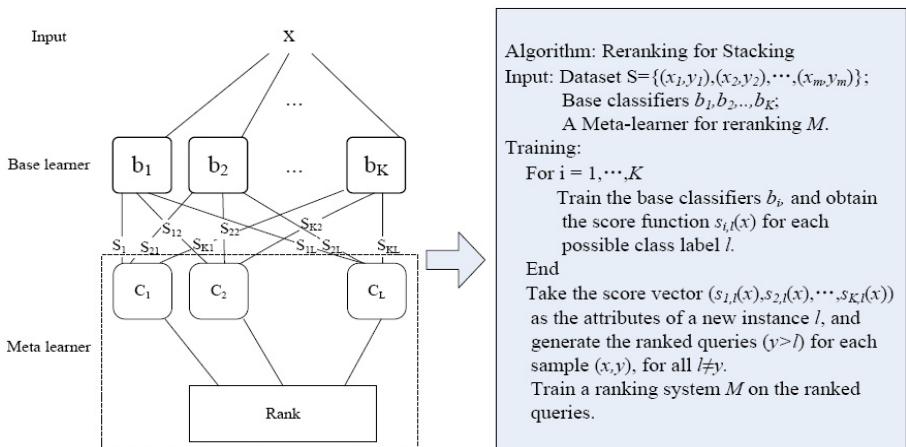


Fig. 2. Reranking for stacking ensemble learning

It is equivalent to make:

$$y = \arg \max_{l \in Y} w^T (s_{1,l}(x), s_{2,l}(x), \dots, s_{K,l}(x))$$

Actually, it is a process of finding a linear optimum combining vector w of the base classifiers on the training samples. Since each base classifier is a special combining scheme, where all items of the combining vector are set to zero except the one corresponding to the base classifier, the meta-classifier with the linear optimum combining vector w must be better than the base classifier with a special linear combining vector. If there is a linear optimum combining vector, a ranking algorithm can be used as the meta-learner to find it automatically.

Table 2. 2-best predictive scores of the base classifiers

	b_1	b_2	b_3	b_4	b_5
l_1	0.8	0.2	0.2	NONE	0.5
l_2	0.1	0.7	NONE	0.4	0.3
l_3	NONE	NONE	0.6	0.5	NONE

Selecting N-best predictive scores returned by the base classifiers for all possible class labels is an alternative way for generating ranked samples. This action may make disagreement, but we can adopt a simple way to solve it. In the case of the example shown in Table 1, if we set $N = 2$, only a part of scores shown in Table 2 can be included in the meta-learner. As shown in Table 2, the smallest score of each base classifier is removed when combining, and its corresponding label is not consistent with each others. In this case, we set all the missing scores to 0.

3 Experimental Setup

In the experiments, we investigate two issues: 1) the performance of reranking for stacking compared with other existing state-of-the-art stacking methods, such as standard stacking or native score-based stacking, et al. 2) the effect of N-best predictive scores on our method. Since the number of base classifiers effects very slightly on stacking [7], we set it to five. The following popular algorithms that have been implemented in the Weka¹ data mining suite are used as the base classifiers:

- Logistic: the multinomial logistic regression learning algorithm [11],
- SVM: the support machine learning algorithm [12],
- J4.8: the decision tree learning algorithm C4.5 [13],
- NB: the naive bayes learning algorithm [14],
- MLP: the multilayer perceptron (backpropagation neural network) learning algorithm.

¹ <http://www.cs.waikato.ac.nz/~ml/weka/>

For the meta-learner, the SVMrank is used for reranking, which can train ranking SVM efficiently [15] (called as RSVM). Following six schemes for ensemble learning are used for comparison:

- BEST: The best base algorithm selected by hand. It is the baseline of our algorithm.
- VOTE: The plurality voting scheme, where each base classifier casts a vote for its prediction. It is a baseline for comparison of ensemble classifiers.
- LSVM: Label-based stacking with label-based Support Vector Machine (SVM) as described in Fig.1(a),
- SSVM: Score-based stacking with SVM as described in Fig.1(b),
- LJ4.8: Score-based stacking with J4.8,
- SJ4.8: Score-based stacking with J4.8.

In order to evaluate the performance of our method, we conducted all experiments on a collection of seventeen datasets from the website: <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>. These datasets have been widely used in related studies [6-11]. Among them, nine datasets only compose of one sample set, and the other eight datasets compose of a training set and a testing set. A summary of these datasets is listed in Table 3.

Table 3. Datasets and their properties.”#Attri” denotes the number of attributes in a given dataset

Dataset	#Samples	#Class	#Attri	Dataset	#Training	#Testing	#Class	#Attri
diabetes	768	2	8	adult	32561	16281	2	123
segment	2310	7	19	dna	2000	1186	3	180
wine	178	3	13	letter	15000	5000	26	16
australian	690	2	14	satimage	4435	2000	6	36
breast-cancer	683	4	10	shuttle	43500	14500	7	9
heart	270	2	13	splice	1000	2175	2	60
iris	150	3	4	ijcnn1	49990	91701	2	22
glass	214	7	9	poker	25010	1000000	10	10
german	1000	2	24					

In our experiments, the predictive accuracy (Acc) is used to evaluate the performance of each stacking algorithm C , denoted as $Acc(C)$. For the datasets that only contain one sample set, the performance is estimated using 10-fold cross validation, and the same folds (randomly generated) are used in all experiments.

For pair-wise comparison of stacking algorithms, we calculate the relative improvement as described below. In order to evaluate the accuracy improvement achieved on a given dataset by using algorithm C_1 as compared to using algorithm C_2 , we calculate the improvement: $Acc(C_1)/Acc(C_2) - 1$.

4 Experimental Results

We firstly compare RSVM with other six ensemble algorithms. The predictive accuracies of them are given in Table 4. RSVM achieves the best accuracy on eleven datasets, BEST on six datasets, SSVM on two datasets, VOTE on one dataset, and the others do not achieve the best accuracy on all datasets. Compared with BEST, RSVM shows better performance on eleven datasets, equal performance on two datasets, and worse performance on four datasets. It does not always achieve better performance than BEST on all datasets, although RSVM can make sure outperform all the base classifiers on the training samples. The different distributions of the training samples and testing samples might lead to this result. Even so, RSVM shows better performance than BEST overall. Compared with VOTE, RSVM achieves better performance on all individual datasets. Compared with other four stacking algorithms, RSVM also shows better performance overall.

Table 4. The accuracies of the stacking algorithms

Dataset	BEST	VOTE	LSVM	SSVM	LJ4.8	SJ4.8	RSVM
diabetes	0.776043	0.773445	0.748718	0.743575	0.738226	0.740858	0.786483
segment	0.968831	0.963636	0.970563	0.972294	0.970996	0.968831	0.973160
wine	0.994444	0.988889	0.988889	0.994444	0.983333	0.959320	0.994444
australian	0.862428	0.855180	0.834910	0.837766	0.836318	0.836192	0.863877
breast-cancer	0.970736	0.970715	0.957565	0.953218	0.956095	0.956073	0.975127
heart	0.855556	0.840741	0.796296	0.800000	0.796296	0.781481	0.848148
iris	0.986666	0.960000	0.980000	0.973333	0.966667	0.966667	0.986666
glass	0.667665	0.677756	0.662009	0.702325	0.662489	0.659418	0.678528
german	0.774000	0.764000	0.713000	0.716000	0.712000	0.714000	0.757000
adult	0.944910	0.843191	0.841840	0.854493	0.841840	0.841226	0.975063
dna	0.936760	0.956155	0.883642	0.885329	0.883642	0.877740	0.956155
letter	0.877000	0.858400	0.873400	0.902600	0.876200	0.8818	0.886200
satimage	0.880000	0.875000	0.864000	0.868000	0.858500	0.8605	0.883000
shuttle	0.999520	0.990690	0.999448	0.999586	0.999517	0.999172	0.998345
splice	0.968350	0.892874	0.860690	0.859310	0.860690	0.854713	0.965977
ijcnn1	0.960210	0.922782	0.936108	0.936108	0.936140	0.947089	0.958779
poker	0.566620	0.501209	0.563061	0.562390	0.563055	0.556179	0.568143

In order to study the significant of the performance improvement of our method, we use the measure of the average relative improvement which is calculated by using the arithmetic mean of relative improvement described in section 3 on individual datasets. The result is shown in Table 5. Moreover, how many times does the algorithm C_1 better or worse than C_2 are also shown in Table 5.

Inspecting Table 5, we find that the four stacking algorithms (we did not consider RSVM at this stage of the analysis) shows almost the same performance, although the stacking algorithms with SVM (LSVM and SSVM) slightly outperform the stacking algorithms with J4.8 (LJ4.8 and SJ4.8). It is a little strange

Table 5. Average relative performance of ensemble algorithms(%). The entry in row X and column Y gives the average relative improvement of X over Y and the number of wins/loss.

	BEST	VOTE	LSVM	SSVM	LJ4.8	SJ4.8	RSVM
BEST		2.7 15/2	3.7 16/1	3.0 12/4	3.9 16/1	4.2 15/1	-0.4 4/11
VOTE	-2.5 2/15		1.0 10/6	0.4 8/9	1.3 11/6	1.6 11/6	-2.9 0/16
LSVM	-3.4 1/16	-0.9 6/10		-0.6 6/10	0.2 7/6	0.5 13/4	-3.8 1/16
SSVM	-2.7 4/12	-0.2 9/8	0.7 10/6		0.9 13/4	1.2 15/2	-3.2 2/14
LJ4.8	-3.6 1/16	-1.1 6/11	-0.2 6/7	-0.9 4/13		0.3 11/4	-4.0 1/16
SJ4.8	-3.9 1/15	-1.4 6/11	-0.5 4/13	-1.2 2/15	-0.3 4/11		-4.3 1/16
RSVM	0.4 11/4	3.2 16/0	4.1 16/1	3.5 14/2	4.4 16/1	4.7 16/1	

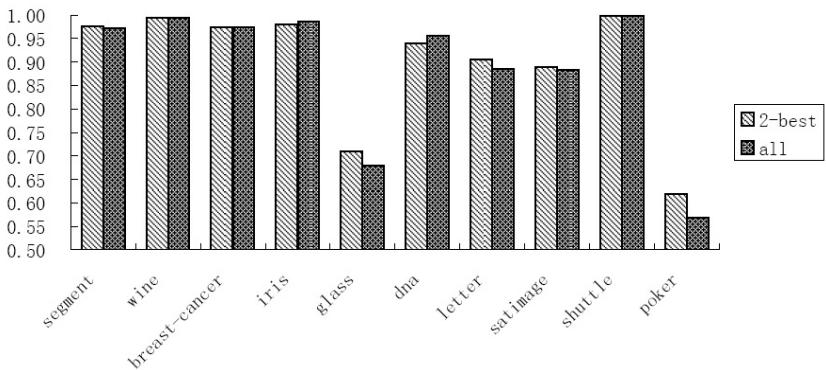


Fig. 3. The influence of N-best predictive scores on RSVM

that VOTE performs better than these four stacking algorithms. In this case, we analyze the results of the base classifiers, and find that the naive bayes learning algorithm is much inferior to the other base algorithms. The unbalance base algorithms may be the reason for this strange result. BEST clearly outperforms VOTE and the four stacking algorithms.

Compared to the other ensemble algorithms, RSVM shows much better performance, with a wins - loss difference of at least 7 and a relative improvement of at least 0.4%. The difference of 7 and relative improvement 0.4% are smallest when compared to BEST. When compared to VOTE, the difference and relative improvement achieve 15 and 3.2% respectively. When compared to other four stacking algorithms, the difference and relative improvement achieve 12 and 3.5% at least.

We next study the influence of N-best predictive scores on RSVM. As the choice of N is limited by the number of class label in a dataset, which is different from each other, it is difficult for us to compare all possible Ns on each dataset. For convenience, we only compare RSVM using 2-best predictive scores returned by the base classifiers with RSVM using all scores on the multi-class

datasets (The 2-best scores are the same as all scores for binary problems). Their predictive accuracies are shown in Fig.3. We find that RSVM using 2-best scores achieves similar performance on six datasets, better performance on three datasets and worse performance on one datasets. It indicates that a good choice of N can bring further improvement.

5 Conclusions and Future Work

We have proposed a new method for stacking, which builds a meta-learner to find a linear optimum combination of base classifiers on the training samples during training process. The meta-learner firstly collects the predictive scores returned by the base classifiers for each possible class label, and then reranks all possible class labels according to the scores. The experimental results have shown that this algorithm clearly outperforms the best base classifier, the voting algorithm, and several state-of-the-art stacking algorithms. Moreover, we have presented an alternative algorithm which uses N-best predictive scores returned by the base classifiers at the meta-level for further improvement. However, "how to select a good number of N-best?" is a issue of further work. In addition, the proposed algorithm is only applied to the classification problems in the present study. It is also suitable for more complex problems, such as structured prediction problems including sequence labeling problems. These problems are potential area for application of this proposed algorithm.

Acknowledgments. We wish to thank the authors of the toolkits used in this paper and the reviewers of this paper. This research has been partially supported by the National Natural Science Foundation of China (No.60435020 and No.90612005) and National 863 Program of China (No.2007AA01Z194).

References

1. Wolpert, D.H.: Stacked generalization. *Neural Networks* 5(2), 241–260 (1992)
2. Merz, C.J.: Using correspondence analysis to combine classifiers. *Machine Learnig* 36, 33–58 (1999)
3. Ting, K.M., Witten, I.H.: Issues in Stacked Generalization. *Journal of Artificial Intelligence Research* 10, 271–289
4. Todorovski, L., Džeroski, S.: Combining multiple models with meta decision trees. In: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, pp. 54–64 (2000)
5. Seewald, A., Fürnkranz, J.: An evaluation of grading classifiers. In: Advances in Intelligent Data Analysis. Fourth International Symposium on Intelligent Data Analysis (IDA), pp. 221–232. Springer, Berlin (2001)
6. Seewald, A.: How to make stacking better and faster while also taking care of an unknown weakness. In: Claude Sammut, A.G.H. (ed.) *Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002)*, Sidney, Australia, Morgan Kaufmann, San Francisco (July 2002)
7. Džeroski, S., Ženko, B.: Is combining classifier better than selecting the best one? *Machine Learning* 54(3), 255–273 (2004)

8. Sill, J., Takacs, G., Mackey, L., Lin, D.: Feature-Weighted Linear Stacking, arXiv:0911.0460 (2009)
9. Reid, S., Grudic, G.: Regularized Linear Models in Stacked Generalization. In: Benediktsson, J.A., Kittler, J., Roli, F. (eds.) MCS 2009. LNCS, vol. 5519, pp. 112–121. Springer, Heidelberg (2009)
10. Ledezma, A., Aler, R., Sanchis, A., Borrajo, D.: GA-stacking: Evolutionary stacked generalization. Intelligent Data Analysis 14(1), 89–119 (2010)
11. le Cessie, S., van Houwelingen, J.C.: Ridge Estimators in Logistic Regression. Applied Statistics 41(1), 191–201 (1992)
12. EL-Manzalawy, Y., Honavar, V.: WLSVM: Integrating LibSVM into Weka Environment (2005) Software, <http://www.cs.iastate.edu/~yasser/wlsvm>
13. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
14. John, G.H., Langley, P.: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, pp. 338–345 (1995)
15. Joachims, T.: Training Linear SVMs in Linear Time. In: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), pp. 217–226 (2006)

A Three-Strategy Based Differential Evolution Algorithm for Constrained Optimization

Saber M. Elsayed, Ruhul A. Sarker, and Daryl L. Essam

School of Engineering and Information Technology, University of New South Wales at
ADFA (UNSW@ADFA), Canberra 2600, Australia

saber.elsayed@student.adfa.edu.au, {r.sarker, d.essam}@adfa.edu.au

Abstract. Constrained Optimization is one of the most active research areas in the computer science, operation research and optimization fields. The Differential Evolution (DE) algorithm is widely used for solving continuous optimization problems. However, no single DE algorithm performs consistently over a range of Constrained Optimization Problems (COPs). In this research, we propose a Self-Adaptive Operator Mix Differential Evolution algorithm, indicated as SAOMDE, for solving a variety of COPs. SAOMDE utilizes the strengths of three well-known DE variants through an adaptive learning process. SAOMDE is tested by solving 13 test problems. The results showed that SAOMDE is not only superior to three single mutation based DE, but also better than the state-of-the-art algorithms.

Keywords: Constrained optimization, differential evolution.

1 Introduction

Solving constrained optimization problems (COPs) has been known as a challenging research area in the computer science, operations research, and optimization domains. COPs can be divided into many different categories based on their characteristics and mathematical properties.

Evolutionary algorithms (EAs) have a long history of successfully solving COPs. Differential evolution (DE) [14] is an efficient variant of EA for solving COPs. DE usually converges fast, incorporates a relatively simple and self-adapting mutation, and the same settings can be used for many different problems [14]. However, from the literature, there is no single DE algorithm which is able to solve a wide range of COPs consistently. One variant may be good for one test problem, and may be poor for another. Also, one variant may be efficient during the early stages of the search process, but may be inefficient later on.

In this research, we present a Self-Adaptive Operator Mix DE (SAOMDE) for solving COPs. This can be seen not only as a better alternative over the trial-and-error based algorithm design, but also as a provider of a better coverage of the problems. In the proposed algorithm, each DE variant has its own sub-population and the sub-population size varies adaptively, as the evolution progresses, depending on the reproductive success of the search operators, with a self-adaptive learning strategy for changing the subpopulation sizes as the evolution progresses. However, the sum of

the size of all of the sub-populations is fixed during the entire evolutionary process. The algorithm is tested by solving a well-known test set of 13 problems. The experimental results show that the proposed algorithm not only performs significantly better than the single mutation operator based DE, but also outperforms the state-of-the-art algorithms.

This paper is organized as follows. After the introduction, section 2 presents the DE algorithm with its parameters overview. Section 3 describes the design of the proposed algorithm. The experimental results and the analysis of those results are presented in section 4. Finally, the conclusions are given in section 5.

2 Differential Evolution

Differential Evolution (DE) is known as a powerful algorithm for continuous optimization. DE is a hybrid algorithm that takes the concept of a larger population from GA and of self-adapting mutation from ES [14]. DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector (mutated vector). Then crossover takes place in which the mutated vector's parameters are mixed with the parameters of another predetermined vector (the target vector) to yield a so-called child vector. If the child vector is better than the target vector, then the child vector replaces the target vector in the next generation. Each population vector has to serve once as the target vector, so that NP competitions take place in one generation [14].

In DE, there are many strategies for mutation, such as DE/rand/1 [15], DE/best/1 [15], DE/rand-to-best/2 [12], rand/2/dir [9], DE/current-to-rand/1 [5], DE/current-to-best/1 [11]. The DE family also uses one of two crossover schemes, exponential and binomial.

From an analysis of the literature, we know that there is no single DE variant that will be able to reach the best results for all types of problems. Mezura-Monte et al. [9] performed a comparative study of several DE variants in solving unconstrained global optimization problems. They found that variants "current-to-best/1" and "current-to-rand/1", with arithmetic recombination, have difficulty to explore the search space when solving multimodal functions. They suggested that a combination of arithmetic recombination with the DE mutation based on differences is not suitable for solving problems with high dimensionality. The alternatives "best/1/bin" and "rand/1/bin" are much better than "best/1/exp" and "rand/1/exp", due to the fact that in the exponential recombination, not all of the corners of the hypercube formed by the mutation vector and the current parent can be sampled, regardless of the " Cr " value used. The variant "best/1/bin" is the most competitive for unimodal with both separable and non-separable test functions. However, for multimodal functions, this variant provides competitive results only when the function is separable. For multimodal and non-separable functions, the variant "rand/2/dir" is more suitable, due to its ability to incorporate information regarding the fitness of the individuals in the mutation process.

Again, in DE, there is no fixed value for each parameter (the amplification factor of scaling factor F , the crossover rate Cr , and the population size P) that will be able to solve all types of problems with a reasonable quality of solution. Many studies have been conducted on parameter selection. Storn et al. [14] recommended a population size of 5–20 times the dimensionality of the problem, and that a good initial

choice of F could be 0.5. Gamperle et al. [4] evaluated different parameter settings of DE, where a plausible choice of the population size NP is between $3D$ and $8D$, with the scaling factor $F = 0.6$ and the crossover rate Cr in $[0.3, 0.9]$. Ronkkonen et al. [13] claimed that typically $0.4 < F < 0.95$ with $F = 0.9$ is a good first choice, and that Cr typically lies in $(0, 0.2)$ when the function is separable, while in $(0.9, 1)$ when the function's parameters are dependent. Qin et al. [12] proposed a novel Self-adaptive Differential Evolution algorithm (SaDE), where the choice of learning strategy and the two control parameters F and Cr are not required to be pre-specified. During evolution, the learning strategy and parameter settings are gradually self-adapted according to the learning experience. Abbass [1] proposed a self-adaptive version (crossover and mutation) for multiobjective optimization problems, where the scaling factor F is generated using a Gaussian distribution $N(0, 1)$.

The multi-operator DE has emerged in the last couple of years, and it has limited use so far. Mallipeddi et al. [7] proposed an ensemble of mutation strategies and control parameters with the DE (EPSDE). In EPSDE, a pool of distinct mutation strategies along with a pool of values for each control parameter coexists throughout the evolution process and competes to produce offspring. The algorithm has been used to solve a set of unconstrained problems. Tasgetiren et al. [16] proposed an ensemble DE, in such a way that each individual is assigned to one of the two distinct mutation strategies or a variable parameter search (VPS).

3 SAOMDE

In the evolution process, the relative performance of the search operators may vary with the progression of generations. This means that one operator may work well in the early stages of the search process and may perform poorly at the later stages, or vice-versa. So, it is inappropriate to give equal emphasis on all the operators throughout the entire evolution process. To give higher emphasis on the better performing operators, we propose to change the subpopulation sizes, through dynamic adaptation, based on the relative performances of the operators. SAOMDE starts with a random initial population which is divided into three subpopulations of equal size. Each subpopulation evolves through its own crossover and mutation operators, and the generated offspring are evaluated according to the fitness function value and/or the constraint violation of the problem under consideration. An improvement index for each subpopulation is calculated using the method as discussed below. Based on the improvement index, the subpopulation sizes are either increased or decreased or kept unchanged. As this process may abandon certain operators which may be useful at the later stages of the evolution process, we set a minimum subpopulation size for each operator. Also, after every few generations (indicated as window size WS) the best solutions among the subpopulations are exchanged. The algorithm continues until the stopping criterion is met. The basic steps of SAOMDE are presented in Table 1.

To measure the improvement of each operator (/subpopulation) in a given generation, we consider both the feasibility status and the fitness value, where the consideration of any improvement in feasibility is always better than any improvement in the infeasibility. For any generation $t > 1$, there arises one of three scenarios. These scenarios, in order from least desirable, to most desirable, are discussed below.

Table 1. Self-Adaptive Multi-operator Differential Evolution (SAOMDE)

STEP 1: In generation $t = 0$, generate an initial random population size PS . The variables in each individual must be within the range shown below:

$$x_{ijk} = x_{i,j,min} + u \times (x_{i,j,max} - x_{i,j,min}) \quad (1)$$

where $x_{i,j,min}, x_{i,j,max}$ are the lower and upper bound for decision variable x_{ijk} (i^{th} subpopulation, j^{th} individual and k^{th} variable) and u is a random number, $u \in [0,1]$.

STEP 2: Divide PS into three equal subpopulations p_i . Each subpopulation has its own operators and its own subpopulation of n_i individuals.

STEP 3: For each DE variant, generate the self-adaptive parameters F and Cr using equations (9) – (12), then generate the vector $O_i = \{O_i^1, O_i^2, \dots, O_i^j, \dots, O_i^n\}$ and update the fitness evaluations (FEs).

STEP 4: Sort the individuals in each subpopulation according to their objective function and/or constraint violation.

If $t \% WS = 0$; then

- 1- Replace the worst 2 individuals in each sub-population with other 2 best solutions (the best solution in each other subpopulation).
- 2- If there is any redundant decision vector, then replace it by generating a random vector.

Else, go to **STEP 5**.

STEP 5: Store the best individual for each operator, based on objective function and/or constraint violation.

STEP 6: If $t > 1$; then

Update n_i according formula (5).

STEP 7: Stop if the termination criterion is satisfied; else, go to **STEP 3**, and set $t = t + 1$.

A) Infeasible to infeasible: for any subpopulation i , the best solution was infeasible at iteration $t - 1$, and is still infeasible in iteration t , then the improvement index is calculated as follows.

$$VI_{i,t} = \frac{|V_{i,t}^{best} - V_{i,t-1}^{best}|}{avg.V_{i,t}} = I_{i,t} \quad (2)$$

where, $V_{i,t}^{best}$ = constraint violation of the best individual at iteration t , $avg.V_{i,t}$ = the average violation. Hence $VI_{i,t} = I_{i,t}$ above represents a relative improvement as compared to the average violation in the current iteration.

B) Feasible to feasible: for any subpopulation i , the best solution was feasible at iteration $t - 1$, and still feasible in iteration t , then the improvement index is:

$$I_{i,t} = \max_i(VI_{i,t}) + |F_{i,t}^{best} - F_{i,t-1}^{best}| \times FR_{i,t} \quad (3)$$

where $I_{i,t}$ = the improvement for subpopulation i at generation t , $FP_{i,t}$ = the objective function for the best individual at iteration t , the feasibility ratio of operator i at iteration t $FR_{i,t} = \frac{\text{Number of feasible solutions in a subpopulation } i}{\text{Subpopulation size at iteration } t}$.

To assign the higher index value to the subpopulation with a higher feasibility ratio, we multiply the improvement in fitness value by the feasibility ratio. To differentiate between the improvement index of feasible and infeasible subpopulations, we add a term $\max_i(VI_{i,t})$ in equation (3). If all the best solutions are feasible, then $\max_i(VI_{i,t})$ will be zero.

C) Infeasible to feasible: for any subpopulation i , the best solution was infeasible at iteration $t - 1$, and it is feasible in iteration t , then the improvement index is:

$$I_{i,t} = \max_i(VI_{i,t}) + |V_{i,t-1}^{best} + F_{i,t}^{best} - F_{i,t-1}^{bv}| \times FR_{i,t} \quad (4)$$

where $F_{i,t-1}^{bv}$ = fitness value of the least violated individual in iteration $t-1$.

To assign a higher index value to an individual that changes from infeasible to feasible, we add $V_{i,t-1}^{best}$ with the change of fitness value in equation (4). We also keep the first term as equation (3).

After calculating the improvement index for each subpopulation, the subpopulation sizes are calculated according to the following equation:

$$n_{i,t} = MSS + \frac{I_{i,t}}{\sum_{i=1}^m I_{i,t}} \times (PS - MSS \times Nopt) \quad (5)$$

where, $n_{i,t}$ is the sub-population size for the i^{th} operator at iteration t , MSS is the minimum subpopulation size for each operator i at iteration t , PS is the total number of population, and $Nopt$ is the number of operators.

In this paper, we measure the superiority of feasible points (during a tournament [3]) as follows: i) between two feasible solutions, the fittest one (according to the fitness function) is better, ii) a feasible solution is always better than an infeasible one, iii) between two infeasible solutions, the one having the smaller sum of its constraint violation is preferred. The equality constraints are transformed to inequalities of the form: $|h_j(\vec{x})| - \varepsilon \leq 0$, for $j = q + 1, \dots, m$, where ε is a small number.

4 Experimental Results and Analysis

From the literature, the DE variants “rand/ */ **” provide a better performance, because it finds the new search directions randomly [9]. Also, Ting et al. [17] investigated the benefits of using more than one difference vector (DV) in DE. The results showed that it is useful to use more than one DV (two or three DV are good enough, more than this may lead to premature convergence). In this research, we have selected the following three variants:

$$1\text{-Var1: rand/ 3/ bin: } \vec{V}_{i,t} = x_{r_1,t} + F \times ((x_{r_2,t} - x_{r_3,t}) + (x_{r_4,t} - x_{r_5,t}) + (x_{r_6,t} - x_{r_7,t})) \quad (6)$$

$$2\text{-Var2: best/ 3/ bin: } \vec{V}_{i,t} = x_{best,t} + F \times ((x_{r_1,t} - x_{r_2,t}) + (x_{r_3,t} - x_{r_4,t}) + (x_{r_5,t} - x_{r_6,t})) \quad (7)$$

$$3\text{-Var3: rand-to-best¤t/ 2/ bin: } \vec{V}_{i,t} = x_{r_1,t} + F \times ((x_{best,t} - x_{r_2,t}) + (x_{r_3,t} - x_{i,t})) \quad (8)$$

The 1st variant provides a random effect to investigate the search space, the variant with best value, as in the 2nd variant is popular and it is a good choice for both separable and non-separable unimodal test problems [9], while the 3rd variant showed encouraging results presented in [10].

One another advantage of our algorithm is the use of self-adaptive DE parameters F and Cr . So, for each decision vector (i), we generate two Gaussian numbers $N(0.5, 0.15)$ [1] one for F_i , while the other number represents Cr_i . Then, for the variants rand/3/bin and best/3/bin, the overall F and Cr are calculated according to the formulas (9) and (10), while the same parameters are calculated for the other two variants according to formulas (11) and (12).

$$\begin{aligned} F &= F_{r_1,G} + N(0, 0.5) \times (F_{r_2,G} - F_{r_3,G}) + N(0, 0.5) \times (F_{r_4,G} - F_{r_5,G}) + \\ &\quad N(0, 0.5) \times (F_{r_6,G} - F_{r_7,G}) \end{aligned} \quad (9)$$

$$Cr = Cr_{r_1,G} + N(0, 0.5) \times (Cr_{r_2,G} - Cr_{r_3,G}) + N(0, 0.5) \times (Cr_{r_4,G} - Cr_{r_5,G}) + N(0, 0.5) \times (Cr_{r_6,G} - Cr_{r_7,G}) \quad (10)$$

$$F = F_{r_1,G} + N(0, 0.5) \times (F_{r_2,G} - F_{r_3,G}) + N(0, 0.5) \times (F_{r_4,G} - F_{r_5,G}) \quad (11)$$

$$Cr = Cr_{r_1,G} + N(0, 0.5) \times (Cr_{r_2,G} - Cr_{r_3,G}) + N(0, 0.5) \times (Cr_{r_4,G} - Cr_{r_5,G}) \quad (12)$$

Table 2. The function values for 30 independent runs achieved via different variants of DE, SAOMDE, ATMES, and SMES, where b is the best value obtained, M is the mean, Sd is the standard deviation, and *Stat. Test* is the non-parametric statistical test results.

Prob.		Var1	Var2	Var3	SAOMDE	ATMES	SMES
g01	<i>b</i>	-15.0000	-15.0000	-15.0000	-15.0000	-15.0000	-15.0000
	<i>M</i>	-15.0000	-15.0000	-15.0000	-15.0000	-15.0000	-15.0000
	<i>Sd.</i>	0.000E+00	0.000E+00	2.783E-12	0.000E+00	1.6E-14	0.00E+00
g02	<i>b</i>	-0.792607	-0.803616	-0.803577	-0.803619	-0.803339	-0.803601
	<i>M</i>	-0.7277786	-0.7794139	-0.8014389	-0.790192	-0.790148	-0.785238
	<i>Sd.</i>	5.268E-02	1.902E-02	4.078E-03	9.624E-03	1.3E-02	1.67E-02
g03	<i>b</i>	-1.001E+00	-1.001E+00	-8.795E-01	-1.001E+00	-1.000E+00	-1.000E+00
	<i>M</i>	-1.001E+00	-1.001E+00	-5.136E-01	-1.001E+00	-1.000E+00	-1.000E+00
	<i>Sd.</i>	5.483E-05	1.290E-05	2.003E-01	1.943E-06	5.9E-05	2.09E-05
g04	<i>b</i>	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539
	<i>M</i>	-30665.539	-30662.699	-30665.539	-30665.539	-30665.539	-30665.539
	<i>Sd.</i>	3.549E-07	8.658E+00	9.254E-05	0.000E+00	7.4E-12	0.00E+00
g05	<i>b</i>	5126.4986	5126.4967	5126.4967	5126.4967	5126.498	5126.599
	<i>M</i>	5139.2409	5133.7068	5146.1066	5126.4967	5127.648	5174.492
	<i>Sd.</i>	3.836E+01	1.992E+01	5.875E+01	4.671E-06	1.8E+00	5.006E+01
g06	<i>b</i>	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814
	<i>M</i>	-6961.814	-6937.417	-6961.813	-6961.814	-6961.814	-6961.284
	<i>Sd.</i>	2.536E-10	4.785E+01	6.564E-04	0.000E+00	4.6E-12	1.85E+00
g07	<i>b</i>	24.307	24.307	24.358	24.306	24.306	24.327
	<i>M</i>	24.3562	24.3289	24.5971	24.3273	24.316	24.475
	<i>Sd.</i>	7.264E-02	2.528E-02	1.343E-01	1.938E-02	1.1E-02	1.32E-01
g08	<i>b</i>	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825
	<i>M</i>	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825
	<i>Sd.</i>	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2.8E-17	0.00E+00
g09	<i>b</i>	680.630	680.630	680.631	680.630	680.630	680.632
	<i>M</i>	680.635	680.634	680.638	680.630	680.639	680.643
	<i>Sd.</i>	1.108E-02	3.979E-03	5.860E-03	3.525E-04	1.0E-02	1.55E-02
g10	<i>b</i>	7052.9074	7049.5698	7139.2700	7049.2483	7052.2530	7051.9030
	<i>M</i>	7207.3266	7105.8437	7259.3120	7091.1806	7250.4370	7253.0470
	<i>Sd.</i>	1.074E+02	7.252E+01	7.076E+01	3.058E+01	1.2E+02	1.36E+02
g11	<i>b</i>	0.7499	0.7499	0.7499	0.7499	0.75	0.75
	<i>M</i>	0.7499	0.7499	0.7499	0.7499	0.75	0.75
	<i>Sd.</i>	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.4E-04	1.52E-04
g12	<i>b</i>	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
	<i>M</i>	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
	<i>Sd.</i>	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.0E-03	0.00E+00
g13	<i>b</i>	0.0539415	0.0539452	0.0539415	0.0539415	0.053950	0.053986
	<i>M</i>	0.0539419	0.0602843	0.0821369	0.0539418	0.053959	0.166385
	<i>Sd.</i>	1.409E-06	9.850E-03	1.352E-01	2.669E-07	1.3E-05	1.77E-01
Stat. Test	<i>b</i>	≈	+	+	Null	≈	+
	<i>M</i>	+	+	+	Null	≈	+

For other parameters settings, PS was set to 80 individuals, the minimum subpopulation size is set to 10% of PS , $WS = 50$ iterations. SAOMDE has been run 30 times for each test problem, where the stopping criterion is to run up to 240K fitness evaluations (FEs). The results, for SAOMDE and each independent DE variant, are presented in Table 2. We also consider two other state-of-the-art algorithms known as (i) Adaptive Tradeoff Model with evolution strategy (ATMES) [18], and (ii) Multimembered evolution strategy (SMES) [8]. We must mention here SAOMDE, and other two algorithms ATMES and SMSS use 240K FEs. The parameter ε for SAMODE is 1.0E-04, while it is set to 4.0E-04 and 5.0E-06 for SMES and ATMES, respectively. The results are based on 30 independent runs for all algorithms. The best (b), mean (M) and standard deviation (Sd) results, for 13 test problems [6], are presented in Table 2.

As compared to the other three independent DE variants, SAOMDE is always better or the same for all test problems, in regards to the best solutions obtained. In regards to the average results, SAOMDE is better or the same for all of the test problems, except g02. In regards to the other two algorithms, we found that SAOMDE is always better or the same in regards to the best solutions obtained, while ATMES is better only for g09, in regards to the average results.

It is also possible, however, to study the difference between any two stochastic algorithms in a more meaningful way. To this end, we have performed statistical significant testing. We have chosen a non-parametric test, Wilcoxon Signed Rank Test [2], that allows us to judge the difference between paired scores when it cannot make the assumption required by the paired-samples t test, such as that the populations should be normally distributed. Based on the best/or average results, and with a 5% significance level, we assign one of three signs (+, -, and \approx) (shown in Table 2 in the last 2 rows, for best and mean results, respectively), where “+” sign means SAOMDE is significantly better than the other algorithm, “-” sign means that SAOMDE is significantly worse, and “ \approx ” sign means that there is no significant difference between the two algorithms.

From the results, it can be seen that SAOMDE is better than each independent DE variant. For the state-of-the art algorithms, SAOMDE is better than SMES in regards to the best and the average results, while there is no significant difference with ATMES.

5 Conclusions

In this paper, we have shown that the efficiency of DE algorithms can be improved by adopting a concept of self-adaptation with multiple DE variants, in which the population is divided into a number of subpopulations where one subpopulation is allocated to each DE variant with a meaningful learning strategy to either increase or decrease, or keep unchanged the subpopulation size, adaptively, depending on the evolution success. An index for measuring the evolution success was also introduced. The algorithm showed a significantly better performance than the independent DE variants.

References

1. Abbass, H.: The Self-Adaptive Pareto Differential Evolution Algorithm. In: Proc. IEEE Congress on Evolutionary Computation (CEC), pp. 831–836 (2002)
2. Corder, G.W., Foreman, D.I.: Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach. John Wiley, Hoboken (2009)

3. Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter evolution. *IEEE Trans. Evol. Comput.* 10(4), 371–395 (2002)
4. Gamperle, R., Muller, S.D., Koumoutsakos, A.: Parameter study for differential evolution. In: Proc. WSEAS NNA-FSFS-EC, Interlaken, Switzerland, pp. 293–298 (2002)
5. Iorio, A., Li, X.: Solving rotated multi-objective optimization problems using differential evolution. In: Australian Conference on Artificial Intelligence, Cairns, Australia, pp. 861–872 (2004)
6. Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello, C.A.C., Deb, K.: Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. Technical Report, Nanyang Technological University, Singapore (2005)
7. Mallipeddi, R., Mallipeddi, S., Suganthan, P.N.: Ensemble strategies with adaptive evolutionary programming. *Information Sciences* 180(9), 1571–1581 (2010)
8. Mezura-Montes, E., Coello, C.A.C.: A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Trans. Evol. Comput.* 9(1), 1–17 (2005)
9. Mezura-Montes, E., Reyes, J.V., Coello, C.A.C.: A Comparative Study of Differential Evolution Variants for Global Optimization. In: Proc. Genetic Evol. Comput. Conf. (GECCO), pp. 485–492 (2006)
10. Mezura-Montes, E., Velazquez-Reyes, J., Coello, C.A.C.: Modified differential evolution for constrained optimization. In: Proc. IEEE Congr. Evol. Comput. (CEC), Vancouver, BC, pp. 25–32 (2006)
11. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential evolution: a practical approach to global optimization. Natural Computing Series. Springer, Berlin (2005)
12. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation* 13, 398–417 (2009)
13. Ronkkonen, J., Kukkonen, S., Price, K.V.: Real parameter optimization with differential evolution. In: Proc. IEEE Congress on Evolutionary Computation, vol. 1, pp. 506–513. IEEE Press, Piscataway (2005)
14. Storn, R., Price, V.: Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report, institute of Company Secretaries of India, Chennai, Tamil Nadu., TR-95-012 (1995)
15. Storn, R.: On the usage of differential evolution for function optimization. In: Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 519–523. IEEE, Berkeley (1996)
16. Tasgetiren, M.F., Suganthan, P.N., Pan, Q., Mallipeddi, R., Sarman, S.: An Ensemble of Differential Evolution Algorithms for Constrained Function Optimization. In: Proc. IEEE Congress on Evolutionary Computation (CEC), pp. 967–975 (2010)
17. Ting, C., Huang, C.: Varying Number of Difference Vectors in Differential Evolution. In: Proc. IEEE Congress on Evolutionary Computation (CEC), pp. 1351–1358 (2009)
18. Wang, Y., Cai, Z., Zhou, Y., Zeng, W.: An adaptive tradeoff model for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* 12(1), 80–92 (2008)

A New Expansion of Cooperative Particle Swarm Optimization

Hong Zhang

Department of Brain Science and Engineering, Kyushu Institute of Technology
2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan
zhang@brain.kyutech.ac.jp

Abstract. We previously proposed multiple particle swarm optimizers with diversive curiosity (MPSO α /DC). Its main features are to introduce diversive curiosity and localized random search into MPSO to comprehensively manage the trade-off between exploitation and exploration for preventing stagnation and improving the search efficiency. In this paper, we further extend these features to multiple particle swarm optimizers with inertia weight and multiple canonical particle swarm optimizers to create two analogues, called MPSOIW α /DC and MCPSO α /DC. To demonstrate the effectiveness of these proposals, computer experiments on a suite of multidimensional benchmark problems are carried out. The obtained results show that the search performance of the MPSO α /DC is superior to that of both the MPSOIW α /DC and MCPSO α /DC, and they have better search efficiency compared to other methods such as the convenient cooperative PSO and a real-coded genetic algorithm.

Keywords: cooperative particle swarm optimization, trade-off between exploitation and exploration, parallel processing, curiosity, stagnation, a real-coded genetic algorithm.

1 Introduction

The technique of particle swarm optimization (PSO) [5,8] has been widely applied to different fields of science, technology, and applications for treating with various optimization problems. During the past several years, a number of studies and investigations on cooperative PSO in relation to symbiosis, group behavior, and sensational synergy are in the researcher's spotlight [6]. Due to improvement of search performance, many kinds of cooperative PSO such as hybrid PSO and multi-population PSO etc. were published [1,3,7,11].

For realization of efficient searching and reinforcement of swarm intelligence, we proposed multiple particle swarm optimizers with diversive curiosity (MPSO α /DC) [16]. Compared to the convenient cooperative PSO, it has the following outstanding features: (1) Decentralization in multi-swarm exploration with hybrid search (MPSO α , where α refers to hybrid search); (2) Concentration in evaluation and behavior control with diversive curiosity (DC); (3) Their effective combination. According to these, MPSO α /DC could be expected to alleviate

stagnation and to enhance the search efficiency in treating with various optimization problems by comprehensively managing the trade-off between exploitation and exploration in the multi-swarm's heuristics.

Based on the above mentioned advantages of the MPSO α /DC, in this paper we further extend its major characteristics to multiple particle swarm optimizers with inertia weight (MPSOIW) [12] and multiple canonical particle swarm optimizers (MCPSO) [4] respectively to create two new analogues, i.e. multiple particle swarm optimizers with inertia weight with diversive curiosity (MPSOIW α /DC) and multiple canonical particle swarm optimizers with diversive curiosity (MCPSO α /DC). The aim to technical expansion is not only to improve the search performance of both the MPSOIW and MCPSO, but also to promote the technical innovation of cooperative PSO for making different optimizers more efficiency with additional search strategy and sound judgment.

2 Basic Algorithms

For the sake of the following description, let the search space be N -dimensional, $\Omega \in \Re^N$, the number of particles in a swarm be P , the position of i th particle be $\mathbf{x}^i = (x_1^i, x_2^i, \dots, x_N^i)^T$, and its velocity be $\mathbf{v}^i = (v_1^i, v_2^i, \dots, v_N^i)^T$.

The PSO: In the beginning of the PSO search, the particle's position and velocity are generated in random, then they are updated by

$$\begin{cases} \mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_{k+1}^i \\ \mathbf{v}_{k+1}^i = w_0 \mathbf{v}_k^i + w_1 \mathbf{r}_1 \otimes (\mathbf{p}_k^i - \mathbf{x}_k^i) + w_2 \mathbf{r}_2 \otimes (\mathbf{q}_k - \mathbf{x}_k^i) \end{cases} \quad (1)$$

where w_0 is an inertia coefficient, w_1 and w_2 are coefficients for individual confidence and swarm confidence. $\mathbf{r}_1, \mathbf{r}_2 \in \Re^N$ are two random vectors in which each element is uniformly distributed over $[0, 1]$, and \otimes is an element-wise operator for vector multiplication. $\mathbf{p}_k^i (= \arg \max_{j=1, \dots, k} \{g(\mathbf{x}_j^i)\})$, where $g(\cdot)$ is the criterion value of i th particle at time-step k .) is the local best position of i th particle up to now, and $\mathbf{q}_k (= \arg \max_{i=1, 2, \dots} \{g(\mathbf{p}_k^i)\})$ is the global best position among the whole swarm. In the original PSO, $w_0 = 1.0$ and $w_1 = w_2 = 2.0$ are used [8].

The PSOIW: To overcome the weakness of the original PSO search in convergence, Shi et al. modified the update rule of velocity in it by constant reduction of the inertia coefficient over time-step as follows [12].

$$\mathbf{v}_{k+1}^i = w(k) \mathbf{v}_k^i + w_1 \mathbf{r}_1 \otimes (\mathbf{p}_k^i - \mathbf{x}_k^i) + w_2 \mathbf{r}_2 \otimes (\mathbf{q}_k - \mathbf{x}_k^i) \quad (2)$$

where $w(k)$ is a variable inertia weight which is linearly reduced from a starting value, w_s , to a terminal value, w_e , with the increment of time-step k as

$$w(k) = w_s + \frac{w_e - w_s}{K} \times k \quad (3)$$

where K is the maximum number of iteration for the PSOIW run. In the original PSOIW, the boundary values, $w_s = 0.9$ and $w_e = 0.4$, are adopted.

The CPSO: Another PSO implementation dubbed PSO constriction coefficient was developed by Clerc [4]. The particle's velocity in it is defined as

$$\mathbf{v}_{k+1}^i = \chi \left(\mathbf{v}_k^i + c_1 \mathbf{r}_1 \otimes (\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2 \mathbf{r}_2 \otimes (\mathbf{q}_k - \mathbf{x}_k^i) \right) \quad (4)$$

where χ is a constriction coefficient. Based on the quantitative analysis to the optimizer, $\chi = 0.7298$ and $c_1 = c_2 = 2.05$ are used in the original CPSO.

In comparison with the update rules of the velocity in Eq.(1) and Eq.(2), the equivalent relationship of the coefficients between the PSO and the CPSO, i.e. $\chi = w_0$, $\chi c_1 = w_1$, and $\chi c_2 = w_2$, can be easily obtained. Since the value of χ is less than 1.0, the convergence of the CPSO search is surely certified.

3 The Proposed Methods

The MPSO α /DC is the creation of the particle swarm optimization with dispersive curiosity (PSO/DC) [14], which provides a good framework for other cooperative PSO to alleviate stagnation and to improve the search efficiency by comprehensively managing the trade-off between exploitation and exploration in their heuristics. Moreover, different variants from the MPSO α /DC can be easily developed through the replacement of an implementation object with the above mentioned characteristics. Fig. 1 shows a common flowchart of the MPSO α /DC, MPSOIW α /DC, and MCPSO α /DC.

In comparison with the PSO/DC in construction, it is self-evident that the most difference (yellow parts) is that plural particle swarms (the subscript S

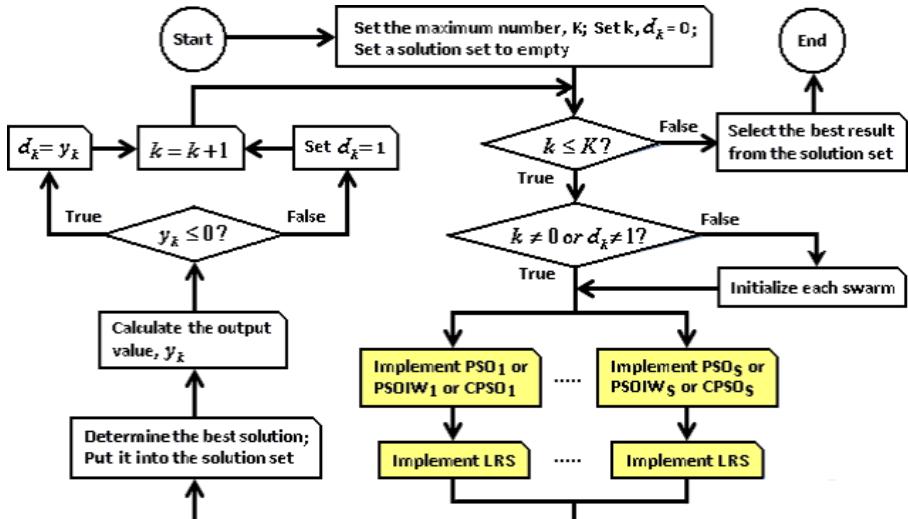


Fig. 1. A common flowchart of the MPSO α /DC, MPSOIW α /DC, and MCPSO α /DC

of the PSO_S , PSOIW_S , and CPSO_S stands for the total number of the used particle swarms) are implemented in parallel, and a localized random search (LRS) is applied to correct the solution found by each particle swarm. The hybrid search, i.e. PSO and LRS, is considered to be close to the HGAPSO [7] in search effect. Concretely, the procedure of the LRS is implemented as follows.

- (1) Give the terminating condition, J , and set $j = 1$. Set $\mathbf{q}_{now}^s = \mathbf{q}_k^s$ (\mathbf{q}_k^s be a solution found by s th particle swarm at time-step k , where $s = 1, 2, \dots, S$.)
- (2) Generate a random data, $\mathbf{d}_j \in \mathbb{R}^N \sim N(0, \sigma_N^2)$ (σ_N is a small positive value). Check whether $\mathbf{q}_k^s + \mathbf{d}_j \in \Omega$ is satisfied or not. If $\mathbf{q}_k^s + \mathbf{d}_j \notin \Omega$ then adjust \mathbf{d}_j for moving $\mathbf{q}_k^s + \mathbf{d}_j$ to the nearest valid point within Ω .
- (3) Set $\mathbf{q}_{new} = \mathbf{q}_k^s + \mathbf{d}_j$. If $g(\mathbf{q}_{new}) > g(\mathbf{q}_{now}^s)$ then set $\mathbf{q}_{now}^s = \mathbf{q}_{new}$.
- (4) Set $j = j + 1$. If $j \leq J$ then go to (2).
- (5) Set $\mathbf{q}_k^s = \mathbf{q}_{now}^s$ to correct the solution at time-step k . Stop the search.

Then, y_k shown in Fig. 1 is the output of internal indicator of the multi-swarm for distinguishing the search behavior [14], which is defined as

$$y_k(L, \varepsilon) = \max\left(\varepsilon - \sum_{l=1}^L \frac{|g(\mathbf{q}_k^b) - g(\mathbf{q}_{k-l}^b)|}{L}, 0\right) \quad (5)$$

where $\mathbf{q}_k^b (= \arg \max_{s=1,\dots,S} \{g(\mathbf{q}_k^s)\})$ denotes the best solution at time-step k . L is a duration of judgment, and ε is a positive tolerance coefficient. This is based on the Loewenstein's doctrine: “diversive curiosity” [2] occupies a critical position at the crossroad of cognition and motivation” in [9]. Specifically, If $y_k = 0$ then set the control signal d_k to 0, this means that the multi-swarm is continually exploring the surroundings of the solution \mathbf{q}_k^b for “cognition”. If once y_k become positive then set d_k to 1 for initialization, this indicates that the multi-swarm has no interest in the surrounding of the solution \mathbf{q}_k^b for “motivation”. It is clear that the internal indicator represents diversive curiosity which plays a central role in preventing stagnation (called boredom behavior in search) for every method.

4 Computer Experiments

In the following computer experiments, we use a suite of multidimensional benchmark problems in Table I to facilitate comparison and analysis of the performance indexes of the proposed methods. For convenience' sake, three particle swarms ($S=3$) are used to explore in each proposed method.

Fig. 2 shows the changes of the best fitness values found by each proposed method, i.e. MPSO* α /DC, MPSOIW α /DC, and MCPSO* α /DC. Note that the the sign * refers to the appropriate values of parameters in the optimizers, which are estimated by the methods of evolutionary particle swarm optimization [4][5]. According to the change of the obtained best fitness value of every particle swarm and their synchronous change, decentralization in multi-swarm search and concentration in behavior control with diversive curiosity in search

Table 1. Functions and criteria to the given suite of benchmark problems. The search space for each benchmark problem is limited to $\Omega \in (-5.12, 5.12)^N$.

Problem	Function	Criterion
<i>Sphere</i>	$f_{Sp}(\mathbf{x}) = \sum_{d=1}^N x_d^2$	$g_{Sp}(\mathbf{x}) = \frac{1}{f_{Sp}(\mathbf{x}) + 1}$
<i>Griewank</i>	$f_{Gr}(\mathbf{x}) = \frac{1}{4000} \sum_{d=1}^N x_d^2 - \prod_{d=1}^N \cos\left(\frac{x_d}{\sqrt{d}}\right) + 1$	$g_{Gr}(\mathbf{x}) = \frac{1}{f_{Gr}(\mathbf{x}) + 1}$
<i>Rastrigin</i>	$f_{Ra}(\mathbf{x}) = \sum_{d=1}^N \left(x_d^2 - 10 \cos(2\pi x_d) + 10 \right)$	$g_{Ra}(\mathbf{x}) = \frac{1}{f_{Ra}(\mathbf{x}) + 1}$
<i>Rosenbrock</i>	$f_{Ro}(\mathbf{x}) = \sum_{d=1}^{N-1} \left[\left(100(x_{d+1} - x_d^2) \right)^2 + (x_d - 1)^2 \right]$	$g_{Ro}(\mathbf{x}) = \frac{1}{f_{Ro}(\mathbf{x}) + 1}$

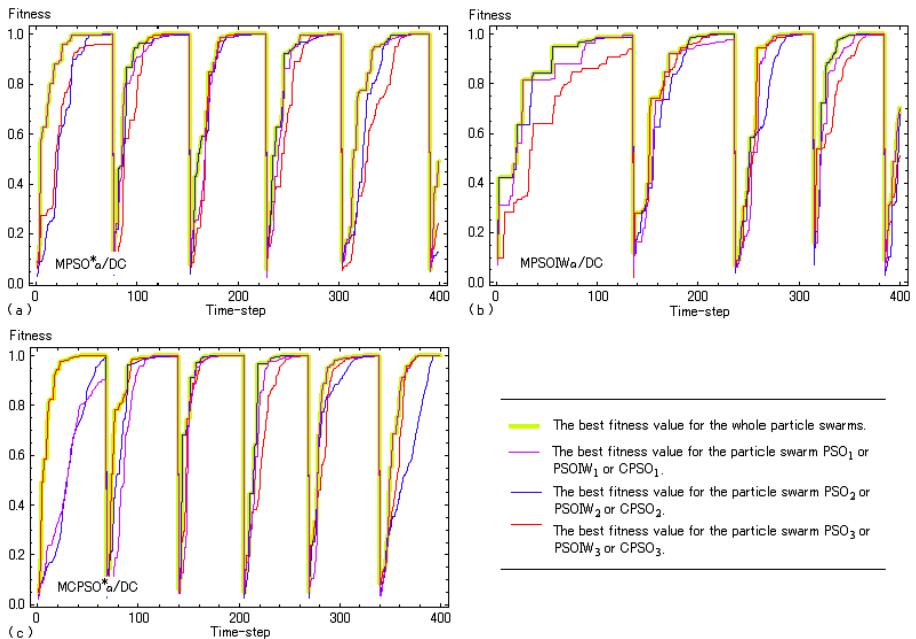


Fig. 2. The changes of the best fitness values obtained by each proposed method under the parameters, $L = 50$ and $\varepsilon = 0.01$, for the *Sphere* problem

process are observed for every method. It is clear that the increment of the frequencies of re-initializations in a fixed period of running time-step will lead to get higher discover probability which finds an optimal solution corresponding to the given optimization problem. Therefore, the search performance of these methods could be greatly improved by comprehensively managing the trade-off between exploitation and exploration in their heuristics.

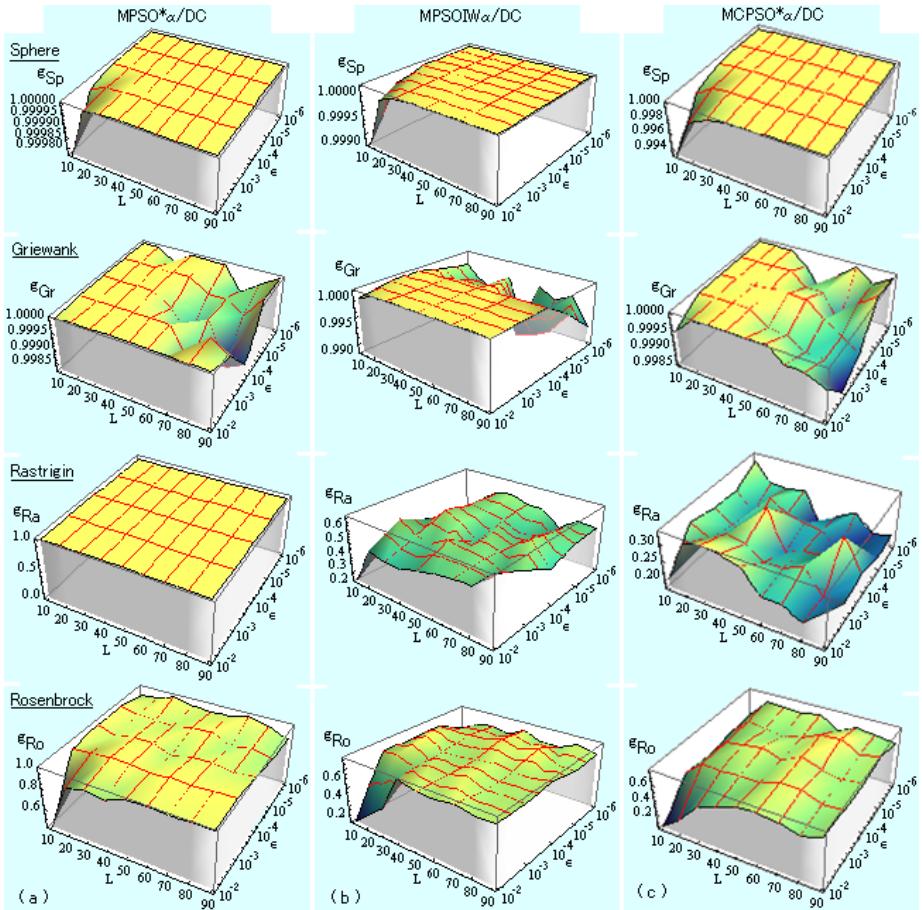


Fig. 3. Distribution of the obtained averages of the best fitness values of each proposed method with tuning the parameters, L and ε , for each problem (20 trials/per case).
 (a) MPSO* α /DC, (b) MPSOIW α /DC, (c) MCPSO* α /DC.

Furthermore, depend on the different frequencies of re-initializations shown in Fig. 2 (a) (b) (c), the relationship of the resulting convergence among the MPSO* α /DC, MPSOIW α /DC, and MCPSO* α /DC is approximately expressed as follows.

$$\text{MCPSO}^*\alpha/\text{DC} \succ \text{MPSO}^*\alpha/\text{DC} \succ \text{MPSOIW}^*\alpha/\text{DC}$$

It objectively reflects the intrinsic characteristics of each method in convergence. To investigate their search performance, Fig. 3 gives the obtained experiment results of the MPSO* α /DC, MPSOIW α /DC, and MCPSO* α /DC, i.e. the average of the best fitness values with tuning the parameters, L and ε . Based on the distribution of the fitness values, the recommended ranges of the parameters, L^* and ε^* , corresponding to each problem are available for

Table 2. The mean and standard deviation of criterion values in each method for each 5D benchmark problem with 20 trials

Problem	MPSO* α /DC	PSO*/DC	EPSO	OPSO	RGA/E
<i>Sphere</i>	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000	0.998 \pm 0.001
<i>Griewank</i>	1.000 \pm 0.000	1.000 \pm 0.000	0.987 \pm 0.010	0.944 \pm 0.043	0.796 \pm 0.117
<i>Rastrigin</i>	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000	0.265 \pm 0.118	0.961 \pm 0.023
<i>Rosenbrock</i>	0.989 \pm 0.012	0.625 \pm 0.232	0.469 \pm 0.280	0.392 \pm 0.197	0.372 \pm 0.136

the proposed methods. By comparison with their best fitness values in Fig. ③ we can see that the search performance of the MPSO* α /DC is better than the MPSOIW* α /DC and MCPSO* α /DC specially for the *Rastrigin* and *Rosenbrock* problems. It is reflected that the MPSO* α /DC has better adaptability. Whereas on the MPSOIW α /DC, its search performance will be expected to be better, if the appropriate values of parameters in the used PSOIW are available.

Furthermore, we compare the search performance with the other methods such as the PSO*/DC, EPSO, OPSO (optimized particle swarm optimization) [10], and RGA/E (a real-coded genetic algorithm with elitism strategy) [13]. The search performance of the MPSO* α /DC, which is a delegate from the proposed methods, is better than that by those methods by comparison with the average of criterion values in Table ②. These results show that the union of multiple hybrid search and diversive curiosity takes the active role in dealing with the given benchmark problems.

In particular, we observe that a big increase, i.e. the average of criterion values by implementing the MPSO* α /DC steeply rises from 0.469 to 0.989, in search performance is achieved well for the *Rosenbrock* problem. It is apparent that the curiosity-driven multi-swarm search is a powerful and effective method.

5 Conclusions

In this paper, we presented three methods of cooperative PSO, i.e. the MPSO α /DC, MPSOIW α /DC, and MCPSO α /DC. Based on the essential concept of decentralization in search and concentration in evaluation and behavior control, the combination of the adopted hybrid search and the execution of diversive curiosity, theoretically, they have good capability which improves the search accuracy to solutions and prevents stagnation by comprehensively managing the trade-off between exploitation and exploration in their heuristics.

Applications of the proposed methods to a suite of multidimensional benchmark problems well demonstrated their respective effectiveness. The experimental results verified that the MPSO α /DC among them has better latent capability in efficiently handling the given different optimization problems.

It is left for further study in applying the methods of curiosity-driven multi-swarm to the applications of data mining, model selection, system identification, multiobjective optimization, neural network training, and so on.

Acknowledgment. This research was supported by Grant-in-Aid Scientific Research(C)(22500132) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. van den Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 8(3), 225–239 (2004)
2. Berlyne, D.: Conflict, Arousal, and Curiosity. McGraw-Hill Book Co., New York (1960)
3. Chang, J.F., Chu, S.C., Roddick, J.F., Pan, J.S.: A parallel particle swarm optimization algorithm with communication strategies. *Journal of Information Science and Engineering* 21, 809–818 (2005)
4. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 6(1), 58–73 (2000)
5. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39–43 (1995)
6. El-Abd, M., Kamel, M.S.: A Taxonomy of Cooperative Particle Swarm Optimizers. *International Journal of Computational Intelligence Research* 4(2), 137–144 (2008)
7. Juang, C.-F.: A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. *IEEE Transactions on Systems, Man and Cybernetics Part B* 34(2), 997–1006 (2004)
8. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia, pp. 1942–1948 (1995)
9. Loewenstein, G.: The Psychology of Curiosity: A Review and Reinterpretation. *Psychological Bulletin* 116(1), 75–98 (1994)
10. Meissner, M., Schmuker, M., Schneider, G.: Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. *BMC Bioinformatics* 7(125) (2006)
11. Niu, B., Zhu, Y., He, X.: Multi-population Cooperation Particle Swarm Optimization. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) *ECAL 2005. LNCS (LNAI)*, vol. 3630, pp. 874–883. Springer, Heidelberg (2005)
12. Shi, Y., Eberhart, R.C.: A modified particle swarm optimiser. In: *Proceedings of the IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, USA, pp. 69–73 (1998)
13. Zhang, H., Ishikawa, M.: Evolutionary Particle Swarm Optimization – Metaoptimization Method with GA for Estimating Optimal PSO Methods. In: Castillo, O., et al. (eds.) *Trends in Intelligent Systems and Computer Engineering. LNEE*, vol. 6, pp. 75–90. Springer, Heidelberg (2008)
14. Zhang, H., Ishikawa, M.: Characterization of particle swarm optimization with diversive curiosity. *Journal of Neural Computing & Applications*, 409–415 (2009)
15. Zhang, H., Ishikawa, M.: The performance verification of an evolutionary canonical particle swarm optimizers. *Neural Networks* 23(4), 510–516 (2010)
16. Zhang, H.: Multiple Particle Swarm Optimizers with Diversive Curiosity. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 (IMECS 2010)*, Hong Kong, pp. 174–179 (2010)

Adaptive Ensemble Learning Strategy Using an Assistant Classifier for Large-Scale Imbalanced Patent Categorization

Qi Kong^{1,2}, Hai Zhao^{1,2}, and Bao-Liang Lu^{1,2,*}

¹ Center for Brain-Like Computing and Machine Intelligence,
Department of Computer Science and Engineering

² MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University, 800 Dongchuan Rd., Shanghai, China, 200240
b1u@cs.sjtu.edu.cn

Abstract. Automatic patent classification is of great practical value for saving a lot of resources and manpower. As real patent classification tasks are often very-large scale and serious imbalanced such as patent classification, using traditional pattern classification techniques has shown inefficient and ineffective. In this paper, an adaptive ensemble learning strategy using an assistant classifier is proposed to improve generalization accuracy and the efficiency. The effectiveness of the method is verified on a group of real patent classification tasks which are decomposed in multiple ways by using different algorithms as the assistant classifiers.

1 Introduction

Automatic patent classification is of great importance. Current patent classification mainly relies on human experts while a large-scale of patents are issued annually, e.g., more than 300,000 Japanese patents per year. This circumstance asks for effective automatic processing techniques. However, patent classification is too large a problem to adopt many popular machine learning algorithms.

In detail, patent classification is a large-scale, hierarchical structure and imbalanced text classification task. A lot of works have been done on the task. Previous works mainly focus on a single classifier and don't bring up satisfied results up to now [4]. Typically, such a single classifier may be the state-of-the-art method, SVM, which requires solving a quadratic optimization problem and costing training time that is at least quadratic to the number of training samples. Therefore, even for the efficient training algorithms for SVM such as SMO, large-scale classification problems are still too tough to tackle.

A parallel and modular method named MIN-MAX-modular (M^3), was proposed in [2], in which a complicated classification problem may be divided into

* Corresponding author.

many smaller independent binary classification problems. Based on MIN-MAX-modular network, Assistant Classifier Learning Strategy (ACLS for short) is proposed in this paper to improve the generalization performance of patent classification and the efficiency as well.

The proposed scheme allows a very large and imbalanced binary classification problem to be divided into independent binary balanced sub-problems. In detail, for the training phase, a large imbalanced training data set will be decomposed into many balanced training subsets and be processed in parallel. Then base learners are trained on all these subsets independently. For each sample in the original training set, the outputs of these base learners are made into vectors and an assistant classifier will learn from the vectors to automatically find an effective ensemble way to output the original class label for each given sample. With all base learners and an assistant ensemble classifier trained, for the recognition phase, an unknown sample is presented to all the base learners; the outputs of all the learners are integrated to make a final solution to the original problem according to the assistant classifier.

The rest of this paper is organized as follows. Section 2 proposes Assistant classifier Learning Strategy and describes its two steps. Section 3 gives the experimental results, which include the comparisons of traditional SVM and ACLS on different decomposition and module combination methods on Japanese patent data set. Section 4 concludes this work.

2 Assistant Classifier Based Module Selection Strategy for Parallel Balanced Learning

Assistant Classifier based Module Selection Strategy (ACMSS) is a novel integration strategy for MIN-MAX-modular classifier (M^3). M^3 classifier is a general framework which is able to solve patent classification problems which is large-scale and imbalanced in a parallel way based on the conquer-and-divide idea.

2.1 Task Decomposition

A large imbalanced real patent training data set is first decomposed into smaller and balanced training sets and parallel processed in the task decomposition phase. The decomposition method has been described in [13,10,35].

Two-class classification is one kind of basic classification problem. Many essential classification schemes often start from binary classifier and then adapt to multi-class classifiers. Let $\mathcal{T} = \{(X_l, Y_l)\}_{l=1}^L$ be the training set of a K -class classification problem and the K classes are represented by C_1, C_2, \dots, C_K , respectively. $X_l \in R^d$ is the input vector, $Y_l \in R^K$ is the expected output, and L is the number of training samples. Suppose the K training input sets, $\mathcal{X}_1, \dots, \mathcal{X}_K$ are expressed as $\mathcal{X}_i = \{X_l^i\}_{l=1}^{L_i}$ for $i = 1, \dots, K$, where L_i is the number of training samples in class C_i , X_l^i is the l -th sample belongs to class C_i and all of $X_l^{(i)} \in \mathcal{X}_i$ have the same expected outputs and $\sum i = 1^K L_i = L$. According to the M^3 network, a K -class problem can be divided into $K \times (K - 1)$

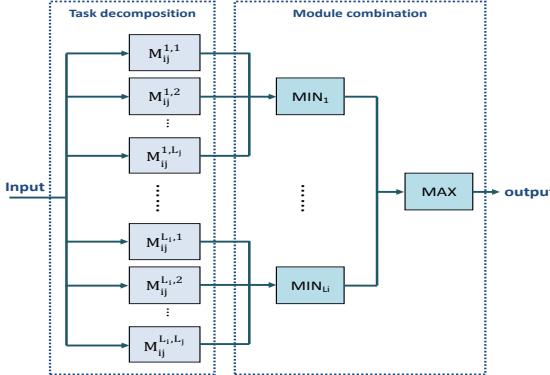


Fig. 1. M^3 contents $L_i \times L_j$ individual network modules, L_i min nodes, and one max node

two-class problems that are trained independently, each of which is given by $T_{ij} = \{(X_l^{(i)}, +1)\}_{l=1}^{L_i} \cup \{(X_l^{(j)}, -1)\}_{l=1}^{L_j}$, for $i = 1, \dots, K-1$ and $j = i+1, \dots, K$. If these two-class problems are still large-scale or imbalanced, they can be further decomposed into relatively smaller two-class problems until suitable for traditional classifiers.

Assume that the input set \mathcal{X}_i is further partitioned into N_i subsets in the form of $\mathcal{X}_{iu} = \{X_l^{(iu)}\}_{l=1}^{L_i^{(u)}}$ for $u = 1, \dots, N_i$, where $L_i^{(u)}$ is the number of training samples included in \mathcal{X}_{iu} and $\cup_u^{N_i} \mathcal{X}_{iu} = \mathcal{X}_i$. After dividing the training input set \mathcal{X}_i into N_i subsets \mathcal{X}_{iu} , the training set for each of the smaller and simpler two class problem can be given by $T_{ij}^{(u,v)} = \{X_l^{(iu),+1}\}_{l=1}^{L_i^{(u)}} \cup \{X_l^{(iv),-1}\}_{l=1}^{L_j^{(v)}}$, for $u = 1, \dots, N_i, v = 1, \dots, N_j, i = 1, \dots, K-1$ and $j = i+1, \dots, K$, where $X_l^{(iu)} \in \mathcal{X}_{iu}$ and $X_l^{(jv)} \in \mathcal{X}_{jv}$ are the input vectors belonging to class C_i and class C_j , respectively, $\sum_{u=1}^{N_i} L_i^{(u)}$ and $\sum_{v=1}^{N_j} L_j^{(v)}$.

After task decomposition, all of the two-class subproblems are treated as completely independent tasks in the learning phase. Therefore, all the two-class sub-problems are efficiently learned in a massively parallel way.

2.2 Module Combination Using Assistant Classifier

After the task decomposition phase, we have trained the modules which are assigned to learn associated sub-problems. An unknown sample is presented to all the base learners. Then the outputs of all the learners are integrated to make a final solution to the original problem.

Without losing the generality, if a base classifier gives a positive class prediction for a test sample, it will be denoted “1”, otherwise “0”. For convenience, the outputs of all base classifiers for a certain test sample are illustrated in a dot matrix with base classifiers in each row sharing the same positive training subsets and in each column sharing the same negative training subsets.

With all of the two-class sub-problems learnt by every corresponding base classifiers, all the trained classifiers are integrated into an M^3 classifier. The original integration strategy in [3] is rule-based. In detail, it requires two specific rules, the minimization principle and maximization principle. Figure 2 illustrates an M^3 classifier, where the function of the Min unit is to find a minimum value from its multiple inputs while Max unit is to find a maximum value.

Further combination algorithms or strategies, AMS, SMS and DTMS, have been proposed to improve the original combination strategy for M^3 classifier [11, 9, 6]. Asymmetric module selection strategy (AMS) is equal to the original MIN-MAX combination but with some heuristic speedup [8]. Symmetric module selection strategy (SMS) [8, 12] is not strictly equal to the original MIN-MAX combination any more, though it is still mostly motivated from the latter. In SMS, a pointer is presented to check each output of base classifier from the top-left corner of the matrix and determine the next to be checked base classifier according to the current output class label. The output class label of the original problem is determined by the position where the pointer is finally located. If the pointer will be at the rightmost column, then the output label will be the positive class, otherwise, the negative class. Examples for AMS and SMS are shown in Figure 2(a,b). Decision tree based module selection (DTMS) [6] is further improved from SMS. The different from the latter is that a decision tree is introduced instead to determine which base classifier should be checked. An example of DTMS is shown in Figure 2(c).

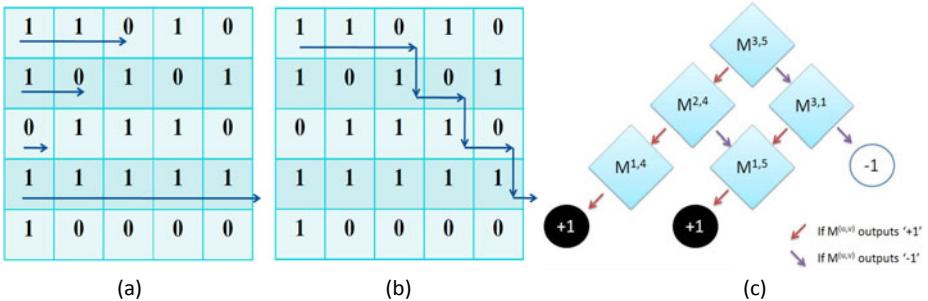


Fig. 2. Classifier selection algorithms: (a) AMS (b) SMS (c) DTMS

However, all the existing integration approaches, either AMS, SMS or DTMS only takes advantage of partial but not all outputs of the base classifiers. For a test sample, some base classifiers do not actually make any contribution for the final output. In addition, all these approaches should work on concrete classifiers whose output should be either 0 or 1, instead of continuous confidence values in the MIN-MAX module combination. ACMSS is thus proposed to overcome the above drawbacks of the existing methods.

ACMSS works in the following way. For the training phase, For each sample in the original training set, the outputs of those base learners are made into vectors

(shown in Figure 3). Then an assistant classifier will learn from the vectors to find an effective ensemble way automatically. With all base learners and an assistant ensemble classifier trained, for the recognition phase, an unknown sample is presented to all the base learners; the outputs of all the learners are integrated to make a final solution to the original problem according to the assistant classifier.

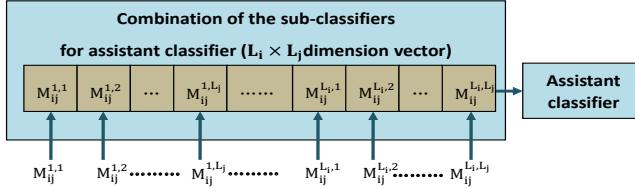


Fig. 3. Assistant classifier learns from the vectors composed by the outputs of $L_i \times L_j$ modules

ACMSS can be considered to acquire and absorb the knowledge which has been learnt by many weak base classifiers. We introduce ACMSS to compute the weights of base classifiers automatically to get a better performance on large-scale imbalanced data. We note that meta-learning in [4] is based on the combine of different classification algorithms while ACMSS is based on task decomposition and base classifiers' combination. The training set size of each base classifier of ACMSS is much smaller than those of meta-learning.

3 Experiments

The data sets used in our experiments are from the NTCIR-8 patent mining shared task which follows the International Patent Classification (IPC) taxonomy. Patent data set is large-scale, hierarchical and imbalanced.

The patent data of year 2002 are adopted in our experiments. The \mathcal{X}_{avg} feature selection method is used for vector space model construction. And the patents are indexed into a single vector with 5000 dimensional by using the *tfidf* algorithm. SVM are selected as the base classifier and the assistant classifier with linear kernel. C (the trade-off between training error and margin) is set to be $\frac{1}{avg.\|x\|^2}$ where x is the samples of the subset.

Data set is decomposed with various strategies according to [13]. These strategies include random(R-), hyperplane(H-) [10], centroid connection(C-) [7] and year and category based prior knowledge (YC-). The main idea of hyperplane and centroid connection decomposition methods is as shown in Figure 4. The prior knowledge decomposition method divides the date set by the information of the release data and IPC label of each patent.

We choose a subset of these Japanese patents of year 2002. The data contains 150,000 samples including 75,000 samples of section G (Physics patents) and 75,000 samples of section H (Electricity patents). There are 100,000 samples

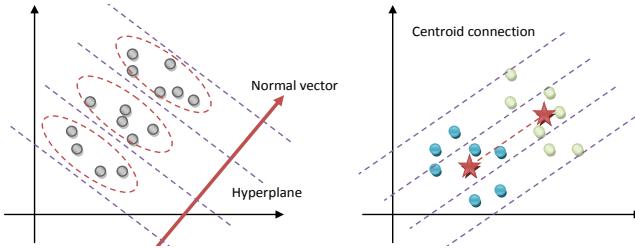


Fig. 4. Left is the hyperplane classification method. hyperplane is orthogonal to the normal vector which is produced by PCA. Right is the centroid connection method. The stars are the centroid of different categories. Samples are decomposed according to the distance to the centroid connection line.

Table 1. The experimental results of balanced and imbalanced patent data

	balanced data			imbalanced data		
	Precision%	Recall%	Macro- F_1 %	Precision%	Recall%	Macro- F_1 %
T-SVM	89.45	89.36	89.40	97.37	61.51	79.92
R-AMS	89.47	88.28	88.87	82.17	93.75	87.58
R-SMS	89.47	88.28	88.87	82.17	93.75	87.58
R-DTMS	88.65	88.42	88.53	83.24	93.78	88.20
R-ACMSS	89.27	90.16	89.71	88.37	88.80	88.58
H-AMS	88.32	87.13	87.72	81.21	92.77	86.61
H-SMS	88.32	87.13	87.72	81.21	92.77	86.61
H-DTMS	87.89	87.56	87.72	82.29	92.69	87.18
H-ACMSS	88.31	89.05	88.68	87.39	87.98	87.68
CC-AMS	89.46	88.33	88.89	82.18	93.73	87.58
CC-SMS	89.47	88.32	88.89	82.18	93.73	87.58
CC-DTMS	88.91	88.45	88.68	83.31	93.61	88.16
CC-ACMSS	89.37	90.16	89.76	88.47	89.08	88.77
YC-AMS	90.13	89.19	89.66	83.09	94.27	88.33
YC-SMS	90.13	89.13	89.66	83.09	94.27	88.33
YC-DTMS	89.94	89.52	89.73	84.35	94.59	89.18
YC-ACMSS	90.39	91.15	90.77	89.51	90.08	89.79

note: Decompose methods: R-Randomly, H-Hyperplane, CC-Centroid Connection, YC-prior knowledge(Year & Category) decomposition. Combination methods: AMS-asymmetric selection, SMS-symmetric selection. DTMS employs C4.5. The parameter \mathcal{C} of ACMSS on balanced data is 0.02, on imbalanced data is 0.025.

for the training and 50,000 for the test. Training data are decomposed into 25×25 sub-classification problems with 2,000 positive samples and 2,000 negative samples in each subset.

Similar with the above balanced patent data, we choose a subset of Japanese patents of year 2002 as an imbalanced data set. It contains 110,000 samples including 80,000 samples for training and 30,000 for test. Training set has 10,000

samples of section G and 70,000 samples of section H. Training data are decomposed into 5×35 sub-classification problems with 2,000 positive samples and 2,000 negative samples in each subset.

Four module selection methods (AMS, SMS, DTMS and ACMSS) are used for combination. The experimental results are shown in Table I.

From the experimental results, we can see that the ACMSS, using SVM as the assistant classifier, yields the best performance in most cases, especially with prior knowledge based decomposition.

Either AMS, SMS or DTMS only takes advantage of partial but not all outputs of the base classifiers. For a test sample, some base classifiers do not actually make any contribution for the final output. ACMSS can be considered to acquire and absorb the knowledge which has been learnt by weak base classifiers. In addition, ACMSS works on concrete classifiers whose output are continuous confidence values, instead of either 0 or 1 in AMS, SMS or DTMS. ACMSS computes the weights of base classifiers automatically to get a better performance on large-scale data.

4 Conclusion

Current patent classification mainly relies on human experts while a large-scale of patents are issued annually. This circumstance asks for effective automatic processing techniques. However, patent classification is too large and imbalanced a problem to adopt many popular machine learning algorithms. The problem is too tough to tackle for a single classifier and the imbalance of data set seriously affects the classification results.

Taking into account this difficulty, the focus is on seeking a parallel classification algorithm. M³ classifier is a general framework which is able to solve patent classification problems which is large-scale and imbalanced in a parallel way based on the conquer-and-divide idea. Complicated classification problem may be divided into many smaller balanced independent binary classification problems. With the popularity of distributed computing, the parallel learning strategy is showing strong advantages for solving practical problems.

ACMSS is proposed in this paper to improve the generalization performance and the efficiency. ACMSS computes the weights of base classifiers automatically to get a better performance on large-scale imbalanced data. ACMSS works on concrete classifiers whose output are continuous confidence values, instead of either 0 or 1 in AMS, SMS or DTMS. ACMSS can be considered to absorb the knowledge learnt by weak base classifiers employing an assistant classifier.

This research enables us to know further about large-scale imbalanced patent categorization strategies, especially the parallel methods based on the conquer-and-divide idea. ACMSS is of better adaptive ability and strong generalization since many classifier algorithm can be employed as the assistant classifier. The adaptive ensemble learning strategy that we propose performs the best in solving large-scale both balanced and imbalanced categorization problems as shown in the experiments, which is valuable for real-world applications.

Acknowledgements

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), the National High-Tech Research Program of China (Grant No. 2008AA02Z315), and the Science and Technology Commission of Shanghai Municipality (Grant No. 09511502400).

References

1. Fujino, A., Isozaki, H.: Multi-label classification using logistic regression models for NTCIR-7 patent mining task. In: Proceedings of NTCIR-7 Workshop Meeting, Tokyo, Japan (2008)
2. Lu, B.L., Bai, Y., Kita, H., Nishikawa, Y.: An efficient multilayer quadratic perceptron for pattern classification and function approximation. In: Proceeding of International Joint Conference on Neural Networks, pp. 1385–1388 (1993)
3. Lu, B.L., Ito, M.: Task decomposition and module combination based on class relations: A modular neural network for pattern classification. *IEEE Transactions on Neural Networks* 10(5)
4. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. *Artificial Intelligence Review* 18(2)
5. Wang, K., Zhao, H., Lu, B.L.: Task decomposition using geometric relation for min-max modular svms. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3496, pp. 887–892. Springer, Heidelberg (2005)
6. Wang, Y., Lu, B.L., Ye, Z.F.: Module combination based on decision tree in min-max modular network. In: Proceedings of International Conference on Neural Computation, Madeira, Portugal, pp. 555–558 (2009)
7. Ye, Z.F.: Parallel min-max modular support vector machine with application to patent classification (in chinese). Master Thesis, Shanghai Jiao Tong University (2009)
8. Zhao, H., Lu, B.L.: Improvement on response performance of min-max modular classifier by symmetric module selection. In: Proceeding of Second International Symposium Neural Networks, Chongqing, China, vol. 3971, pp. 39–44 (2005)
9. Zhao, H., Lu, B.: Analysis of fault tolerance of a combining classifier. In: Proceedings of 12th International Conference on Neural Information, Taipei, Taiwan, pp. 888–893 (2004)
10. Zhao, H., Lu, B.: Determination of hyperplane by pca for dividing training data set. In: Proceeding of 12th International Conference on Neural Information, Taipei, Taiwan, vol. 3173, pp. 755–760 (2005)
11. Zhao, H., Lu, B.: A general procedure for combining binary classifiers and its performance analysis. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3610, pp. 303–312. Springer, Heidelberg (2005)
12. Zhao, H., Lu, B.: A modular reduction method for k-NN algorithm with self-recombination learning. In: Wang, J., Yi, Z., Žurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3971, pp. 537–542. Springer, Heidelberg (2006)
13. Zhao, H., Lu, B., Wen, Y.M., Wang, K.A.: On effective decomposition of training data sets for min-max modular classifier. In: Proceeding of 12th International Conference on Neural Information, Taipei, Taiwan, pp. 343–348 (2005)

Adaptive Decision Making in Ant Colony System by Reinforcement Learning*

Keiji Kamei¹ and Masumi Ishikawa²

¹ Nishinippon Institute of Technology, 1-11 Aratsu Kanda,
Miyakogun, Fukuoka, 800-0394, Japan

² Kyushu Institute of Technology, 1-1 Sensui,
Tobata, Kitakyushu 804-8550, Japan
kamei@nishitech.ac.jp
ishikawa-m@office.kyutech.ac.jp
<http://www3.nishitech.ac.jp/>

Abstract. Ant Colony System is a viable method for routing problems such as TSP, because it provides a dynamic parallel discrete search algorithm. Ants in the conventional ACS are unable to learn as they are. In the present paper, we propose to combine ACS with reinforcement learning to make decision adaptively. We succeeded in making decision adaptively in the Ant Colony system and in improving the performance of exploration.

Keywords: Ant Colony System, Reinforcement Learning, Traveling Salesman Problem, Adaptive Decision Making.

1 Introduction

Routing problems have attracted wide attention recently, because they frequently appear in large network systems such as internet, automobile navigation systems, and assembly of electronic devices. One of the most famous problems in routing is the Traveling Salesman Problem(TSP). Given a set of nodes and distances between nodes, TSP seeks to find the shortest path which visits each node exactly once. TSP is a well known combinatorial optimization problem with NP-hard, and is frequently used as benchmark for search algorithms.

The Ant Colony System(ACS)[1] is a viable method for routing problems such as TSP, because it provides dynamic parallel discrete search algorithms with multiple agents. Agents in other parallel discrete search algorithms such as genetic algorithms(GAs), particle swarm optimization(PSO), and Multi-Agent System(MAS) do not communicate with each other. In contrast, agents in ACS communicate with each other with pheromone observed in natural ants. Ants secrete pheromone during exploration, hence the pheromone builds up on the

* In 2007 he obtained his PhD at the Department of Brain Science and Engineering, Kyushu Institute of Technology. Since 2007, he has been a lecturer, Nishinippon Institute of Technology.

traversed links between nodes. During exploration, an ant selects succeeding link probabilistically based on the intensity of the pheromone. From this characteristic, an ant is able to share contextual information of exploration with other ants. Because of this, ACS algorithms show better performance than other routing algorithms such as GA, PSO, and MAS.

However, artificial ants in the conventional Ant Colony System use only one kind of pheromone for their communication, and they do not communicate with each other. On the other hand, natural ants use haptic information, and cooperate by communication among them. Moreover, it is conjectured that natural ants learn the environment with their micro brain.

Dorigo et al. proposed to merge Ant Colony System and Reinforcement Learning[7]. The Q-learning, which is one of the most popular reinforcement learning methods, uses the temporal difference learning for all pairs of the state of the environment and action of an agent. Their proposal, Ant-Q[2][3], merged ACS with the Q-learning to update the amount of pheromone on the succeeding link.

Erik J. Dries et al. proposed to improve the performance of the Ant-Q by hierarchical reinforcement learning(HRL)[4]. The HRL aims to split Markov Decision Process(MDP) into smaller MDPs, each of which can be solved easily. Their proposal splits the set of nodes for TSP into sub-domains; HRL splits and ACS solves smaller TSP.

Arita et al. proposed to combine a genetic algorithm and the Ant Colony System[5][6]. Their proposal assumes that ants use 2 kinds of pheromone, i.e., one shows the distance between the ant and the nest, and the other shows the distance between the ant and the food. An ant has a micro brain which is assumed to be constructed by a three layer feed-forward neural network. For the ants to make use of 2 kinds of pheromone, synaptic connections in the neural network are iteratively adjusted by a GA.

As far as we know, proposals which combine Ant Colony System and other methods are improve the performance of ACS, but artificial ants do not learn. We propose to introduce Reinforcement Learning into the ACS with one kind of pheromone. Each goal has a parameter indicating the degree to which an ant favors the goal. Our proposal aims to realize that artificial ants use not only the contextual information by pheromone but also goal oriented information by state value function from TD(0)-learning, and that ants decide one of the goals adaptively. This proposal expands the conventional ACS inspired by natural life by introducing reinforcement learning. An automobile navigation system will be able to suggest a destination adaptively by our method.

Section 2 presents the details of Ant Colony System. Section 3 explains our proposal in depth. Section 4 presents experiments and the results. Section 5 concludes this paper.

2 Ant Colony System

It is said that natural ants communicate each other through pheromone which is emitted by them to optimize the path between their nest and food. The Ant

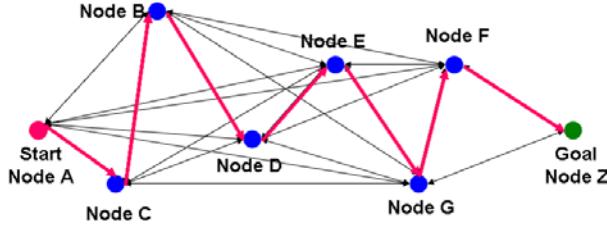


Fig. 1. The example of path diagram

Colony System(ACS) mimics such the pheromone communication system. The pheromone is accumulated if an ant through a path, and is transpired time goes by. Then, the pheromone that is secreted by few ants will not have accumulated after long-sustained. The ACS is dynamic parallel discrete search because the artificial ants choose a going path probabilistically based on a intensity of the pheromone.

Consider the path diagram of Fig. II here. The start A is a nest of ants and the goal Z is a food. Each ants depart the start A simultaneously to search foods for goal. An ant chooses a next node randomly in the initial search stage for a goal. One ant might go to node B, and another might go to node C. Ants never choose visited node, and iteratively choose next city until to reach a goal. An ant returns to the nest in reverse way when it reaches to a goal. An ant secrete pheromone between previous node and posterior node while an ant returns to nest. The intensity of pheromone which is secreted depends on the distance between nodes. The secretion volume of pheromone is as follows.

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (1)$$

where τ_{ij} is the volume of pheromone between node i and j , $\Delta\tau_{ij}^k(t)$ is secretion volume by ant k , m is the number of total ants, ρ is a transpiration rate. $\Delta\tau_{ij}^k(t)$ is,

$$\Delta\tau_{ij}^k(t) = \begin{cases} 1/L^k(t) & \text{if } (i, j) \in T^k(t) \\ 0 & \text{if } (i, j) \notin T^k(t) \end{cases} \quad (2)$$

where T^k is a set of visited link, L^k is the distance of visited link. These equations show that pheromone on link between node i and j accumulate at a rate of ρ at time t , and the pheromone will be added from sum of secretion by ant which visited link between node i and j .

In later search stage, ants choose the next node probabilistically based on the intensity of pheromone. The probability explained as,

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad \forall j \in N_i^k \quad (3)$$

where τ_{ij} is volume of pheromone between node i and j , η_{ij} is heuristic information, which is defined as reciprocal of distance between node i and j , α and β

is constant value of importance weight for pheromone or heuristic information. For example, ants will place more emphasis on pheromone and less on heuristic information if the α is greater than the β .

The ACS repeats these procedure several times for getting the optimal path such as bold-line in Fig. 2.

3 Ant Colony System with Reinforcement Learning

Pheromone in conventional Ant Colony System is updated by equation (2) only, and is updated offline. The Ant-Q algorithm proposed to merge reinforcement learning to improve performance of ACS. The pheromone in Ant-Q is updated by Q-learning like formula, and is updated online. It is possible to say that the Ant-Q is one of the extension of conventional ACS because it changed only the formula and procedure for pheromone update.

Our proposal, in contrast to Ant-Q, expands ACS inspired by natural life. An assumption adopted here is that artificial ants have a micro brain and sense organ. Ants are able to recognize a state by their eyes and to communicate each other via haptic sense.

There have been diverse algorithm in reinforcement learning. Temporal difference (TD) learning is the most-used reinforcement learning algorithm. TD(0)-learning is the basic learning method because it estimates the optimal values as a function of states.

All ants have state value function which like TD(0)-learning in their brain. Fig. 2 illustrates the use of value function. The value function is defined between nodes. Ants exchange the information of value function via haptic sense. From this ability, all of ants have same value function; it seems to share the value function. The value function is defined two kinds on one link because ACS is as directed graph. For example, a value function between node i and j is defined as $V_{ij}(s)$ and $V_{ji}(s)$. An ant uses $V_{ij}(s)$ if the ant visited node i earlier than j.

Our proposal perform online step-by-step the value function, $V_{ij}(s)$, updates by,

$$V_{ij}(s) \leftarrow V_{ij}(s) + A [r + \gamma V_{jk}(s') - V_{ij}(s)] \quad (4)$$

where s is a state, s' is a next state, A is learning rate, r is a reward, γ is a discount rate. This value function will be used for update pheromone after all ants to reach goals, thus pheromone in our proposal is offline update. The formula of pheromone update is expressed as,

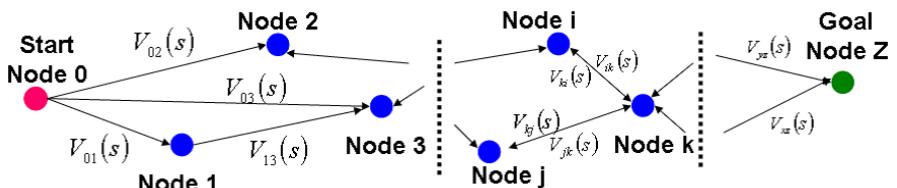


Fig. 2. The example of definition of the value function

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q_g V_{ij}(s) / L^k(t) & \text{if } (i, j) \in T^k(t) \\ 0 & \text{if } (i, j) \notin T^k(t) \end{cases} \quad (5)$$

where Q_g is a quality for goals.

In the beginning stage of process, ants chose next node in a similar way to conventional ACS, because pheromone which is between nodes have a little intensity. Therefore the large number of ants go the nearest goal which has low quality. After several exploration, ants learn the state value between nodes from the quality of goals and the goal qualities influence the secretion volume. Thereby, ants will be able to chose the way of high quality goal.

By using our proposal that is ACS with reinforcement learning, it is possible to achieve a path to the high quality goal which is far from a nest.

4 Experiments

Table II shows a standard experimental condition, and Fig. 3 illustrates the layout of nodes. That is predetermined by a coordination table which is randomly generated except nodes for start and goal. The coordinate of start node is (100, 100), goal 1 is (25, 25), goal 2 is (300, 40), goal 3 is (200, 200). The goal 1 and the goal 3 is near than the goal 2. The goal 2, however, has higher quality than others as illustrated in Fig. 3.

Table 1. A standard condition for experiments. “#nodes” stands for the number of nodes, “#goals” stands for the number of goals, “#ants” stands for the number of ants.

Condition for a part of Ant Colony System								
# nodes	# goals	# ants	α	β	ρ	Q_{g1}	Q_{g2}	Q_{g3}
246	3	25	1.0	5.0	0.5	1.0	80.0	1.0
Condition for a part of Reinforcement Learning								
A	γ	Reward for goal	Reward for moving					
0.2	0.9	1.0	0.0					

4.1 Results of Experiment 1: Standard Conditions

In this section, we presents the results of experiments for a standard conditions as Table II. Fig. 4 illustrates the transition of the path to the goals.

Fig. 4(a) shows that the ants randomly choose among nodes. The average distance at the first exploration in Table 2 is too long, and the most ants goes the goal 1 which has low quality because the goal 1 is the nearest from start. The paths in Fig. 4(b) look like semi-sophisticated than first exploration. The average distance at the second exploration is very short by comparing with the first, and the ants head for all goals discretely. The path to the goal is more better and shorter than the second exploration by Fig. 4(c) and Table 2. Fig. 4(e) illustrates that the paths near the goals are sophisticated, but the paths in upper left on

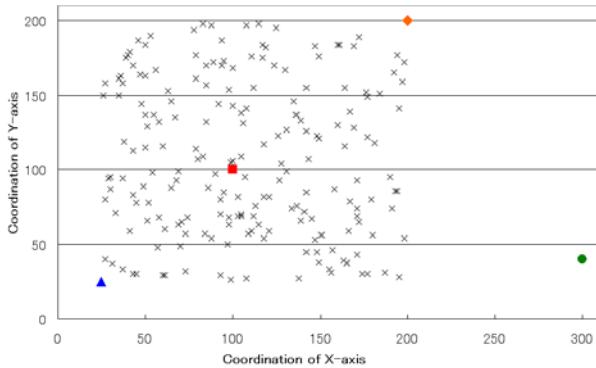


Fig. 3. Layout of nodes. The square node is start, the triangle node is the goal 1, the circular node is the goal 2, the rhomboid-shaped node is the goal 3.

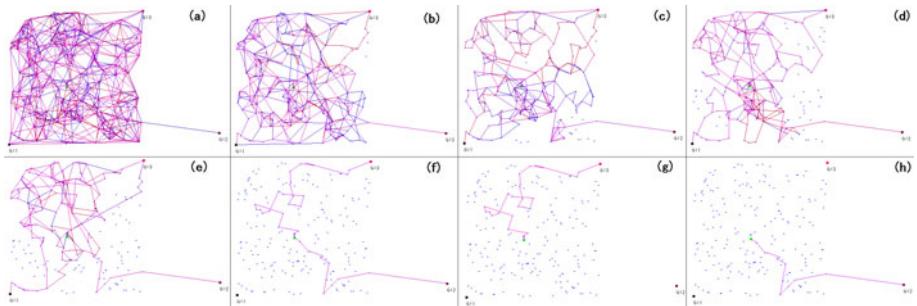


Fig. 4. The transition of the path to the goal. (a), (b), (c), (d), (e), (f), (g) and (h) stand for first, second, third, fourth, fifth, seventh and fourteenth exploration, respectively. “G1”, “G2” and “G3” in each stand for goal 1, goal 2 and goal 3, respectively.

Table 2. The number of ants to the goals and the average distance of the path to the goals. “#ants” stands for the number of ants to reach the goals. “Avg. dist.” stands for the average distance to reach the goals. “NA” stands for not assigned.

		The number of exploration								
		1	2	3	4	5	6	7		14
Goal 1	#ants	17	7	10	5	3	0	0		0
	Avg. dist.	1618.73	695.75	442.46	398.03	739.23	NA	NA		NA
Goal 2	#ants	1	9	9	15	7	1	0		25
	Avg. dist.	2289.23	578.99	396.22	460.55	461.99	274.57	NA		274.57
Goal 3	#ants	7	9	6	5	15	24	25		0
	Avg. dist.	1972.08	672.09	389.33	396.39	429.34	400.82	400.82		NA

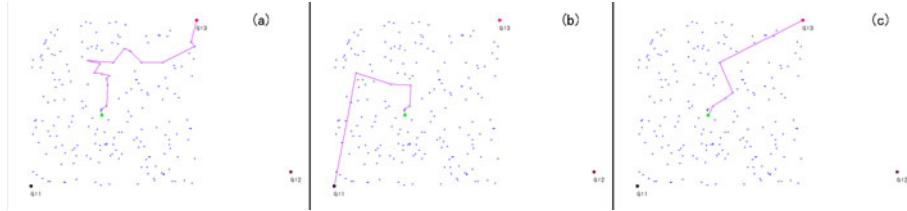


Fig. 5. The path to the goals in experiment 2. (a), (b), (c) stand for the case A, the case B, the case C, respectively.

the environment are not enough optimized. Since then, the average distance in Table 2 is longer than previous exploration. Although the explorations that the ants head for low quality goal such as Fig. 4(f) and (g), the all of ants ultimately reach to goal 2 which has the highest goal as Fig. 4(h).

4.2 Results of Experiment 2: Alteration for Qualities for Goals

In this section, we presents the results of alteration for quality for goals. We conduct the three cases of experiments. The case A is that the three goals have same quality, i.e., all of quality is 1.0, and the case B is that the quality for goal 1 is very bigger than others, i.e., quality for goal 1 is 1000.0, and the case C is

Table 3. The results of alteration for qualities for goals

		The number of exploration						
		1	2	3	4	5	6	11
A	Goal 1 $Q_{g1}: 1.0$	#ants	17	13	16	8	4	0
		Avg. dist.	1618.73	841.13	474.68	234.33	250.56	NA
	Goal 2 $Q_{g2}: 1.0$	#ants	1	1	1	1	0	0
		Avg. dist.	2289.23	850.96	388.4	300.18	NA	NA
B	Goal 1 $Q_{g1}: 1000.0$	#ants	7	11	8	16	21	25
		Avg. dist.	1972.08	816.18	483.75	466.68	287.91	245.19
	Goal 2 $Q_{g2}: 80.0$	#ants	1	2	0	0	0	0
		Avg. dist.	2289.23	395.51	NA	NA	NA	NA
C	Goal 1 $Q_{g1}: 1.0$	#ants	7	5	2	0	0	0
		Avg. dist.	1972.08	387.08	234.72	NA	NA	NA
	Goal 2 $Q_{g2}: 80.0$	#ants	17	6	8	0	0	0
		Avg. dist.	1618.73	31.7	335.24	NA	NA	NA
	Goal 3 $Q_{g3}: 1000.0$	#ants	1	4	4	1	0	0
		Avg. dist.	2289.23	658.58	387.19	254.4	NA	NA
		Avg. dist.	1972.08	552.75	287.89	181.05	169.33	169.33
		Avg. dist.	1972.08	552.75	287.89	181.05	169.33	169.33

that quality for goal 3 is bigger than others, quality for goal 3 is 1000.0. Table 3 and Fig. 5 shows the results of experiments.

In case A, a multitude of ants head for goal 1 in beginning stage. The most of ants, however, head for the goal 3 after fourth exploration. The all of ants eventually reach to the goal 3. It is possible to think that the path to the goal 3 is simple because the number of nodes near the goal 3 is fewer than the others. Compare with the other cases, this case needs a lot of exploration because it takes much time so that the average distance to reach to reach goals become short.

In case B, most of ants head for the goal 1 form initial stage, and the average distance to reach the goal 1 decrease drastically whenever the ants explore. All of ants reach to the goal 1 after fourth exploration, and the average distance to reach the goal 1 stay fairly since then

The case C is same as the case B. The most of ants head for goal 3 which has the highest quality from the beginning, and all of ants reach the goal 3 after fifth exploration.

Table 3 and Fig. 5 shows that the case C achieve a good path, which is without big detour.

5 Conclusions

In this paper, we proposed to introduce reinforcement learning into the Ant Colony System inspired by natural ants. This proposal realizes that artificial ants can use not only the contextual information by pheromone but also goal oriented information by state value function from reinforcement learning. A key idea is that ants have a state value function representing spatial information of goals. Owing to this ability, ants can have spatio-temporal information. In our experiments, ants explore to find 3 goals. Firstly, we carried out an experiment using a standard condition to demonstrate the effectiveness of our idea. The results of this experiment is that most of ants head for goal 1 and goal 3, because they are nearer the start than goal 2 in the beginning stage. All ants, however, ultimately reach goal 2, because it is more favored than other goals.

Secondly, we changed the the degree to which an ant favors goals to demonstrate how it affects exploration. The results shows that it affects decision making for goal selection and the performance of exploration.

We succeeded in making a decision adaptively in Ant Colony system, and furthermore in improving the performance of exploration. To clarify the relation between ACS and reinforcement learning is left for further study.

Acknowledgment

This research was supported by Grant-in-Aid for Special Research Equipments from Nishinippon Institute of Technology, Japan.

References

1. Dorigo, M., Caro, G.D.: Ant Algorithm for Discrete Optimization. *Artificial Life* 5(2), 137–172 (1999)
2. Dorigo, M., Gambardella, L.M.: A study of some properties of Ant-Q. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 656–665. Springer, Heidelberg (1996)
3. Gambardella, L.M., Dorigo, M.: Ant-Q: A Reinforcement Learning approach to the traveling salesman problem. In: Proc. of 12th ICML, pp. 252–260 (1995)
4. Dries, E.J., Peterson, G.L.: Scaling ant colony optimization with hierarchical reinforcement learning partitioning. In: GECCO 2008, pp. 25–32 (2008)
5. Arita, T., Koyama, Y.: Evolution of Linguistic Diversity in Simple Communication System. *Artificial Life* 4(1), 109–124 (1998)
6. Nakamichi, Y., Arita, T.: Diversity Control in Ant Colony Optimization. *Artificial Life and Robotics* 7(4), 198–204 (2004)
7. Sutton, R.S., Barto, A.G.: Reinforcement Learning. MIT Press, Cambridge (1998)

A Cooperative Coevolutionary Algorithm for the Composite SaaS Placement Problem in the Cloud

Zeratul Izzah Mohd Yusoh and Maolin Tang

Queensland University Technology,
2 George Street, Brisbane, QLD 4001, Australia

Abstract. Cloud computing has become a main medium for Software as a Service (SaaS) hosting as it can provide the scalability a SaaS requires. One of the challenges in hosting the SaaS is the placement process where the placement has to consider SaaS interactions between its components and SaaS interactions with its data components. A previous research has tackled this problem using a classical genetic algorithm (GA) approach. This paper proposes a cooperative coevolutionary algorithm (CCEA) approach. The CCEA has been implemented and evaluated and the result has shown that the CCEA has produced higher quality solutions compared to the GA.

1 Introduction

Cloud computing [1] is an emerging computing paradigm in which applications, data and IT resources are provided as a service to users over the Internet. One kind of services that can be offered through the Cloud is SaaS [1]. With increasing demands for SaaS each year, Cloud infrastructure is by far the best option for supporting SaaS as it provides scalability that is much needed by SaaS [2].

A SaaS deployed in a Cloud is usually composed of several components and data, where each of the components represents a business function of the SaaS that is being delivered [3]. For SaaS placement in the Cloud, the problem relates to how a composite SaaS should be placed in a Cloud by the Cloud's provider such that its performance is optimal based on its estimated execution time. The challenges in the SaaS placement process rely on several factors, including SaaS interactions between its components and SaaS interactions with data components. The latter plays a significant role in SaaS execution time as the data are located at the Cloud servers instead of local machines. Existing SaaS placement methods were not designed for the Cloud. The methods mostly focus on the resource consumption by the components and are not concerned with the placement of the component's data [4,5].

This research will address this problem by proposing an algorithm which considers not only the placement of the software components of a SaaS, but the placement of data of the SaaS as well. In our previous research of the problem, we have proposed a classical GA (GA) [6]. In this paper, we present a cooperative coevolutionary algorithm (CCEA) for the problem.

The remaining paper is organized as follows. Section 2 describes and formulates the SaaS deployment problem. Section 3 elaborates the CCEA. Section 4 is about the evaluation that has been carried out. Finally, the concluding remark is presented in Section 5.

2 SaaS Placement Problem (SPP)

There are three main roles in a SaaS scenario of a Cloud. They are the SaaS vendor, Cloud infrastructure provider and SaaS customer. The SaaS vendor submits the composite SaaS with its data to the Cloud provider, where the Cloud provider is responsible for hosting and placing the SaaS in the Cloud based on an agreement between the SaaS vendor and the Cloud provider. The placement of the composite SaaS has to be done strategically, as the SaaS components and associated data are dependant upon each other. This is so called *SaaS placement problem*, or SPP. The objective of the problem is to decide where each of the SaaS components and data chunks should be placed such that some requirements are satisfied and the SaaS performance is optimal based on its estimated execution time. The decision of SPP will be based on estimated SaaS execution time on a set of selected Cloud servers. The following sections describe SPP formulation and numerical attributes applied in the calculation process.

2.1 SPP Problem Formulation

There are three main inputs of the problem:

- SaaS Modelling: The SaaS formulation is made general enough to represent a SaaS. It includes SaaS components' requirements such as its minimum processing capacity, memory requirement, storage requirement and data access requirement. The SaaS workflows are also formulated as an input.
- Cloud Resource Modelling: The Cloud consists of computing servers and storage servers where each server has its own attributes that are relevant to the SaaS requirements. Among the attributes are processing capacity, storage capacity as well as memory capacity.
- Cloud Network Modelling: The Cloud network is represented by an undirected graph, $G = \{V, E\}$. $V = (P, D)$ is the sets of vertices including computation servers and storage servers, $e \in E$ is the set of undirected edges connecting the vertices, if and only if there exists a physical link transmitting information from v_i to v_j , where $v_i, v_j \in V$. $B_{v_i, v_j} : E \rightarrow R^+$ and $L_{v_i, v_j} : E \rightarrow R^+$ are bandwidth and latency functions of the link from to respectively.

Detail formulations for SaaS and Cloud resource modelling can be found in [6].

2.2 SPP Attributes

To determine a suitable server for a component, four numerical attributes will be used. These attributes define the total estimated execution time (in seconds) of the SaaS that is being deployed. Followings are the descriptions of the attributes.

- Estimated Data Transfer Time (EDTT): The estimated time taken for transferring data between storage servers and computing servers. Given a current placement for a component sc_i , EDTT is calculated based on AD_{sc_i} , the total bytes of amount of read/write task of sc_i , B_{p_i, d_j} , bandwidth of the links involved, and L_{p_i, d_j} , the latency that may occur in those links.

$$EDTT(sc_k) = \sum_{v_i, v_j \in E} \frac{AD_{sc_k} \times 8}{B_{p_i, d_j}} + L_{p_i, d_j} \quad (1)$$

- Estimated Processing Time (EPT): The EPT for a component in a selected computing server. It is based on the task size of a component, TS_{sc_i} , the processing capability of the selected server PC_{p_i} , the value of EDTT (if there is any), and the transfer time of the output of its predecessor component, $TR_{sc_{k-1}}$. If a component accesses more than one storage servers or has more than one predecessor, only the maximum value of these attributes will be considered.

$$EPT(sc_k) = \frac{TS_{sc_k}}{PC_{p_k}} + max(EDTT) + max(TR_{sc_{k-1}}) \quad (2)$$

- Estimated Execution Time (EET): The EET of a path in a workflow. It is based on the sum of EPT of each component in a path. The EET is defined as:

$$EET(path_k) = \sum_{sc \in path_k} EPT(sc_i) \quad (3)$$

- Estimated Total Execution Time (ETET): ETET for the whole SaaS that is being placed in the Cloud. From Eq. 3, the EET for all paths in the SaaS workflows is obtained. These values are used to determine the critical path of a workflow if the workflow has more than one path. Details explanation on the critical path determination is presented in [6]. Then, the ETET is defined by the sum of the EET of the critical path of each workflow multiplied by its weighing as shown in Eq. 4.

$$ETET(SaaS) = \sum_{i=1}^q EET(criticalpath) \times WF_{wf_i} \quad (4)$$

3 Cooperative Coevolutionary Algorithm

From the computational point of view, SPP is a large-scale and complex combinatorial optimization problem with constraints, and an evolutionary approach would be suitable for it. We have developed a classical GA in a previous research [6]. Aiming at further improve the quality of solutions, this section proposes a new coevolutionary algorithm for SPP.

The coevolutionary algorithm is an evolutionary algorithm inspired by the coevolution in biological systems. Coevolution is defined as an evolution of one

species as a result of its responses to characteristics of another species in a common ecosystem [7]. Coevolutionary algorithms are classified into two types: cooperative coevolutionary algorithms and competitive coevolutionary algorithms. Our coevolutionary algorithm is a cooperative coevolutionary algorithm.

In a cooperative coevolutionary algorithm, the fitness of a species (or it is usually referred to as subpopulation) is calculated on how well it ‘cooperates’ with the other subpopulation in order to produce a good solution. The population in a cooperative coevolutionary algorithm is divided into several subpopulations. The subpopulations evolve cooperatively. Among existing work is the cooperative coevolutionary algorithm for function optimization [8]. In that paper, an optimization problem with N variables is decomposed naturally into N subpopulations. An individual is rewarded when it works well with individuals from the other subpopulations and is punished when they perform poorly together. The cooperative coevolutionary algorithms have also been applied to other areas, such as artificial neural network [9].

3.1 Cooperative Coevolutionary Algorithm for SPP

Based on the characteristics of SPP, we decomposed the problem into two interacting subproblems: 1) the placement of SaaS components in computation servers, and 2) the placement of SaaS’s data chunks in storage servers. The subproblems depends on each other. Each of the subproblems is solved using a classical genetic algorithm, which is similar to the classical genetic algorithm for SPP that we developed previously. Since the two subproblems are dependent on each other, the two subpopulations evolve cooperatively, rather than independently.

Encoding: The first subpopulation contains n genes, each of which corresponds to a software component, representing the computation server where the software component would be placed in the placement plan, where n is the total number of software components in the SaaS. The second subpopulation is encoded in similar fashion for data chunks and its corresponding storage server. Each gene in the chromosome is represented in a triple $\langle C, R, S \rangle$, where C , R and S are the IDs for continent, region and server, respectively.

Genetic Operators: The crossover operation is a classical one point crossover between the segments of genes in a chromosome. The crossover operation combines segments from two selected parents and produces two children. The top two fittest among them are selected into the next generation. The mutation operator applied is a random mutation operator.

Fitness Evaluation: In order to calculate the fitness of an individual, a partner from the other subpopulation is selected and combined with the individual to form a complete SaaS placement solution. This solution then is evaluated using the fitness function in Eq. 5. The partner selection is based on the individual’s fitness from the previous generation of the algorithm. The fittest 50% individuals

from each subpopulation are selected, and paired up randomly. A fitter individual represents a better placement solution.

$$Fitness(X) = \frac{ETET(X)}{ETET^*(P)} \quad (5)$$

where $ETET^*(P)$ is the maximum value of *Estimated Total Execution Time* in a population, P .

The Algorithm Description:

1. randomly generate an initial subpopulation for the SaaS component placement problem.
2. randomly generate an initial subpopulation for the SaaS data chunk placement problem.
3. while the termination condition is not true
 - (a) for each individual in the SaaS component population:
 - i. select a partner from the SaaS data chunk population;
 - ii. calculate the fitness value of the SaaS placement formed by combining the individual and the partner;
 - (b) select individuals from the SaaS component population based on their fitness values for recombination and pair them up;
 - (c) probabilistically apply the crossover operator to generate new individuals;
 - (d) probabilistically select individuals for mutation;
 - (e) use the new individuals to replace the old individuals in the SaaS component population.
 - (f) for each individual in the SaaS data chunk population:
 - i. select a partner from the SaaS component population;
 - ii. calculate the fitness value of the SaaS placement formed by combining the individual and the partner;
 - (g) select individuals from the SaaS data chunk population based on their fitness values for recombination and pair them up;
 - (h) probabilistically apply the crossover operator to generate new individuals;
 - (i) probabilistically select individuals for mutation;
 - (j) use the new individuals to replace the old individuals in the SaaS data chunk population.
4. output the best SaaS placement solution in the history.

4 Evaluation

We have implemented the CCEA in Microsoft Visual Studio C++. The GA was also implemented in the same programming language previously. The evaluation was conducted by empirically comparing the quality of the solutions produced

by the CCEA with the quality of the solutions produced by the GA and by studying the scalability of the CCEA.

In the experiments, the population size for both algorithms were set at 200. Crossover and mutation rates were set at 0.95 and 0.15 respectively. The termination condition was ‘no improvement on the best solution for 100 consecutive generations’. The experiments were conducted on several randomly generated SaaS placement problems. The Cloud was randomly created as well where the attributes of the servers were generated using the models presented in [10]. All the experiments were carried out in a computer with 2.66 GHz Intel Core 2 Duo CPU and 3.23GB RAM.

The first experiment was to compare the quality of the solutions produced by the CCEA with the quality of the solutions produced by the GA. The comparison was based on the ETET of the SaaS that was calculated using Eq. 4. In this experiment, we tested both of the algorithms for a Cloud that contained from 200 to 1400 servers, with an increment of 200. The numbers of SaaS components and data chunks were both fixed at 10. For each of the configurations, one test problem was randomly generated and we applied the two algorithms to solve the randomly generated test problem. Considering the stochastic nature of the algorithms, each of the experiments was repeated 20 times. Fig. 1 illustrates the ETET values of both the CCEA and the GA, and Fig. 2 shows the computation times taken by the two algorithms.

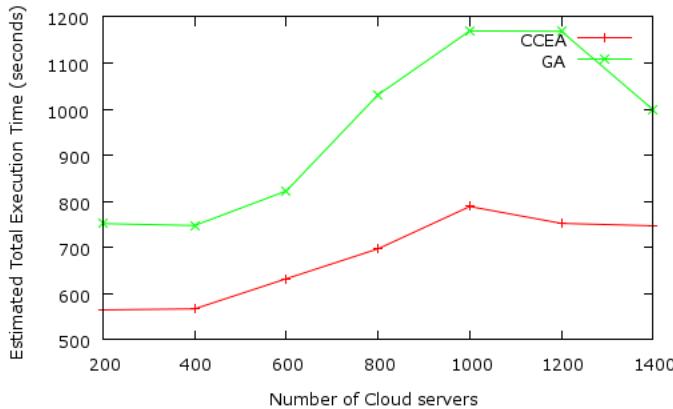


Fig. 1. Comparison of the quality of solutions obtained by the CCEA and the GA

It can be seen from Fig. 1 that the CCEA has a shorter ETET, hence a better placement solution than the GA. It is also shown from the figures that the CCEA has longer computation times for all test problems. However, the longest time it took was still less than eight minutes, which is acceptable considering the placement plan is conducted in an offline mode. It can also be seen from Fig. 2 that the computation time of the CCEA increased close to linearly with the size of the Cloud. The blip in the curve resulted from that the randomly generated

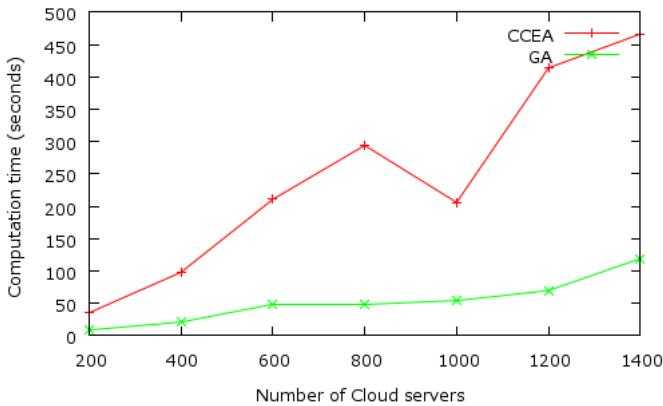


Fig. 2. Comparison of computation time for the CCEA and the GA

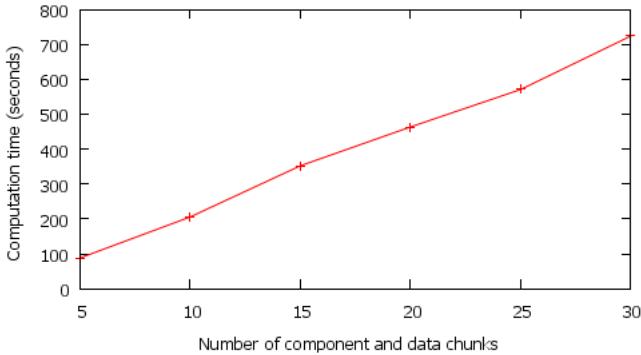


Fig. 3. The effect of the number of components and data chunks on the computation time of the CCEA

test problem for 1000 cloud servers was relatively easy to solve and therefore took less time to converge to an optimal or near-optimal solution.

The second experiment was to further evaluate the scalability of the CCEA by observing the growth trend of its computation time when the number of components and data chunks increases. In this experiment, the numbers of SaaS components and data chunk were set from five to 30 with an increment of five, and the number of servers in the cloud was set at 800. The same server's capacities and network were used in each configuration. Experiments for each configuration were repeated for 20 times. Fig. 3 shows the average computation time on finding a solution. The result shows that the computation time of the CCEA increases close to linearly with the number of components and data chunks.

5 Conclusion

This paper has proposed a cooperative coevolutionary genetic algorithm for the composite SaaS placement problem in the Cloud. Different from the previous approach, this cooperative coevolutionary algorithm decomposes the population into two sub-populations, which evolve cooperatively and iteratively throughout the evolution process. To the best of our knowledge, this is the first cooperative coevolutionary algorithm for the composite SaaS placement problem.

Experimental results have shown that the proposed cooperative coevolutionary algorithm can produce better quality solutions than the classical genetic algorithm for all the tested problems although its computation time is noticeably longer than that of the classical genetic algorithm. The computation time of the cooperative coevolutionary algorithm can be reduced by parallelling the evolution of the two subpopulations, which is our future work on the SaaS deployment problem. We will also try to use other effective cooperative coevolutionary algorithms, such as [11], to tackle the SaaS deployment problem.

References

1. Foster, I., Yong, Z., Raicu, I., Lu, S.: Cloud computing and grid computing 360-degree compared. In: Grid Computing Environment Workshop, pp. 1–10 (2008)
2. Candan, K.S., Li, W.S., Phan, T., Zhou, M.: Frontiers in information and software as a service. In: IEEE 25th International Conference on Data Engineering, pp. 1761–1768 (2009)
3. Laplante, P.A., Jia, Z., Voas, J.: What's in a name? Distinguishing between SaaS and SOA. *IT Professional* 10(3), 46–50 (2008)
4. Karve, A., Kimbrel, T.: Dynamic placement for clustered web applications. In: 15th International Conference on World Wide Web, pp. 595–604. ACM, Scotland (2006)
5. Zhu, X., Santos, C., Beyer, D., et al.: Automated application component placement in data centers using mathematical programming. *International Journal of Network Management* 18(6), 467–483 (2008)
6. Yusoh, Z., Tang, M.: A penalty-based genetic algorithm for the composite SaaS placement problem in the cloud. In: IEEE World Congress on Computational Intelligence, pp. 600–607. IEEE, Spain (2010)
7. Janzen, D.H.: When is it coevolution? *Evolution* 34(3), 611–612 (1980)
8. Potter, M., De Jong, K.: A cooperative coevolutionary approach to function optimization. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 249–257. Springer, Heidelberg (1994)
9. Moriarty, D.E., Mikkulainen, R.: Efficient reinforcement learning through symbiotic evolution. *Machine Learning* 22(1), 11–32 (1996)
10. IBM System Storage, <http://www-07.ibm.com/storage/au/>
11. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. *Information Sciences* 178(15), 2985–2999 (2008)

A Swarm Intelligence Approach to the Quadratic Multiple Knapsack Problem

Shyam Sundar and Alok Singh

Department of Computer and Information Sciences
University of Hyderabad, Hyderabad 500046, India
`{mc08pc17,alokcs}@uohyd.ernet.in`

Abstract. In this paper we present an artificial bee colony (ABC) algorithm to solve the quadratic multiple knapsack problem (QMKP) which can be considered as an extension of two well known knapsack problems viz. multiple knapsack problem and quadratic knapsack problem. In QMKP, profit values are associated not only with individual objects but also with pairs of objects. Profit value associated with a pair of objects is added to the total profit if both objects of the pair belong to the same knapsack. The objective of this problem is to assign each object to at most one knapsack in such a way that the total weight of the objects in each knapsack should not exceed knapsack's capacity and the total profit of all the objects included into the knapsacks is maximized. We have compared our approach with three genetic algorithms and a stochastic hill climber. Computational results show the effectiveness of our approach.

Keywords: Artificial Bee Colony Algorithm, Constrained Optimization, Quadratic Multiple Knapsack Problem, Swarm Intelligence.

1 Introduction

The quadratic multiple knapsack problem (QMKP) is an extension of two well known combinatorial optimization problems viz. multiple knapsack problem (MKP) and quadratic knapsack problem (QKP). Both MKP and QKP are generalization of 0-1 knapsack problem. MKP is the problem of assigning a subset of n objects into K distinct knapsacks such that the total profit of all the objects included into the knapsacks is maximized without violating the capacity constraint of any knapsack. When $K = 1$, MKP reduces to 0-1 knapsack problem. On the other hand, QKP seeks a subset of n objects to be assigned to a single knapsack such that the total profit of all the objects included into the knapsack is maximized without violating the knapsack's capacity. However, in QKP profit values are associated not only with individual objects but also with pairs of objects. Profit value associated with a pair of objects is added to the total profit if both objects of the pair belong to the knapsack. When quadratic property (profit values associated with each pair of objects) of QKP is excluded, QKP reduces to 0-1 knapsack problem.

QMKP combines the features of QKP and MKP. In QMKP, we are given a set of n objects and K knapsacks. Each object i , $i \in \{1, 2, \dots, n\}$ has a profit p_i and a weight w_i , and, each pair of objects (i and j) has a profit p_{ij} , and each knapsack k , $k \in \{1, 2, \dots, K\}$ has a capacity C_k . The profit p_{ij} associated with a pair of objects i and j is added to the total profit if both i and j belong to the same knapsack. The objective of QMKP is to assign each object to at most one knapsack in such a way that the total weight of the objects in each knapsack k should not exceed its capacity C_k and the total profit of all the objects included into the knapsacks is maximized. To define QMKP formally, binary variables x_{ij} are introduced to indicate whether object i is included into the knapsack j ($x_{ij} = 1$) or not ($x_{ij} = 0$). QMKP can be formulated as:

$$\text{Maximize} \quad \sum_{i=1}^n \sum_{k=1}^K x_{ik} p_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^K x_{ik} x_{jk} p_{ij}$$

Since QMKP is an extension of multiple knapsack and quadratic knapsack problems, it is also a \mathcal{NP} -Hard problem [1].

Hiley and Jultstrom [1] were the first to define and study QMKP. They mentioned a real application of QMKP which is as follows: Suppose a manager has to assign his employees for his multiple projects, each with its own budget. The manager knows the salary and productivity of each employee, both individually and in pairs. So, the manager will try to assign a subset of employees to each project at the same time in such a way that the total productivity, without exceeding the budget of any project, is maximized.

Hiley and Jultstrom [1] assumed that all knapsack capacities C_k are equal. They proposed three approaches – a greedy heuristic, a stochastic hill climber and a generational genetic algorithm for this QMKP. Thereafter, Singh and Baghel [2] proposed a grouping genetic algorithm [3], and Sarac and Sipahioglu [4] presented a genetic algorithm for this problem.

QMKP is a grouping problem [3], i.e., a problem that seeks an optimal assignment of objects according to a given cost function into different groups subject to some constraints. In this paper, we propose an artificial bee colony (ABC) algorithm in such a way that grouping property of QMKP is preserved as far as possible while solving QMKP. The ABC algorithm is a recent technique based on the intelligent foraging behaviour of honey bee swarm. This technique was proposed by Karaboga [5]. To our knowledge, this is the first time an ABC algorithm is used to solve a grouping problem. We compare our ABC algorithm with other best performing approaches. Computational results show that our algorithm performs well in comparison to these approaches.

The rest of this paper is organized as follows: Section 2 provides a brief introduction to artificial bee colony algorithm. Section 3 describes in detail our ABC approach for QMKP. Computational results are presented in Section 4, whereas conclusions are outlined in Section 5.

2 Artificial Bee Colony Algorithm

The artificial bee colony algorithm (ABC) is a recent population-based search technique motivated by the intelligent foraging behavior of honey bee swarm. It was proposed by Karaboga [5] and then further extensions are carried out by [5,6,7,8,9]. On the basis of their foraging behavior, real bees are classified into three groups – employed bees, scouts and onlookers. Employed bees are those bees that are currently exploiting the food sources. All employed bees are responsible for bringing loads of nectar from their food sources to the hive sharing information about their food sources with onlookers. Onlookers are those bees which are waiting in the hive for employed bees to share information about their food sources. Scouts are those bees which are searching for a new food source in the vicinity of the hive. The employed bees share the information about their food sources with onlookers by dancing in a common area. The nature and duration of the dance of an employed bee depends on the quality of the food source currently being exploited by her. Onlookers watch numerous dances of employed bees before choosing a food source. The probability of choosing a food source is directly proportional to its quality. Therefore, the good food sources attract more onlookers than the poor ones. Whenever a bee, whether it is scout or onlooker, finds a food source it becomes employed. Whenever, a food source is fully exploited, the associated employed bee abandons it and becomes scout. As soon as, scout searches a new food source in the vicinity of its hive, it again becomes employed.

In ABC model, the colony of artificial bees are also categorized into same three groups – employed bees, onlookers and scouts. Employed bees are the first half of the colony, while onlookers are the latter half. Each food source represents a candidate solution to the problem and the nectar amount of a food source corresponds to the the quality (fitness) of the solution being represented by that food source.

The ABC algorithm is an iterative search technique. It starts by associating each employed bee with a randomly generated food source (solution). Then, during each iteration, each employed bee determines a new food source in the neighborhood of its currently associated food source and computes the nectar amount of this new food source. If the nectar amount of this new food source is higher than that of its currently associated food source, then this employed bee moves to the new food source abandoning the old one, otherwise it continues with the old one. After completion of this process by all employed bees, they start sharing information about their food sources with onlookers. Onlookers choose the food sources with probability proportional to their nectar amounts. Once all onlookers select their food sources, each of them determines a new food source in the neighborhood of its selected food source in a manner similar to employed bee. The nectar amount of each new food source is computed. Among all the neighboring food sources computed by onlookers associated with a particular food source i and food source i itself, best food source will be the new location of food source i . If the solution i does not improve for a predetermined number of iterations (called *limit* parameter), then the solution i is deemed to

be completely exploited and its associated employed bee abandons it to become scout. This scout is again made employed by associating it with a new randomly generated food source (solution). When the new locations of all food sources are determined, then the next iteration of the ABC algorithm begins. This process is repeated until termination criterion is satisfied.

A good survey on the ABC and other algorithms simulating bee swarm intelligence can be found in [10].

3 The ABC Algorithm for QMKP

Karaboga's original ABC algorithm was designed for multivariable and multi-modal continuous function optimization. In order to solve QMKP which is a grouping discrete optimization problem, we propose a new version of the basic ABC algorithm that solves QMKP efficiently. We have incorporated group preserving property into our ABC algorithm for solving QMKP. The resulting ABC algorithm is hybridized with a perturbation procedure and a local search. Our ABC algorithm for QMKP is described below:

3.1 Initialization

Each initial employed bee solution is generated in a manner similar to the procedure of Singh and Baghel [2]. Initially, all knapsacks are empty. The procedure selects first object randomly from the set of unselected objects and assigns it to the first knapsack. Then iteratively one of unselected objects, which can be assigned to this knapsack without violating its capacity constraint and has highest relative value density [1] with respect to the objects already present in this knapsack, is added to this knapsack. This process is repeated until it is not possible to add any more unselected object to this knapsack. Same procedure is applied to the next knapsack. This whole process is repeated until all knapsacks have been filled.

3.2 Probability of Selecting a Food Source

In our ABC algorithm, onlooker uses binary tournament selection method for selecting a food source. In binary tournament selection method, two different food sources are selected randomly from the food sources associated with employed bees. With probability b_t , the food source containing higher nectar amount among the two food sources is selected otherwise lower one is selected.

3.3 Determination of a Neighboring Food Source

In order to determine a new food source in the neighborhood of a food source say i , a copy i^c of the food source i is created. After that knapsack replacement procedure is followed with probability P_{nbr} , otherwise perturbation procedure is followed.

Knapsack Replacement Procedure. In this procedure, a food source (solution) j , which is different from the food source (solution) i is selected with the help of binary tournament selection. A knapsack k_{ic} , which is randomly selected from among all knapsacks of the solution i^c , is replaced by a knapsack k_j , which is also selected randomly from among all knapsacks of the solution j . This process is repeated $round(K/3)$ times. In the subsequent iterations, the solution j and the knapsack k_{ic} of the solution i^c are forbidden for selection. After all iterations are over, we check for those objects that occur more than once in different knapsacks of solution i^c due to knapsack replacement procedure. To make the solution feasible, all such objects are removed first from those knapsacks of solution i^c , which have not participated in the knapsack replacement procedure, and then from other knapsacks which have participated in the knapsack replacement procedure. This idea is used to preserve grouping information as far as possible. After that this procedure iteratively tries to assign as many unselected objects as possible into the knapsacks of solution i^c without violating the capacity constraints. This procedure checks all knapsack one-by-one and iteratively tries to add as many unselected objects as possible. During each iteration, the unselected object, which can be assigned and has highest relative value density \boxed{II} with respect to the objects already present in this knapsack, is added to this knapsack.

Perturbation Procedure. The perturbation strategy helps in diversification of the solutions and also prevents the algorithm from converging prematurely. In this strategy, one objected is removed randomly from each knapsack and then it iteratively tries to assign as many unselected objects as possible into the knapsacks of solution i^c without violating the capacity constraints. This procedure checks all knapsack one-by-one and iteratively tries to add as many unselected objects as possible. During each iteration, the unselected object, which can be assigned and has highest relative value density \boxed{II} with respect to the objects already present in this knapsack, is added to this knapsack.

3.4 Local Search

To ameliorate the solution quality, we have implemented a first improvement local search. The local search considers each knapsack one-by-one in a circular manner and exchanges an unassigned object with an object in the knapsack if such an exchange can improve the solution without violating the knapsack's capacity. As soon as a profitable exchange is found, the local search performs that exchange. Thereafter, the local search considers the next knapsack. If no such exchange is found, then anyway local search considers the next knapsack. The local search is used repeatedly until it is not possible to further improve the profit of any knapsack. To speed up the local search without significantly deteriorating the solution quality, the local search is used when difference between fitness of the global best solution and fitness of the current solution before local search is less than P_{ls} .

3.5 Other Features

If a food source (solution) does not improve for a predetermined number of iterations $limit$, then an employed bee associated with the food source abandons it and becomes a scout. This scout will generate a new food source randomly. After generating a food source, this scout again becomes employed. Since, the parameter $limit$ plays an important role in the ABC algorithm as it controls both exploration and exploitation. A large value of limit parameter favors exploitation over exploration whereas reverse is true for small.

4 Computational Results

Our ABC algorithm for QMKP has been implemented in C and executed on a Linux based 3.0 GHz Core 2 duo system with 2 GB RAM. In all our computational experiments with our ABC algorithm, we have used a population of 30 bees, 10 of these bees are employed and the remaining 20 are onlookers. We have set $limit = 50$, $b_t = 0.95$, $P_{nbr} = 0.5$, $P_{ls} = C_k * h$ (where h is 5 for $K = 3$, h is 10 for $K = 5$ and h is 20 for $K = 10$), and total number of generations = 300. All these parameter values are chosen empirically. We have experimented our ABC algorithm on same 60 instances as used in [12]. These instances have three significant features - the density d (proportion of non-zero p_{ij}), number of objects n , and the number of knapsacks K . Density d of the instance shows

Table 1. Results of HJ-SHC, HJ-GGA, SS-GA, SB-SSGGA and ABC on the QMKP instances with $d = 0.25$

Instance n K Num C	HJ-SHC			HJ-GGA			SS-GA			SB-SSGGA			ABC			
	Best	Avg	SD	Best	Avg	SD	Best	Avg	SD	Best	Avg	SD	Best	Avg	SD	Time
100 3 1 688	28144	27635	294	28665	27904	339	28807	28514.40		28798	28485	195	29139	28753.00	209.68	3.80
100 3 2 738	29915	26222	377	28059	27044	421	28456	28225.20		28036	27507	247	28443	28004.00	285.91	3.69
100 3 3 663	25945	25193	365	26780	25991	344	26754	26573.80		26936	26411	283	26901	26585.33	192.90	2.87
100 3 4 804	27109	26127	431	28199	27265	497	28383	28035.40		28418	27473	483	28563	28109.03	342.19	3.74
100 3 5 723	26288	25617	297	27550	26683	397	27582	26243.00		27617	26971	295	27849	27073.67	274.77	3.57
100 5 1 413	21584	20911	289	21914	21315	316	22039	21734.60		22038	21662	191	22390	22117.12	141.17	2.70
100 5 2 442	20934	19768	322	21216	20472	326	21249	20723.80		21459	21046	199	21584	21224.03	188.87	2.93
100 5 3 398	19454	18765	255	20243	19763	296	20862	20444.40		21012	20279	495	21093	20771.12	215.23	2.40
100 5 4 482	20173	19730	279	21698	20923	291	21601	21417.00		21987	21344	409	22178	21767.50	194.69	3.26
100 5 5 434	19932	18843	266	20808	20248	259	20928	16779.00		21057	20304	370	21301	20875.47	199.51	3.18
100 10 1 206	15232	14737	240	13521	12499	419	15778	15505.20		15663	15201	177	15953	15573.65	170.84	1.42
100 10 2 221	14210	13684	243	12859	12019	299	14835	14601.20		15002	14654	207	15487	14896.35	192.44	1.84
100 10 3 199	13334	12918	196	11790	11245	278	14348	14136.20		14231	13716	220	14339	14027.83	191.24	1.58
100 10 4 241	14321	13867	225	13316	12593	333	15495	15178.80		15979	15310	351	15807	15397.00	244.28	2.10
100 10 5 217	13405	12929	210	11909	11389	269	14770	11665.80		14510	14018	225	14719	14376.80	177.96	1.82
200 3 1 1381	99232	98169	548	97469	95497	991	99853	99216.00		99573	99286	289	100662	100103.02	283.79	23.99
200 3 2 1246	106730	105857	469	106162	100521	3242	104277	101179.00		107475	107036	210	107958	107545.20	240.77	18.61
200 3 3 1335	103529	102475	505	101291	97157	2099	—	—		103607	102952	254	104521	104006.98	311.00	29.85
200 3 4 1413	97407	97067	831	95649	93968	812	97700	97525.00		98276	97092	477	98791	98344.32	268.10	38.93
200 3 5 1358	100827	99762	628	99458	96077	1815	98326	97979.60		101463	100612	294	102049	101406.48	344.51	29.22
200 5 1 828	72277	70776	593	70731	68705	974	73619	72600.20		73040	72216	372	74922	74132.95	519.19	19.88
200 5 2 747	77551	76643	502	76297	72924	1200	74883	74403.20		78428	77236	404	79506	79073.32	278.65	16.75
200 5 3 801	75409	74497	594	74377	72924	2050	—	—		76321	75835	212	77607	77069.52	244.68	22.86
200 5 4 848	71307	69417	612	70264	67416	1138	71936	71338.80		71964	70892	399	73181	72607.25	372.38	28.07
200 5 5 815	74287	73229	465	72745	69978	1439	73825	72006.00		74936	74538	173	76022	75455.98	248.11	20.74
200 10 1 414	48006	46960	609	42016	39791	982	48119	47652.80		49212	48065	426	49883	49079.47	425.35	10.37
200 10 2 373	51438	50622	404	45483	42739	1303	51666	50410.20		52153	51568	238	53298	51831.55	459.79	8.48
200 10 3 400	50717	49688	480	45698	42475	1861	—	—		51205	50694	231	52281	51324.28	359.07	11.15
200 10 4 424	47296	45751	645	41623	39446	963	48792	47906.80		47853	47001	323	49210	48190.60	466.33	12.83
200 10 5 407	50402	49431	514	46811	42399	2023	49504	48698.00		51000	50267	293	51921	51437.97	296.41	10.99

Table 2. Results of HJ-SHC, HJ-GGA, SS-GA, SB-SSGGA and ABC on the QMKP instances with $d = 0.75$

Instance n K Num C	HJ-SHC			HJ-GGA			SS-GA			SB-SSGGA			ABC			
	Best	Avg.	SD	Best	Avg.	SD	Best	Avg.	SD	Best	Avg.	SD	Best	Avg.	SD	Time
100 3 1 669	67896	67616	327	67969	68941	480	64335	63757.20		69935	69694	133	69721	69373.07	231.72	2.07
100 3 2 714	69056	68508	313	69146	68639	340	68164	66584.80		69344	69203	71	69462	69041.00	236.38	1.86
100 3 3 686	68547	67939	361	68763	67557	832	67643	66257.00		68776	68518	199	68635	67960.05	406.74	1.86
100 3 4 666	69646	69003	437	69907	69101	861	68626	65018.40		69696	69677.40		69986	69687.68	217.28	1.88
100 3 5 668	69480	68578	341	69410	68856	306	—	—		69520	69262	146	69679	69136.40	246.88	2.12
100 5 1 401	48888	48138	286	48663	47678	749	47449	45902.00		48675	48414	125	49222	48937.47	196.48	2.07
100 5 2 428	48868	48208	317	48990	48175	398	47766	47031.80		48916	48376	249	49313	48908.05	202.63	1.96
100 5 3 411	47396	46970	244	47512	46623	503	48008	46586.80		48126	47815	156	48472	47874.50	380.30	1.71
100 5 4 400	49468	48864	254	49845	49194	295	46921	46063.00		49724	49297	150	50199	50017.93	197.12	1.83
100 5 5 400	47982	47298	311	47925	47230	554	—	—		48746	48148	267	48710	48409.75	133.12	2.01
100 10 1 200	29136	28665	262	26603	25681	688	28767	27723.20		29179	28762	136	29875	29429.20	208.12	1.22
100 10 2 214	30367	30031	187	28663	27815	391	29824	29344.40		30640	30357	120	30939	30697.80	134.28	1.45
100 10 3 205	28838	28297	238	26176	25038	562	27960	27281.60		28857	28561	152	29465	28983.78	246.93	1.30
100 10 4 200	30624	30346	199	29701	28592	333	30712	29134.60		31039	30581	238	31663	31218.85	176.33	1.40
100 10 5 200	29375	28956	206	27130	25937	650	—	—		29641	29240	147	30219	29736.47	213.33	1.43
200 3 1 131	269447	267765	809	268919	265523	1820	261106	254300.80		269351	268506	381	269736	267117.92	1070.76	14.11
200 3 2 1414	255340	253628	917	252977	249300	3409	—	—		255759	254816	438	256195	253916.75	896.46	16.27
200 3 3 1342	268682	267331	863	26773	264689	2860	269393	269061	210	268890	267079.03	1124.57	11.87	—	—	—
200 3 4 1565	245229	243881	846	243192	237837	5179	—	—		245751	244343	367	246205	244618.40	1022.02	30.64
200 3 2 1336	277221	275980	833	27762	274254	3109	—	—		277842	277048	459	279490	276605.00	1443.72	10.46
200 5 1 786	182374	181203	596	179525	177448	1331	173905	170447.20		183318	182197	485	184448	183046.65	735.60	12.34
200 5 2 848	172119	170505	947	168021	163917	3014	—	—		172158	170861	748	173575	171738.85	735.60	12.34
200 5 3 805	184362	182979	595	181412	178516	2361	—	—		184727	183975	403	186107	185059.52	469.38	12.10
200 5 4 939	163832	162584	745	160146	156246	3435	—	—		164066	163083	525	165273	164042.20	777.57	27.03
200 5 5 801	189756	188597	664	187333	185471	1179	—	—		190069	189183	334	192764	190474.27	1021.33	14.06
200 10 1 393	110238	109028	507	102002	98962	1374	106008	100939.60		110528	109755	304	111000	109624.73	714.45	7.64
200 10 2 424	102734	101595	605	92359	87400	3301	—	—		103363	102222	430	103540	102603.18	522.59	9.96
200 10 3 402	111770	110442	537	103848	100528	2362	—	—		112273	111658	274	112509	111388.20	509.42	8.04
200 10 4 469	95435	94544	532	85801	81481	2077	—	—		95839	95019	338	96859	95681.70	545.54	14.81
200 10 5 400	114260	112828	586	105078	102857	1742	—	—		114585	113607	442	115125	113909.60	590.96	8.21

the level of interaction among the objects of the instance. For each instance, the knapsacks' capacities are set to 80% of the sum of the instances objects weights divided by the number of knapsacks. The values of d , n and K are $\{0.25, 0.75\}$, $\{100, 200\}$ and $\{3, 5, 10\}$ respectively. We compare our ABC algorithm with four best approaches viz. stochastic hill climber (HJ-SHC) and generational genetic algorithm (HJ-GGA) proposed in [1], steady state grouping genetic algorithm (SB-SSGGA) proposed in [2], and genetic algorithm (SS-GA) proposed in [4].

Table 1 and Table 2 show the comparison of HJ-SHC, HJ-GGA, SS-GA and SB-SSGGA approaches with our ABC approach. For each instance these tables report the number of objects n , the number of knapsacks K in it, its number and knapsacks capacity C_k . Both tables report for each instance the best and average value of the solution and standard deviation of solution values obtained through HJ-SHC, HJ-GGA, SS-GA and SB-SSGGA. Last four columns of these tables report for each instance the best and average value of the solution, the standard deviation of solution values and average execution time obtained through ABC. Table 1 presents the results with $d = 0.25$ while Table 2 presents the results with $d = 0.75$. Among all four approaches, SS-GA presented results only on 42 instances out of 60 instances. To allow a fair comparison, we have run our ABC algorithm 40 times on each instance, which is same as in other four approaches.

Tables 1 and 2 clearly show the superiority of our approach over all the other approaches in terms of solution quality. Table 1 shows that our ABC algorithm is able to improve best known results on 25 out of 30 instances. As far as comparison of average solution quality is concerned, except for two instances (100_3_2 and

100_10_3), it outperforms all the four approaches. Table 2 shows that our ABC algorithm is able to improve best known results on 26 out of 30 instances. On 20 out of 30 instances, the average solution quality of our ABC approach is better than all the other approaches. However, our approach is slower than HJ-SHC, HJ-GGA and SB-SSGGA. As average execution time is not reported in [4], we can not compare the average execution time of our approach with SS-GA.

5 Conclusions

In this paper, we have proposed a new variant of the artificial bee colony algorithm for the quadratic multiple knapsack problem. Our proposed algorithm has been compared with four best approaches viz. HJ-SHC, HJ-GGA, SS-GA and SB-SSGGA. Computational results show the superiority of our approach over these approaches in terms of solution quality. To our knowledge, this is the first time, an ABC algorithm is used to solve a grouping problem. Similar approaches can be designed for other grouping problems also.

References

1. Hiley, A., Julstrom, B.A.: The Quadratic Multiple Knapsack Problem and Three Heuristic Approaches to it. In: Proceedings of the GECCO 2006, pp. 547–552. ACM Press, New York (2006)
2. Singh, A., Baghel, A.S.: A New Grouping Genetic Algorithm for the Quadratic Multiple Knapsack Problem. In: Cotta, C., van Hemert, J. (eds.) EvoCOP 2007. LNCS, vol. 4446, pp. 210–218. Springer, Heidelberg (2007)
3. Falkenauer, E.: Genetic Algorithms and Grouping Problems. John Wiley & Sons, Chichester (1998)
4. Saraç, T., Sipahioglu, A.: A Genetic Algorithm for the Quadratic Multiple Knapsack Problem. In: Mele, F., Ramella, G., Santillo, S., Ventriglia, F. (eds.) BVAI 2007. LNCS, vol. 4729, pp. 490–498. Springer, Heidelberg (2007)
5. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey (2005)
6. Karaboga, D., Basturk, B.: On the Performance of Artificial Bee Colony (ABC) Algorithm. *Applied Soft Computing* 8, 687–697 (2008)
7. Singh, A.: An Artificial Bee Colony Algorithm for the Leaf-Constrained Minimum Spanning Tree Problem. *Applied Soft Computing* 9, 625–631 (2009)
8. Sundar, S., Singh, A.: A Swarm Intelligence Approach to the Quadratic Minimum Spanning Tree Problem. *Information Sciences* 180, 3182–3191 (2010)
9. Pan, Q.-K., Tasgetiren, M.F., Suganthan, P.N., Chua, T.J.: A Discrete Artificial Bee Colony Algorithm for the Lot-Streaming Flow Shop Scheduling Problem. *Information Sciences* (in Press), doi:10.1016/j.ins.2009.12.025
10. Karaboga, D., Akay, B.: A Survey: Algorithms Simulating Bee Swarm Intelligence. *Artificial Intelligence Review* 31, 61–85 (2009)

Rough-Set-Based Association Rules Applied to Brand Trust Evaluation Model

Shu-Hsien Liao¹, Yin-Ju Chen², and Pei-Hui Chu²

¹ Department of Management Sciences and Decision Making, Tamkang University,
No. 151, Yingjuan Road, Danshuei Jen, Taipei 251, Taiwan, ROC

² Department of Management Sciences, Tamkang University,
No. 151, Yingjuan Road, Danshuei Jen, Taipei 251, Taiwan, ROC
michael@mail.tku.edu.tw, s5515124@ms18.hinet.net,
mailsphi@gmail.com

Abstract. Of the consumers who often patronize retail stores, 87 of 100 respondents visited a convenience store in the past three months. The superstore/hypermarket and the supermarket came in second and third, respectively. This demonstrates that retail channels are essential to the day-to-day life of the common populace. With the social and economic evolution, not only have product sales and shopping habits changed, but the current marketing concepts have also changed from being product-oriented to being consumer-oriented. In this research, we first provide new algorithms modified from the Apriori algorithm. The new approach can be applied in finding association rules, which can handle an uncertainty, combined with the rough set theory, and then to find the influence degree of the consumer preferences variables for the marketing decision-makers used.

Keywords: Data mining, Rough set, Association rule, Retailing industry, Brand trust.

1 Introduction

We can judge consumer decisions on the basis of certain rules of thumb when the consumer choice factors taken into account are simple. However, when we have a variety of choices as well as an increasing number of factors to consider, it is important to determine how a simple analysis and consumer rule of thumb can help to determine the shopping behavior of consumers. In such a case, we may need a more rigorous approach to help us determine future consumer decision making and to find a complex combination of factors that affect the decision making irrespective of whether the effects of these factors are tangible. The rest of this paper is organized as follows. Section 2 reviews relevant literature related to the research and the problem statement. Section 3 presents the data processing for the new algorithm. Closing remarks and a discussion of future work are presented in Section 4.

2 Literature Review and Problem Statement

Data mining (DM), sometimes referred to as knowledge discovery in a database, is a systematic approach used for finding underlying patterns, trends, and relationships

buried in data. DM has drawn serious attention from both researchers and practitioners because of its wide applications in crucial business decisions [7]. The related applications using these methodologies can be summarized as classification, prediction, clustering, summarization, dependency modeling, linkage analysis, and sequential analysis [8]. The rough set theory is different from the fuzzy theory and neural networks. Further, a majority of scholars have mentioned the use of association rules that are required for dealing with uncertainties or inaccurate information. In this research, we further explore the use of the rough set theory to improve the use of association rules.

“If the gender is Male, age is 30 years, and income is more than 35,000, then the favorite milk brand is Lin-Feng-Ying.” In the case of R2, the decision-making rule is “If the gender is Male, age is 45 years, and income is more than 80,000, then the favorite milk brand is Wei-Chuan.” From the information given above, we observe that the attributes of “age” and “income” affect the preference for a certain milk brand. Further, we may want to know whether “a male whose favorite milk brand is Lin-Feng-Ying would relatively like the Wei-Chuan brand of milk” or whether “a male whose favorite milk brand is Lin-Feng-Ying would absolutely like the Wei-Chuan brand of milk.” That is, we may wish to know the relationships between attributes that are substitutes or complements and so on. However, without further information of rules, generated by the traditional rules, we cannot derive the necessary information. Therefore, when the rules have a hierarchical or ordinal structure, provided by the application of knowledge is very meaningful. If “age (A) is 30 years (a1) and revenue (R) is 35,000 or more (r1),” certain conditions of $a_1 \leq A \leq a_2$ and $r_1 \leq R \leq r_2$, where a1 and a2 and r1 and r2 correspond respectively to A and R, are satisfied.

3 Methodology—Algorithm Concept and Data Processing

We hope to figure out the consumer’s subjective or objective point of view and preferences by using ratio-scale algorithms combined with the rough set theory, and then find the influence degree of the consumer preferences variables for the marketing decision-makers used. Fig. 1 presents the algorithm concept and data processing.

First: Data processing—Definition 1—Information system: Transform the questionnaire answers into information system $IS = (U, Q)$, where $U = \{x_1, x_2, \dots, x_n\}$ is a finite set of objects. Q is usually divided into two parts, $G = \{g_1, g_2, \dots, g_m\}$ is a finite set of general attributes/criteria, and $D = \{d_1, d_2, \dots, d_l\}$ is a set of decision attributes. $f_g = U \times G \rightarrow V_g$ is called the information function, V_g is the domain of the attribute/criterion g , and f_g is a total function such that $f(x, g) \in V_g$ for each $g \in Q$; $x \in U$. $f_d = U \times D \rightarrow V_d$ is called the sorting decision-making information function, V_d is the domain of the decision attributes/criterion d , and f_d is a total function such that $f(x, d) \in V_d$ for each $d \in Q$; $x \in U$.

```

Algorithm-Step1
Input:
Information System (IS);
Output:
{Trust Value};
Method:
1. Begin
2.   IS = (U,Q);
3.    $x_1, x_2, \dots, x_n \in U$ ; /* where  $x_1, x_2, \dots, x_n$  are the objects of set  $U$  */
4.    $G, D \subset Q$ ; /*  $Q$  is divided into two parts  $G$  and  $D$  */
5.    $g_1, g_2, \dots, g_m \in G$ ; /* where  $g_1, g_2, \dots, g_m$  are the elements of set  $G$  */
6.    $d_1, d_2, \dots, d_l \in D$ ; /* where  $d_1, d_2, \dots, d_l$  are the elements of set  $D$  */
7.   For each  $g_m$  and  $d_i$  do;
8.     compute  $f(x,g)$  and  $f(x,d)$ ; /* compute the information function in IS as
described in definition1*/
9.     compute  $R(g_m)$  and  $R(d_i)$ ; /* compute the similarity relation in IS as
described in definition2*/
10.    compute  $F_p$ ; /* compute the  $V_g$  as condition attributes in  $R(g_m)$  as
described in definition3*/
11.    compute  $F(g_m^v, d_i^v)$ ; /* compute the  $P(d_i^v)$  as condition attributes in  $R(d_i)$ 
as described in definition4*/
12.    compute  $E_c$ ; /* compute the brand image trust model as described in
definition5*/
13.  Endfor;
14.  Output { Trust Value };
15. End;

Algorithm-Step2
Input:
Decision Table (DT);
Output:
{Classification Rules};
Method:
1. Begin
2.   DT = (U,Q);
3.    $x_1, x_2, \dots, x_n \in U$ ; /* where  $x_1, x_2, \dots, x_n$  are the objects of set  $U$  */
4.    $Q = (G, D)$ ;
5.    $g_1, g_2, \dots, g_m \in G$ ; /* where  $g_1, g_2, \dots, g_m$  are the elements of set  $G$  */
6.    $d_1, d_2, \dots, d_l \in D$ ; /* where  $d_1, d_2, \dots, d_l$  are the "trust value" generated in
Step1*/
7.   For each  $d_i$  do;
8.     compute  $f(x,g)$ ; /* compute the information function in DT as described in
definition1*/
9.     compute  $R_m$ ; /* compute the similarity relation in DT as described in
definition2*/
10.    compute  $ind(G)$ ; /* compute the relative reduct of DT as described in
definition3*/
11.    compute  $ind(G - g_m)$ ; /* compute the relative reduct of the elements for element
 $m$  as described in definition3*/
12.    compute  $\underline{G}(X)$ ; /* compute the lower-approximation of DT as described in
definition4*/
13.    compute  $\bar{G}(X)$ ; /* compute the upper-approximation of DT as described in
definition4*/
14.    compute  $Bn_G(X)$ ; /* compute the bound of DT as described in definition4*/
15.    compute  $Sup(ind(B))$ ; /* compute the support as described in definition5*/
16.    compute  $Conf(ind(B) \rightarrow d_{g_m})$ ; /* compute the confidence as described in
definition5*/
17.  Endfor;
18.  Output {Classification Rules};
19. End;

```

Fig. 1. Algorithm concept and data processing

Example: According to Tables 1 and 2, x_1 is a male who is thirty years old and has an income of 35,000. He ranks beer brands from one to eight as follows: Heineken, Miller, Taiwan light beer, Taiwan beer, Taiwan draft beer, Tsingtao, Kirin, and Budweiser.

Table 1. Information system

U	Q	General attributes G			Decision-making D
		Item1: Gender g_1	Item2: Age g_2	Item3: Income g_3	
x_1		Male g_{11}	30 g_{21}	35,000 g_{31}	As shown in Table 4.
x_2		Male g_{11}	40 g_{22}	60,000 g_{32}	As shown in Table 4.
x_3		Male g_{11}	45 g_{23}	80,000 g_{34}	As shown in Table 4.
x_4		Female g_{12}	30 g_{21}	35,000 g_{31}	As shown in Table 4.
x_5		Male g_{11}	40 g_{22}	70,000 g_{33}	As shown in Table 4.

Table 2. Beer brand recall ranking table

U	D the sorting decision-making set of beer brand recall							
	Taiwan beer d_1	Heineken d_2	Taiwan light beer d_3	Miller d_4	Taiwan draft beer d_5	Tsingtao d_6	Kirin d_7	Budweiser d_8
x_1	4	1	3	2	5	6	7	8
x_2	1	2	3	7	5	6	4	8
x_3	1	4	3	2	5	6	7	8
x_4	3	1	6	2	5	4	8	7
x_5	1	3	6	2	5	4	8	7

Definition 2—Similarity relation: According to the specific universe of discourse classification, a similarity relation of the general attributes $g \in G$ is denoted as U/G , and a similarity relation of the decision attributes $d \in D$ is denoted as U/D

$$U|G = \{[x_i]_G | x_i \in U\} \quad U|D = \{[x_i]_D | x_i \in U\}$$

Example:

$$\begin{aligned} R(g_2) &= U/g_2 = \{\{x_1, x_4\}, \{x_2, x_5\}, \{x_3\}\} & R(d_1) &= U/d_1 = \{\{x_1\}, \{x_2, x_3, x_5\}, \{x_4\}\} \\ R(g_3) &= U/g_3 = \{\{x_1, x_4\}, \{x_2\}, \{x_5\}, \{x_3\}\} & R(d_2) &= U/d_2 = \{\{x_1, x_4\}, \{x_2\}, \{x_3, x_5\}\} \end{aligned}$$

Definition 3—Core value: We distinguish the hidden relation between the general attribute g_{i-1} and g_i , and then set the similarity relation of the quantity attribute as the top priority. $R^+(g_i)$ denotes the set of the general attribute objects as $g_{i_1} < g_{i_2} < \dots < g_{i_n} \cap f(g_{i_1}) < f(g_{i_2}) < \dots < f(g_{i_n})$ for $\forall g_i \in G$, otherwise $R^-(g_i)$. $R^+(g_j)$ denotes the set of the general attribute objects as $g_{j_1} < g_{j_2} < \dots < g_{j_n} \cap f(g_{j_1}) < f(g_{j_2}) < \dots < f(g_{j_n})$ for $\forall g_j \in G$, otherwise $R^-(g_j)$. $F_P(g_i, g_j)$ is the ratio relation between U/g_i and U/g_j . In the condition U/g , the set of x_i is defined as follows:

$$\begin{aligned}
F_P^+(g_i, g_j) &= \left\{ x_i \mid U/g, R^+(g_i) = R^+(g_j) \cup R^-(g_i) = R^-(g_j) \right\} \\
F_P^-(g_i, g_j) &= \left\{ x_i \mid U/g, R^+(g_i) = R^-(g_j) \cup R^-(g_i) = R^+(g_j) \right\} \\
F_P^0(g_i, g_j) &= \left\{ x_i \mid U/g, R^-(g_i) \neq R^+(g_j) \cup R^+(g_i) \neq R^-(g_j) \cup R^+(g_i) \neq R^+(g_j) \cup R^-(g_i) \neq R^-(g_j) \right\}
\end{aligned}$$

Example: According to Table 1, age and income are quantity attribute, and ratio relation between those two attributes is $F_P^+(g_2, g_3)$, that mean the income increases along with the age grows and are defined, respectively, as

$$R^+(g_2) = \{x_1, x_4, x_2, x_5, x_3\} \quad R^+(g_3) = \{x_1, x_4, x_2, x_5, x_3\} \quad F_P^+(g_2, g_3) = \{x_1, x_4, x_2, x_5, x_3\}$$

Definition 4—Similarity relation between general attribute and decision attributes: The decision attributes in the information system are an ordered set, therefore, the attribute values will have an ordinal relation defined as follows:

$$F(g_{ij}, d_l) = \begin{cases} P^+(d_l) : R(d_{l_1}) = F_P^+(1) \cap R(d_{l_i}) = F_P^+(i) \\ P^-(d_l) : R(d_{l_1}) = F_P^+(i) \cap R(d_{l_i}) = F_P^+(1) \\ P^0(d_l) : \text{else} \end{cases}$$

Example: The similarity relation between the general attribute and the decision attributes given in Tables 1 and 2 is presented in Table 3. For example, $R(d_1) = U/d_1 = \{\{x_1\}, \{x_2, x_3, x_5\}, \{x_4\}\}$ and $R(d_2) = U/d_2 = \{\{x_1, x_4\}, \{x_2\}, \{x_3, x_5\}\}$ show that the two decision attributes correspond to the quantity attribute, such as the quantity attribute value increased as the better ranking of Taiwan beer, but the quantity attribute value increased as the Worse ranking of Heineken. From this, we may conclude that Taiwan beer was well received by old people, and Heineken was well received by young people.

Table 3. Similarity relation between general attribute and decision attributes

d_l	$R(d_{l_1})$	$R(d_{l_i})$	$F_P^+(1)$	$F_P^+(i)$	$F(g_{ij}, d_l)$
d_1	$\{x_2, x_3, x_5\}$	$\{x_1\}$	x_1	x_3	$P^-(d_1)$
d_2	$\{x_1, x_4\}$	$\{x_3\}$	x_1	x_3	$P^+(d_2)$
			\vdots		

Definition 5—Assessment model to establish brand trust: An assessment model for establishing brand trust, denoted by E_i , is defined as $E_c = \alpha \times (V_d)^{-1}$, $0 \leq E_c \leq 2$. Here, α represents the weight of the assessment model. When the similarity relation between the general attribute and the decision attributes is a positive

correlation or a negative correlation, such as $P^+(d_l)$ and $P^-(d_l)$, we obtain $E_c = \alpha \times (V_d)^{-1}$, where $\alpha = 2$, and the remainder $\alpha = 1$.

Example: By calculating the brand trust value shown in Table 4, we obtain the following: $E_1 = 2 \times 1/8 \times \text{Heineken}(d_2) = 1/4 = 0.25$, which indicates that the brand trust value of x_1 with respect to age and income is 0.25. $E_3 = 2 \times 1/5 \times \text{Heineken}(d_2) = 2/5 = 0.4$ indicates that the Heineken brand trust value of x_1 with respect to age and income is 0.4. The total brand trust value of Heineken is $(2+1+2/3+1+2/3)/5 = 16/15 = 1.067$.

Table 4. Brand trust value

U		$E_c^{x_2, x_3}$					Total brand trust value
		x_1	x_2	x_3	x_4	x_5	
Taiwan beer	d_1	1/2	2	2	2/3	2	41/30 = 1.376
Heineken	d_2	2	1	2/3	1	2/3	16/15 = 1.067

Second: Generated rough associational rule—Definition 1—Decision table: In the first step of this study, we found examples of the potential relationship between the attributes and calculated the degree of brand trust. Then, we generated rough association rules. To consider other attributes into, and to establish the decision table with the brand trust value as the highest decision-making attributes, shown as Table 5.

Table 5. Table of degree-of-brand-trust-based decision-making

Q		General attributes				Decision attributes	
		Product features g_1	Product information source g_2	Consumer behavior g_3	Channels g_4	Total trust value	Product
U	x_1	Price	Seen on shelves	Purchase by promotions	Convenience stores	1/2	Taiwan beer
	x_2	Price	Advertising	Purchase by promotions	Hypermarkets	2	Taiwan beer
	x_3	Brand	Seen on shelves	Will not purchase by promotions	Convenience stores	2	Taiwan beer

Definition 2—Similarity relation: According to the specific universe of the discourse classification, the similarity relation of the general attributes $g \in G$ is denoted as U/G . All the similarity relations are denoted by $K = (U, R_1, R_2 \dots R_{m-1})$.

$$U|G = \{[x_i]_G | x_i \in U\}$$

Example:

$$R_1 = U/g_1 = \{\{x_1, x_2\}, \{x_3\}\} \quad R_5 = U/g_1 g_3 = \{\{x_1, x_2\}, \{x_3\}\}$$

Definition 3—Reduct and core: According to the similarity relation, and then finding the reduct and core. The attribute g , which was ignored from G , and the set G will not be affected, thereby g is the unnecessary attribute and we can reduct it. $R \subseteq G$ and $\forall g \in R$. A similarity relation of the general attributes from the decision table are denoted as $ind(G)$. If $ind(G) = ind(G - g_1)$, then g_1 is the reduct attribute, and if $ind(G) \neq ind(G - g_1)$, then g_1 is the core attribute.

Example: $U|ind(G) = \{\{x_1\}, \{x_2\}, \{x_3\}\}$

$$U|ind(G - g_1) = U|(\{g_2, g_3, g_4\}) = \{\{x_1\}, \{x_2\}, \{x_3\}\} = U|ind(G)$$

$$U|ind(G - g_1 g_3) = U|(\{g_2, g_4\}) = \{\{x_1, x_3\}, \{x_2\}\} \neq U|ind(G)$$

When g_1 is considered alone, g_1 is the reduct attribute; however, when g_1 and g_3 are considered simultaneously, g_1 and g_3 are the core attributes.

Definition 4—Lower approximation and upper approximation: The lower approximation, denoted as $\underline{G}(X)$, is defined as the union of all the elementary sets that are contained in $[x_i]_G$. More formally:

$$\underline{G}(X) = \bigcup \{[x_i]_G \in U/G | [x_i]_G \subseteq X\}$$

The upper approximation, denoted as $\bar{G}(X)$, is the union of the elementary sets that have a non-empty intersection with $[x_i]_G$. More formally:

$$\bar{G}(X) = \bigcup \{[x_i]_G \subseteq U/G | [x_i]_G \cap X \neq \emptyset\}$$

The difference: $Bn_G(X) = \bar{G}(X) - \underline{G}(X)$ is called a boundary of $[x_i]_G$.

Example: $\{x_1, x_2, x_4\}$ are the customers that we are interested in; therefore, $\underline{G}(X) = \{x_1\}$, $\bar{G}(X) = \{x_1, x_2, x_3, x_4, x_5\}$, and $Bn_G(X) = \{x_2, x_3, x_4, x_5\}$.

Definition 5: By using the traditional association rule to calculate the value of Support and Confidence, we derive the formula as follows:

$$Sup(ind(B)) = \left| \{ind(B) | \underline{G}(X) \subseteq \bar{G}(X)\} \right| = \frac{|ind(B)|\underline{G}(X)}{|\bar{G}(X)|}$$

$$Conf(ind(B) \rightarrow d_{g_m}) = \left| \{ind(B) \cap d_{g_m} | Sup(ind(B))\} \right| = \frac{Sup(ind(B) \cap d_{g_m})}{Sup(ind(B))}$$

Definition 6: Rough-set-based association rules.

$$\{x_1\}/g_1 g_3 : g_{1_1} \cap g_{3_1} \Rightarrow d_{d_1}^1 = 4 \quad \{x_1\}/g_1 g_2 g_3 g_4 : g_{1_1} \cap g_{2_1} \cap g_{3_1} \cap g_{4_1} \Rightarrow d_{d_1}^1 = 4$$

4 Conclusion and Future Work

The traditional association rule should be fixed in order to avoid both the retention of only trivial rules and the discarding of interesting rules. In fact, using a relative comparison to express the association rule is more complete than those that use an absolute comparison. In this study, a new approach was applied to find association rules, which could handle the uncertainty in the classification process and was suitable for the ratio scale data. Private brands are increasingly finding their way into new product/market shares. Using the suggested methodology, a decision maker can make accurate decisions regarding the segmentation and assistance required for developing a new product. The system needs to re-calculate and find new rules when the conditions of the traditional association rules change. Thus, in this study, we extended the concept of a utility function used for establishing the demand for users to adjust the brand image with the brand-trust evaluation model.

Acknowledgements. This research was funded by the National Science Council, Taiwan, Republic of China, under contract No. NSC 98-2410-H-032-038-MY2.

References

1. Lee, J.-S., Back, K.-J.: Attendee-based brand equity. *Tourism Management* 29, 331–344 (2008)
2. Kent, R.J., Kellaris, J.: Competitive interference effects in memory for advertising: are familiar brands exempt? *J. Mark Commun.* 7, 59–69 (2001)
3. Walczak, B., Massart, D.L.: Rough sets theory. *Chemometrics & Intelligent Laboratory Systems* 47,1, 1–16 (1999)
4. Greco, S., Inuiguchi, M., Slowinski, R.: Fuzzy rough sets and multiple-premise gradual decision rules. *International Journal of Approximate Reasoning* 41, 179–211 (2006)
5. Liu, M., Chen, D., Wu, C., Li, H.: Fuzzy reasoning based on a new fuzzy rough set and its application to scheduling problems. *Computers and Mathematics with Applications* 51, 1507–1518 (2006)
6. Lian, W., Cheung, D.W., Yiu, S.M.: An efficient algorithm for finding dense regions for mining quantitative association rules. *Computers and Mathematics with Applications* 50, 471–490 (2005)
7. Lee, T.-S., Chiu, C.-C., Chou, Y.-C., Lu, C.-J.: Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Computational Statistics and Data Analysis* 50(4), 1113–1130 (2006)
8. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM* 39, 27–34 (1996)

A Genetic Algorithm to Find Pareto-optimal Solutions for the Dynamic Facility Layout Problem with Multiple Objectives

Kazi Shah Nawaz Ripon, Kyrre Glette, Mats Høvin, and Jim Torresen

Department of Informatics, University of Oslo, Norway
`{ksripon, kyrrehg, matsh, jimtoer}@ifi.uio.no`

Abstract. In today's volatile manufacturing scenario, manufacturing facilities must operate in a dynamic and market-driven environment in which production rates and production mixes are continuously changing. To operate efficiently within such an environment, the facility layout needs to be adaptable to changes. The dynamic facility layout problem (DFLP) deals with changes of layout over time. DFLPs are usually solved just considering quantitative aspect of layout alone, ignoring the qualitative aspect. Few attempts have been made to date to deal with the multi-objective DFLP. These most often use the weighted-sum method to combine different objectives and thus, inherit the well-known problems of this method. The objective of this paper is to introduce an evolutionary approach for solving multi-objective DFLP that presents the layout as a set of Pareto-optimal solutions optimizing both quantitative and qualitative objectives simultaneously. Experimental results obtained with the proposed approach are promising.

Keywords: Pareto-optimal solutions, Dynamic facility layout problem, Multi-objective evolutionary optimization.

1 Introduction

The facility layout problem (FLP) asks for an efficient arrangement of departments within a facility like manufacturing plants, administrative office buildings, or service facilities to optimize several objectives. It is an intriguing combinatorial problem of high complexity and the runtime for solving it quickly increases with the number of departments to be laid out. Layout planning plays an important role in the manufacturing process and has a serious impact on a company's profitability. A good layout can contribute to the overall efficiency of operations and can reduce the total operating expenses up to 50% [1].

Traditionally, there are two basic types of objectives for FLP. The first one is the quantitative (distance-based) objective aiming at minimizing the total material handling (MH) cost between departments based on a distance function. The second one is the qualitative (adjacency-based) goal, aimed at maximizing the closeness relationship (CR) scores between departments based on the placement of departments that utilize common materials, personnel, or utilities adjacent to one another, while separating departments

for reasons of safety, noise, or cleanliness. Over the years, extensive research has been conducted on FLP. Yet, most of the research conducted in this field has concerned a single objective, either qualitative or quantitative goodness of the layout. In general, minimization of the total MH costs is often used as the optimization criterion in FLP. However, closeness, hazardous movement or safety, and similar criteria are also important in FLP. Inherently, real-life layout problems are multi-objective by nature and the FLP must consider both quantitative and qualitative objectives simultaneously. Consequently, FLP falls into the category of multi-objective optimization problem (MOOP).

Although dealing with multiple objectives has received attention over the last few years [2], [3], these approaches are still considered limited, and mostly dominated by the unrealistic weighted-sum method. In this method, multiple objectives are added up into a single scalar objective using weighted coefficients. However, there are several disadvantages of this technique [4]. Firstly, as the relative weights of the objectives are not exactly known in advance and cannot be pre-determined by the users, the objective function that has the largest variance may dominate the multi-objective evaluation. As a result, inferior non-dominated solutions with poor diversity will be produced. Secondly, the user always has to specify the weight values for functions and sometimes these will not have any relationship with the importance of the objectives. Thirdly, solutions are obtained one at a time. If we are interested in obtaining a set of feasible solutions, it has to be run several times. This also is not a guarantee that the solutions obtained in different runs are different. In practice, the layout designer – based on his/her past experience – randomly selects the layout having multiple objectives. This restricts the designing process to be completely designer dependent and thus, the layout varies from designer to designer. To overcome such difficulties, Pareto-optimality has become an alternative to the classical weighted-sum method.

In the past, most of the research generally concerns static FLP (SFLP), where the material flows between facilities in the layout have been assumed to be invariant over time. However, most manufacturing facilities today operate in a dynamic environment in which layouts are constantly changing, either in response to customers' demands for changes in product designs and functionalities or to keep pace with technological innovations. The introduction of new products/machines and the removal of others or the realization of an increase or decrease in throughput volume can render the existing layout completely unreliable in yielding improved productivity. MH costs and other qualitative objectives are no longer constant over the planning horizon, creating the need for radical modifications in layout. It is therefore often necessary to analyze and redesign the current layout to accommodate the constant changes in flow and demand over the planning horizon. As a result, the DFLP has garnered much interest recently.

Problems related to facility layout are computationally difficult (NP-hard). In an n department, t period DFLP, the maximum number of different layouts is $(n!)^t$. We would have to evaluate 1.93×10^{14} possibilities for even just a 6-department, 5-period problem. Due to the combinatorial nature of the problem, optimal algorithms have been successfully applied only to small problems, but they require high computational efforts and extensive memory capabilities. As the number of departments increases, so does the number of possible combinations to be evaluated. Since exact approaches are often found unsuitable for large problems, numerous researchers have developed heuristics and meta-heuristics for the FLP. Among those approaches, the genetic algorithm (GA) seems to become quite popular in solving FLP [3]. In recent years, numerous intelligent

and meta-heuristic techniques have been developed to cope with DFLPs. However, often the research in this area is focused on single objective. Very few attempts have been made to date for the combined areas of multi-objective and dynamic FLP [5]. With the exception of the weighted-sum method, there are no known papers published for finding Pareto-optimal solutions for multi-objective DFLP.

Thus, this is the first paper to present an evolutionary approach for solving multi-objective DFLP to find a set of Pareto-optimal layouts. It is based on the non-dominated sorting genetic algorithm-2 (NSGA-2) [6]. The experimental results demonstrate that the proposed Pareto-optimal based multi-objective dynamic FLP (PMDFLP) approach can optimize multiple objectives simultaneously throughout the entire evolutionary process that provide a wide range of alternative layout choices. Furthermore, each solution is near optimal and is not dominated by consideration of a particular objective. In the next section, we describe the importance of Pareto-optimality in the FLP. In Section 3, we describe relevant works. Section 4 presents the mathematical formulation for the multi-objective DFLP. Section 5 outlines the proposed approach. The proposed PMDFLP has been experimentally verified in Section 6. Finally, we provide the conclusion of the paper in Section 7.

2 Importance of Pareto-optimality in FLP

In the world around us, there are few problems concerned with a single value or objective. It is also common that conflicts exist among the objectives. In such MOOPs, there is no single optimal solution; rather there is a set of alternative solutions. These solutions, namely Pareto-optimal solutions [7], are optimal in the wider sense that no other solutions in the search space are superior when all the objectives are considered. In a Pareto-optimal set, any two solutions of this set do not dominate each other and one or more members of this set dominate all other solutions in the search space excluding this set. Fig. 1 presents a Pareto front for two objectives, which are subject to minimization.

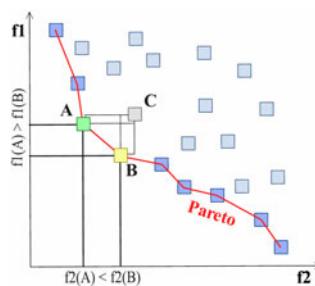


Fig. 1. Pareto Front [7]

Historically, FLPs have been solved only for one goal, either quantitative or qualitative goodness of the layout. In real-world FLPs, it is often necessary to optimize both criteria simultaneously. In fact, real-life layout problems are multi-objective by nature and they require the decision makers to consider both quantitative and qualitative objectives before

arriving at any conclusion. A solution that is optimal with respect to a certain given criterion might be a poor candidate for some others. Hence, it is desirable to generate many near-optimal layouts considering multiple objectives according to the requirements of the production order or customer demand. Then, the production manager can choose the most suitable of the generated layouts for specific order or customer demands. Based on the principle of multi-objective optimization, obtaining an optimal solution that satisfies all objectives is almost impossible. Objectives tend to conflict; improvement with respect to one objective is usually accompanied by decline with respect to another. Accordingly, it is desirable to obtain a variety of solutions, which should be converged to, and diverse along the Pareto-optimal front.

3 Related Works

The FLP has been extensively studied over the last few decades and a variety of approaches have been proposed. However, they are mostly static in nature. Comprehensive surveys are found in [3], [8]. Rosenblatt has been the first to address the basic DFLP and sketch out a method based on dynamic programming to solve the DFLP [9]. Since then, a large number of exact and heuristic algorithms have been designed and published for the DFLP. Though exact algorithms find the best solution, they are not suitable for large DFLPs due to their high computational effort and memory requirements. As a result, heuristic and meta-heuristic algorithms have received attention due to their ability to generate feasible solutions in the least possible computational time. Among those approaches, GA has found wide application in research intended to solve DFLP due to its capability to generate feasible solutions in a minimum amount of time. Conway and Venkataraman presented an early effort to apply GA in solving the DFLP [10]. Balakrishnan *et al.* extended the existing GA to a hybrid GA to address the DFLP [11]. A recent application of GA using new genetic operators can be found in [12], [13]. Unfortunately, in all existing DFLP models, attention has been given mainly to the quantitative objective to find the minimum total cost among various alternative layouts. To date, there have been only a few attempts to tackle the multi-objective DFLP [5], but they have used the weighted-sum method. Thus, they have ignored the prospects of Pareto-optimal solutions in solving multi-objective DFLP.

4 Problem Formulation

Traditionally, the FLP has been presented as a quadratic assignment problem (QAP), which assigns n equal area departments to n locations. The DFLP can be formulated as an extension of SFLP by selecting a static layout for each period and then deciding whether to change to a different layout in the next period or not. Fig. 2 presents a 2x3 DFLP instance with 6-department and 3-time period. In this example, departments 3, 5, 2, 6, 1, and 4 are assigned to locations 1, 2, 3, 4, 5, and 6 respectively in period 1, and so on.

<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>3</td><td>5</td><td>2</td></tr> <tr><td>6</td><td>1</td><td>4</td></tr> </table>	3	5	2	6	1	4	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1</td><td>5</td><td>2</td></tr> <tr><td>6</td><td>4</td><td>3</td></tr> </table>	1	5	2	6	4	3	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1</td><td>6</td><td>5</td></tr> <tr><td>3</td><td>4</td><td>2</td></tr> </table>	1	6	5	3	4	2
3	5	2																		
6	1	4																		
1	5	2																		
6	4	3																		
1	6	5																		
3	4	2																		
$t = 1$	$t = 2$	$t = 3$																		

Fig. 2. A 2x3 DFLP instance with 6-department, 3-period

In this work, we separately apply MH costs and CR scores among various departments as quantitative and qualitative objective, respectively. The following mathematical model expresses the objectives:

$$F_1 = \sum_{t=2}^P \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} Y_{tijl} + \sum_{t=1}^P \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N f_{tik} d_{tjl} X_{tij} X_{tik} \quad (1)$$

$$F_2 = \sum_{t=1}^P \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N W_{ijkl} X_{tij} X_{tik} \quad (2)$$

Subject to

$$\sum_{i=1}^N X_{tij} = 1, \quad j=1,2,\dots,N, \quad t=1,2,\dots,P \quad (3)$$

$$\sum_{i=1}^N X_{tij} = 1, \quad i=1,2,\dots,N, \quad t=1,2,\dots,P \quad (4)$$

$$Y_{tijl} = X_{(t-1)ij} X_{til}, \quad i,j,l=1,2,\dots,N, \quad t=2,3,\dots,P \quad (5)$$

$$X_{tij} = 0 \text{ or } 1 \quad (6)$$

$$W_{ijkl} = \begin{cases} r_{ik}, & \text{if locations } j \text{ and } l \text{ are neighbors} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Where, i, k are departments; j, l are locations in the layout; Y_{tijl} is 1 for shifting i from j to l in period t , otherwise 0; A_{ijl} is the fixed cost of shifting i from j to l in period t (where $A_{tijj} = 0$); X_{tij} is 1 if i is at j in period t , otherwise 0; f_{tik} is the flow cost for unit distance from facility i to k in period t ; d_{tjl} is the distance from j to l in period t ; r_{ik} is the closeness ranking value in period t when departments i and k are neighbors with common boundary; N is the number of departments in the layout; and P is the number of periods for the planning horizon. Here, the aim of the first fitness function is to minimize the sum of the layout rearrangement cost (first term) and the MH costs (second term) over the planning horizon (Eq.(1)). The second fitness function is subject to maximization of the adjacency requirement based on closeness ranking value (Eq.(2)). Since the rearrangement cost should be calculated in the first fitness function, it is not included in the second objective for the same chromosome.

5 Pareto-optimal Based Multi-objective Dynamic FLP (PMDFLP) Approach

In the proposed PMDFLP, a form of direct representation for chromosomes is used. The solution is represented as a string of integers of length $N \times P$, where N is the total number of departments, and P is the number of periods for the planning horizon. The chromosomes are encoded as $(a_{11}a_{12}a_{13}\dots\dots a_{1N})(a_{21}a_{22}a_{23}\dots\dots a_{2N})\dots\dots(a_{P1}a_{P2}a_{P3}\dots\dots a_{PN})$. The integers denote the departments and their positions in the string

denote the positions of the departments in the layout for that period. For example, the 6-department, 3-period problem mentioned in fig. 1 will be represented as (352614) (152643) (165342). This representation is very useful in the case of GA because a chromosome will be chosen for any operation if it has a good objective value, which is easy to calculate. We extended the crossover operator used for SFLP in [14], [15] to DFLP. To keep the chromosome valid for every period after crossover, some repairs are required, especially for the periods before and after the crossover point. For mutation, we use swap mutation with slight modification to adapt to the multi-objective DFLP. In this approach, two genes randomly chosen from the same time period are exchanged. The resultant chromosomes are legal and no repair is required. After performing all these operations, a non-dominated sorting strategy employing the crowding-distance assignment [6] is performed to achieve elitism for the next generation.

6 Computational Results

All experiments are conducted using a population of 150 chromosomes and 200 generations for problems with up to 15 departments; and 300 chromosomes and 400 generations for problems with more than 15 departments. The probabilities of crossover and mutation are 0.9 and 0.3, respectively. We used traditional tournament selection with tournament size 2. Each problem is tested 30 times with different seeds. Then each of the final generations is combined and non-dominated sorting is performed to produce the final non-dominated solutions.

There are currently no benchmark datasets published for the combined areas of dynamic and multi-objective FLP. In [5], the author proposed an ACO based approach to solve multi-objective DFLP (ACO-DML), which is the only available literature in these combined fields. Yet, the author justified his approach separately using dynamic (Lacksonen and Enscore [5]) and multi-objective (Dutta and Sahu [16]) data sets. Moreover, in case of multi-objective FLP, he used the weighted-sum method and ignored the prospects of Pareto-optimality. We have ourselves extended the dynamic data sets as multi-objective dynamic data sets by creating the data sets for CR score to compare with [5]. In [5], the author also proposed two data sets for multi-objective FLP (we call it ‘Chen’) and separately mentioned the values for both objectives. For fair comparison, we have also compared our approach with these values.

Table 1 shows the comparison of the proposed PMDFLP with ACO-DML in terms of dynamic and multi-objective aspects. We mentioned earlier that ACO-DML used these data sets only for dynamic, single-objective cases. As it is seen in Table 1, the obtained MH cost by PMDFLP for most of the test problems outperforms the MH cost found by ACO-DML, with percentage deviations as high as 27.76%. Table 2 summarizes the comparison results of these two approaches in terms of both MH cost and CR score. As indicated by Table 2, the performance of the proposed approach is clearly superior to the results obtained by ACO-DML. This time the performance deviation is as high as 56.72%. From both tables, we can find some values with negative deviations. However, these are very small in count and the margin is relatively insignificant compared to those of positive deviations.

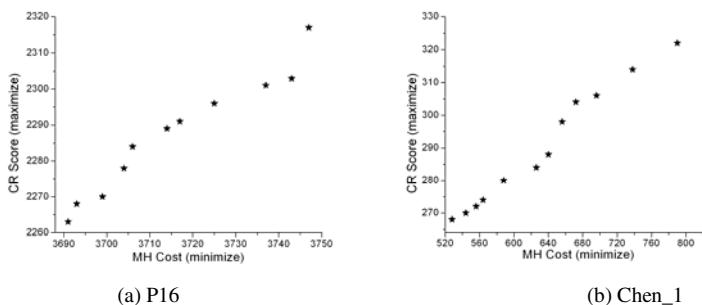
Table 1. Comparison results for Multi-objective DFLP (Lacksonen and Enscore)

Data Set		Prob.	ACO-DML (MH Cost only)	PMDFLP				PMDFLP Deviation (MH Cost)
				MH Cost		CR Score		
Depts.	Periods		Best	Avg	Best	Avg		
3		P1	267	267	344	344	0%	
		P2	260	256	352	352	1.54%	
		P3	363	360	367.36	350	0.83%	
6		P4	299	216	216.8	354	349.2	27.76%
		P5	442	436	441.25	590	582.25	1.35%
5		P6	589	468	472.16	570	566.7	20.54%
		P7	424	425	426.75	590	587.25	-0.24%
		P8	429	455	471.2	576	560.8	-6.06%
		P9	1678	1671	1684.3	1416	1407.5	0.42%
		P10	2023	2004	208.75	1401	1391.9	0.94%
3		P11	1747	1729	1743.12	1387	1360.6	1.03%
		P12	2113	2115	2132.4	1267	1242.5	-0.09%
		P13	3065	3065	3097.3	2360	2309.75	0%
12		P14	3803	3502	3571.8	2298	2231.2	7.91%
		P15	2961	2962	3001.5	2421	2377.2	-0.03%
		P16	3740	3691	3727.6	2317	2286.5	1.31%

Table 2. Comparison results for Multi-objective DFLP (Chen)

Prob.	ACO-DML		PMDFLP		PMDFLP Deviation (MH Cost)	PMDFLP Deviation (CR Score)
	MH	CR	MH	CR		
1	1220	260	528	324	56.72%	24.61%
2	1253	363	630	312	49.72%	-14.05%

MOEAs do not try to find one optimal solution but all the trade-off solutions, and deal with two goals – diversity and convergence. Pareto-optimal solutions of the final generation obtained by the PMDFLP are shown in Fig. 3 to illustrate the convergence and diversity of the layouts. It is worthwhile to mention that in all cases, most of the solutions of the final population are Pareto-optimal. In the figures, the occurrences of the same non-dominated solutions are plotted only once. From these figures, it can be

**Fig. 3.** Final Pareto-optimal layouts

observed that the final solutions produced by the PMDFLP are well spread and converged. For this reason it is capable of finding extreme solutions. Also, it provides a wide range of alternative layout choices for the designers.

Fig. 4 demonstrates the convergence behavior of the proposed methods over generations for MH cost and CR score. From the figures, it can be found that the proposed method is able to optimize both the best and average values for MH cost (minimize) and CR score (maximize) from the first to the last generation very efficiently. It is interesting to observe that the PMDFLP can not only find the best values successfully, but also can obtain very satisfactory average values. This is further justified by the average values mentioned in Table 1.

In Table 3, a comparison with the D+S method [16] is given to show the performance of the PMDFLP as a multi-objective approach by using the data set used in [5]. In existing literature, we found the values for both objectives separately only for the D+S method. However, these data sets are composed of 6 and 8 departments only. Also, the

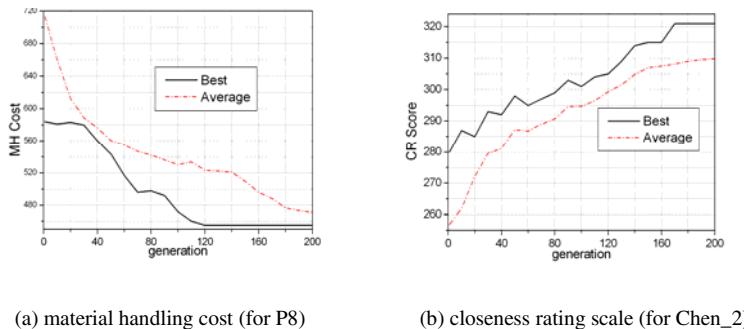


Fig. 4. Two objectives over generations

Table 3. Comparison with D+S [16] for multi-objective FLP

Prob.	Problem size 6 X 6						Problem size 8 X 8													
	W1		W2		D+S		PMDFLP		W1		W2		D+S		PMDFLP					
	MH	CR	MH	CR	MH	CR	MH	CR		MH	CR	MH	CR		MH	CR	MH	CR		
1	0.5-	0.5-	98	27	96	48	1	0.5-	0.5-	211	50	179	82	2	0.5-	0.5-	195	49		
	0.8	0.2						0.8	0.2			0.9	53		0.5-	0.5-	190	49		
	0.9	0.1	98	29				0.9	0.1	211	53	179	82		0.7-	0.3-	190	49		
2	0.5-	0.5-	98	27	96	48		0.5-	0.5-	195	49	179	82		0.6	0.4	194	50		
	0.9	0.1						0.6	0.4						0.7-	0.3-				
	0.7-	0.3-	100	30				0.8	0.2						0.9	0.1				
3	0.5-	0.5-	100	27	96	48		0.7-	0.3-	190	49	179	82		0.6	0.4	194	50		
	0.6	0.4						0.8	0.2						0.9	0.1				
	0.7-	0.3-	100	30				0.9	0.1						0.7-	0.3-				
4	0.5	0.5	92	25	96	48		0.5-	0.5-	195	48	179	82		0.6	0.4	190	49		
	0.6	0.4	100	30				0.7-	0.3-						0.8	0.2				
	0.7-	0.3-	102	30				0.9	0.1						0.9	0.1				

Table 4. Comparison with our static multi-objective FLP [15]

Prob.	MH Cost		CR Score		
	Best	Avg	Best	Avg	
ds6	PMDFLP	96	96.8	48	43.46
	Static	96	96.8	48	43.40
ds8	PMDFLP	179	211.59	82	69.87
	Static	179	209.84	82	70.3
singh6	PMDFLP	94	98.3	56	40.48
	Static	94	98.28	48	40.48
singh8	PMDFLP	179	198.27	82	73.29
	Static	179	199.84	82	73.1
ct9	PMDFLP	4818	4821.63	90	76.04
	Static	4818	4822.9	90	74.79
naug30	PMDFLP	3062	3084.92	292	247.06
	Static	3062	3081.02	292	254.05
sko42	PMDFLP	15798	16894.27	378	328.19
	Static	15796	16876.56	370	325.13
sko72	PMDFLP	66034	67643.42	602	536.8
	Static	66034	67658.33	602	536.8
will100	PMDFLP	273974	280118.54	1084	969.12
	Static	273988	280126.62	1084	977.42

existing results, including the results in [5], are presented as a combined result using weights for both objectives. As indicated by the table, our proposed method clearly dominates the D+S methods for both objectives for these test problems.

To justify the efficiency of the proposed PMDFLP in case of large multi-objective FLP, we also perform experiments for problems composed of 30, 42, 72 and 100 departments and compare it with our previously proposed static multi-objective FLP approach [15]. Again, we have ourselves created the test data sets for CR score where no such data exists, and the solutions are obtained by setting the period to 1 for PMDFLP. The comparison results are given in Table 4. Against these data sets, the PMDFLP is on par or even outperforms our previous static multi-objective approach. In some rare cases, the obtained results are slightly inferior to the previous approach. However, in all cases, the margins are negligible.

7 Conclusion

The dynamic and multi-objective FLPs have emerged as popular topics in recent years. Previous research typically focused either on dynamic or multi-objective aspect of FLP. In addition, objectives have been combined by many different methods and stated as a single objective in solving multi-objective FLP. This paper presents an evolutionary approach for solving the multi-objective DFLP to find a set of Pareto-optimal layouts, which better reflects real-world scenarios. A comparative analysis with the previous studies in the literature shows that the proposed method generates better solutions. Also, it is capable of finding a set of Pareto-optimal layouts that optimizes both MH cost and CR score simultaneously throughout the entire evolutionary process and provides a wide range of alternative layout choices, allowing decision makers to be more flexible and to make better decisions based on market circumstances. Accordingly, it promises to be a useful tool in solving multi-objective DFLP.

References

1. Tompkins, A.: Facilities Planning, 3rd edn. John Wiley & Sons, New York (2003)
2. Singh, S.P., Singh, V.K.: An Improved Heuristic Approach for Multi-Objective Facility Layout Problem. *Int. J. Prod. Res.* 48(4), 1171–1194 (2010)
3. Drira, A., Pierrevval, H., Hajri-Gabouj, S.: Facility Layout Problems: A survey. *Annual Rev. Con.* 31, 255–267 (2007)
4. Grosan, C., Abraham, A., Tigan, S., Chang, T.G.: How to Solve a Multicriterion Problem for Which Pareto Dominance Relationship Cannot Be Applied? A Case Study from Medicine. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) *KES 2006. LNCS (LNAI)*, vol. 4253, pp. 1128–1135. Springer, Heidelberg (2006)
5. Chen, G.Y.-H.: Multi-Objective Evaluation of Dynamic Facility Layout using Ant Colony Optimization. PhD Thesis, The University of Texas at Arlington, USA (2007)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evo. Comp.* 6(2), 182–197 (2002)
7. Pareto Efficiency, http://en.wikipedia.org/wiki/Pareto_efficiency
8. Singh, S.P., Sharma, R.R.K.: A Review of Different Approaches to the Facility Layout Problems. *Int. J. Adv. Manuf. Technol.* 30, 425–433 (2006)

9. Rosenblatt, M.J.: The Dynamics of Plant Layout. *Manag. Sc.* 32(1), 76–86 (1986)
10. Conway, D.G., Venkataraman, M.A.: Genetic Search and the Dynamic Facility Layout Problem. *Comp. & Oper. Res.* 21(8), 955–960 (1994)
11. Balakrishnan, J., Cheng, C.H., Conway, D.G., Lau, C.M.: A Hybrid Genetic Algorithm for the Dynamic Plant Layout Problem. *Int. J. Prod. Econ.* 86, 107–120 (2003)
12. Ripon, K.S.N., Glette, K., Høvin, M., Tørresen, J.: Dynamic Facility Layout Problem with Hybrid Genetic Algorithm. In: 9th IEEE International Conference on Cybernetic Intelligent Systems 2010, University of Reading, UK (2010)
13. Singh, S.P., Sharma, R.R.K.: Genetic Algorithm based Heuristics for the Dynamic Facility Layout Problem. *Eur. J. Manag.* 8(1), 128–134 (2008)
14. Suresh, G., Vinod, V.V., Sahu, S.: A Genetic Algorithm for Facility Layout. *Int. J. Prod. Res.* 33(12), 3411–3423 (1995)
15. Ripon, K.S.N., Glette, K., Mirmotahari, O., Høvin, M., Tørresen, J.: Pareto Optimal Based Evolutionary Approach for Solving Multi-Objective Facility Layout Problem. In: Chan, J.H. (ed.) ICONIP 2009, Part II. LNCS, vol. 5864, pp. 159–168. Springer, Heidelberg (2009)
16. Dutta, K.N., Sahu, S.: A Multigoal Heuristic for Facilities Design Problems: MUGHAL. *Int. J. Prod. Res.* 20(2), 147–154 (1982)

Topological Hierarchical Tree Using Artificial Ants

Mustapha Lebbah and Hanane Azzag

LIPN-UMR 7030
Universit Paris 13 - CNRS
99, av. J-B Clment - F-93430 Villetteuse
`{firstname.secondname}@lipn.univ-paris13.fr`

Abstract. We propose in this paper a new approach for topological hierarchical tree clustering inspired from self-assembly behavior of artificial ants. Our method called THT (Topological Hierarchical Tree) builds, autonomously and simultaneously, a topological and hierarchical partitioning of data. Each "cluster" associated to one cell of a 2D grid is modeled by a tree. The artificial ants that we define dissimilarly build a tree where each ant represents a node/data. The benefit of this novel approach is the intuitive representation of hierarchical relations in the data. This is especially appealing in explorative data mining applications, allowing the inherent structure of the data unfold in highly intuitive fashion.

Keywords: Hierarchical clustering, self-organization, Artificial Ants.

1 Introduction

The accumulation of large structured dataset collected from different sources requires developing a new methods for clustering and visualization in order to understand the informations involved in data. Mining and visualizing these structured dataset represent a real challenge in machine learning. Data clustering and visualization are identified as one of the major problems in data mining. Popularity and different variations of clustering problem, have given birth to several methods [1][5][6]. These methods use both heuristic and mathematics principles. Concerning heuristics methods, the numerous abilities of ants have inspired researchers for more than ten years regarding designing new clustering algorithms [2][3]. The initial and pioneering work in this area is due to Deneubourg and his colleagues [2]. These researchers have been interested in the way real ants sort objects in their nest by carrying them and dropping them in appropriate locations without a centralized control policy [4]. The next step toward data clustering has been performed in [3]. These researchers have adapted the previous idea by assuming that an object is a datum and by tuning picking/dropping probabilities according to the similarities between data. These ants-based algorithms inherit from real ants interesting properties, such as the local/global optimization of the clustering, the absence of need of a priori information on an initial partitioning or number of classes, or parallelism. Furthermore, the results are presented as a visualization : a property which is coherent with an important actual trend in data mining called 'visual data mining' where results are presented in a visual and interactive way to the expert. Many authors have used and extended this model especially in the case of real and more difficult data [5][6][7].

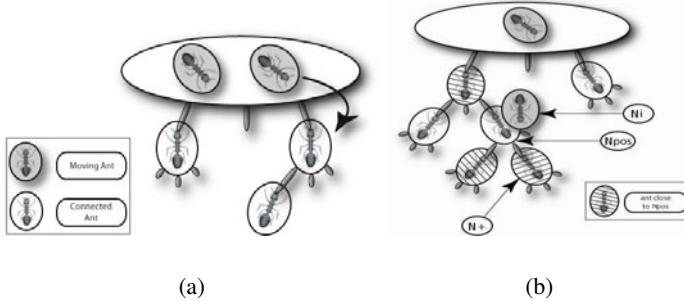


Fig. 1. General principles of tree building with artificial ants (a), and the computation of an ant's neighborhood (b)

In this work we introduce a new method named THT : Topological Hierarchical Tree using artificial ants that provide a hierarchical and topological clustering. Data moves autonomously toward a 2D Grid respecting bio-inspired rules. Where each cell represents a tree structured data. Thus we obtain in one pass : topological clustering and hierarchical clustering (tree in each cell). The topological function of the proposed algorithm is based on Kohonen approach [10] and the rules for building tree are based on artificial ants method named AntTree [8].

2 From Real to Artificial Ants : AntTree Algorithm

The self-assembly behavior of individuals can be observed in several insects like bees or ants. We are interested here in the complex structures which are built by ants [9]. These insects may become fixed to one another to build live structures with different functions. Ants may thus build "chains of ants" in order to fill a gap between two points. These structures disaggregate after a given period of time. From those elements, we define the outlines of AntTree computer model which simulates this behavior for tree building (see figure 1(a)). The n ants N_1, \dots, N_n represent each of the n data x_1, \dots, x_n of the database. Let \mathcal{R}^d be the euclidean data space and $\mathcal{A} = \{x_i; i = 1, \dots, n\}$ a set of observations, where each observation $x_i = (x_i^1, x_i^2, \dots, x_i^d)$ is a continuous vector in \mathcal{R}^d . Ants locally perceive the structure: a moving ant N_i located over a connected ant N_{pos} perceives a neighborhood V_{pos} (see the striped ants in figure 1(b)), which is limited (1) to the (mother) ant to which N_{pos} is connected, and (2) to the (daughter) ants which are connected to N_{pos} . N_i can perceive the similarities between the data x_i it is representing and the data represented by ants of V_{pos} . According to these similarities, N_i may either get connected to N_{pos} , or move to one of the ants in V_{pos} .

At each step, an ant N_i is randomly selected and connects itself or move according to the similarity (euclidean measure) with its neighbors. While there is still a moving ant N_i , we simulate an action for N_i according to its position denoted by N_{pos} (i.e. on the support or on another ant). We also denote N^+ as the ant (daughter) connected to N_{pos} which is the most similar to N_i . As will be seen in the following, when an ant

moves toward another one, this means that it is similar enough to that ant. So N_i will become connected to N_{pos} provided that it is dissimilar enough to ants connected to N_{pos} . N_i will thus form a new sub-category of N_{pos} which is as dissimilar as possible from the other existing sub-categories. For this purpose, let us denote by $T_{Dist}(N_{pos})$ the highest distance value which can be observed among the daughters of N_{pos} . N_i is connected to N_{pos} if and only if the connection of N_i increases further this value. The test that we perform consists of comparing N_i to the most similar ant N^+ . If these two ants are dissimilar enough ($Dist(N_i, N^+) < MaxT_{Dist}(N_{pos})$), then N_i is connected to N_{pos} , else it is moved toward N^+ .

In [8] authors have presented a complete studies of AntTree algorithms and have shown that AntTree algorithms clearly outperformed traditional clustering methods and had a low complexity (near the $n \log(n)$). This is due to the fact that AntTree exploits the tree-structure very well and avoids exploring the whole tree (only sub-trees) when connecting a new ant.

3 Self-organizing Hierarchical Tree

A variety of topological maps algorithms are derived from the first original model proposed by Kohonen. All models are different from each other but share the same idea: depict large datasets on a simple geometric relationship projected on a reduced topology (2D). The model that we propose uses the same grid process, combined with a new concept of neighborhood. Our model THT seek to find an automatic clustering that provides a hierarchical and topological organization of a set of observations \mathcal{A} . This model is presented as regular grid in 2D that has a topological order of p cells. Each cell c is the 'root support' of a sub-tree denoted $Tree_c$ and each node N_{x_i} of the sub-tree represents a data x_i . More precisely the proposed model defines a set of sub-trees projected on a 2D grid called \mathcal{C} .

Taking into account the proximity between two sub-trees on the map \mathcal{C} is a useful information which allows to define a topological neighborhood relation previously used in traditional topological maps. Thus, for each pair of cells c and r on the map, the distance $\delta(c, r)$ is defined as the length of the shortest chain linking cells r and c on the grid associated to sub-trees $Tree_c$ et $Tree_r$. To model the influence of two cell r and c (depending on their proximity), we use a neighborhood function \mathcal{K} . Thus, the mutual influence between two sub-trees $tree_c$ and $tree_r$ which are roots of two cells c and r is defined by the function $\mathcal{K}^T(\delta(c, r))$ where T represents the temperature function that controls the size of the neighborhood. In practice, as for traditional topological map we use smooth function to control the size of the neighborhood as $\mathcal{K}(\delta(c, r)) = \exp\left(\frac{-\delta(c, r)}{T}\right)$.

We also associate to each sub-tree a representative point denoted w_c which is a given data denoted x_i in the sub-tree $tree_c$ ($w_c = x_i \in tree_c$). Choosing a representative point allows easily adapting our algorithm to any type of data (categorical, binary, and mixed data data ... etc). The quality and the objective function of self-organizing trees is written as follows:

$$\mathcal{R}(\phi, \mathcal{W}) = \sum_{c \in \mathcal{C}} \sum_{x_i \in Tree_c} \sum_{r \in C} \mathcal{K}(\delta(\phi(x_i), r)) \|x_i - w_r\|^2 \quad (1)$$

Minimizing the cost function $\mathcal{R}(\phi, \mathcal{W})$ is a combinatorial optimization problem. In practice, we seek to find the best (optimal) solution by using batch version. In this work we propose to minimize the cost function in the same way as "batch" version but using statistical characteristics provided by sub-trees (associated to each cell) to accelerate the convergence of the algorithm. Three necessary basic steps for minimizing the cost function are defined as follows :

- Step of building Tree

In this step we seek to find the best position of a given data x_i in the tree $Tree_c$ associated to cell c . We use connections/disconnections rules inspired from AntTree [8]. The particularity of the obtained tree is that each node N whether it is a leaf or an internal node represents a given data x_i . In this case, N_{x_i} denotes the node that is connected and associated to the data x_i , $N_{x_{pos}}$ represents current node of the tree and $N_{x_{i+}}$ the node connected to $N_{x_{pos}}$, which is the most similar (closest by distance) to N_{x_i} . As it was defined in section 2, $T_{Dist}(N_{x_{pos}})$ is the highest distance value which can be observed among the local neighborhood V_{pos} . x_i is connected to $N_{x_{pos}}$ if and only if the connection of N_{x_i} further increases this value:

$$\begin{aligned} T_{Dist}(N_{x_{pos}}) &= \text{Max}_{j,k} \|N_{x_j} - N_{x_k}\|^2 \\ &= \text{Max}_{j,k} \|x_j - x_k\|^2 \end{aligned} \quad (2)$$

In other words, connections rules consist of comparing a node N_{x_i} to the nearest node $N_{x_{i+}}$. In the case where both nodes are sufficiently far away ($\|N_{x_i} - N_{x_{i+}}\|^2 > T_{Dist}(N_{x_{pos}})$) then the node N_{x_i} is connected to its current position $N_{x_{pos}}$.

Otherwise, the node N_{x_i} associated to data x_i is moved toward the nearest node $N_{x_{i+}}$. Therefore, the value T_{Dist} increases for each node connected to the tree. In fact, each connection of a given data x_i implies a local minimization of the value of the corresponding T_{Dist} . Therefore a minimization of the cost function (II). At the end of the tree construction step, each cell c of the map \mathcal{C} will be associated to a sub-tree $tree_c$. Connections rules are based on Nearest Neighbor approach. Each data will be connected to its nearest neighbor.

- Assignment step

Each datum x_i is connected in the sub-tree $Tree_c$ forming a hierarchical relation noted parent-child. We use $nodeChild(x_i)$ function which provides all child node of a same node parent N_{x_i} associated to the data x_i . At step $t = 0$, $nodeChild(x_i) = x_i$. Assignment step consists of finding for each given data x_i a cell called "Winner" using the assignment function named χ . This cell is also designated as winner cell for all k-nearest-neighbors of x_i . In other words, a complete root sub-tree N_{x_i} is recursively assigned to the winning cell. The assignment function is defined as follows:

$$\chi(nodeChild(x_i)) = \arg \min_r \sum_{c \in C} \mathcal{K}^T(\delta(r, c)) \|x_i - w_c\|^2 \quad (3)$$

- Representation step

Minimizing the cost function $\mathcal{R}(\phi, \mathcal{W})$ with respect to w_c corresponds to finding the point that minimizes all local distances.

$$\mathbf{w}_c = \min_{\mathbf{w}_c \in tree_c} \sum_{\mathbf{x}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \chi(\mathbf{x}_i))) \|\mathbf{x}_i - \mathbf{w}_c\|^2, \\ \forall c \in C \quad (4)$$

The temperature parameter T varies according to the iterations from T_{max} to T_{min} in the same way as in traditional topological maps.

4 Comparatives Results

We have tested and compared the proposed algorithm on several datasets that have been generated with Gaussian and Uniform distributions. Others have been extracted from the machine learning repository [11] and have several difficulties (fuzzy clustering, no relevant feature). Before comparing our numerical results, we present a map visualization of THT. Treemap is a visualization technique introduced in [2]. An important feature of treemaps is that they make very efficient use of display space. Thus it is possible to display large trees with many hierarchical levels in a minimal amount of space (2D). Treemap can be especially helpful when dealing with large clustered tree. Treemaps lend themselves naturally to showing the information encapsulated in the clustering tree. Viewing a tree at some level of abstraction, the viewer is really looking at nodes belonging to some level in the tree. A treemap can displays the whole structure of trees and allow the users to place the current view in context. In the proposed visualization technique each tree of the cell is represented by a treemap. This aims to obtain an automatic organization of treemaps on a 2D map. Figure 2 shows an example of four tree structures with its corresponding treemaps. The positioning of tree nodes in a treemap is a recursive process. The nodes are represented as rectangles of various shapes. First, the children of the root are placed across the display area horizontally. Then, for each node N already displayed, each of N 's children is placed across vertically within N 's display area. This process is repeated, alternating between horizontal and vertical placement until all nodes have been displayed. We note that each rectangle is colored according to the real label of its corresponding node/data. This makes easy a visual comparison of their relevance and homogeneous clusters (see below visual results discussion).

Concerning visual results, to best analyze the obtained result, we have learned for each dataset 1×1 map in order to build a single treemap. Figures 3 and 4 display some example of 1×1 map and 4×4 map. Observing both maps on each dataset, we find that our algorithm provides a map of treemaps which is a multi-divisions of the main treemap. We can see that topological and hierarchical organization of data is more apparent. In order to visualize the coherence between intra-organization of treemaps and the labelling of points, we assign one color to each label. In each figure (3, 4), we distinguish two regions on the map of treemaps that are dedicated to the pure and mixed clusters. In figure 3b we can observe that diagonal from right to left is dedicated to one class (colored in blue) and the treemap positioned in the bottom right is a mixed cluster. We observe in this treemaps, that yellow data is positioned in a lower level on the tree and the yellow classes are situated in the neighborhood. Same remarks concern Tetra dataset. In Figure 4 observing the top right treemap and in the bottom left, we

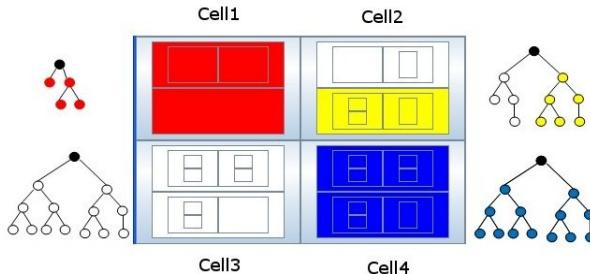


Fig. 2. Forest of trees and its treemaps representation: 2×2 Map of TreeMaps

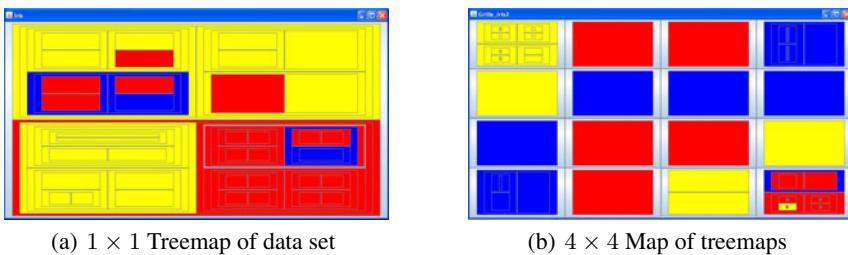


Fig. 3. Iris Dataset

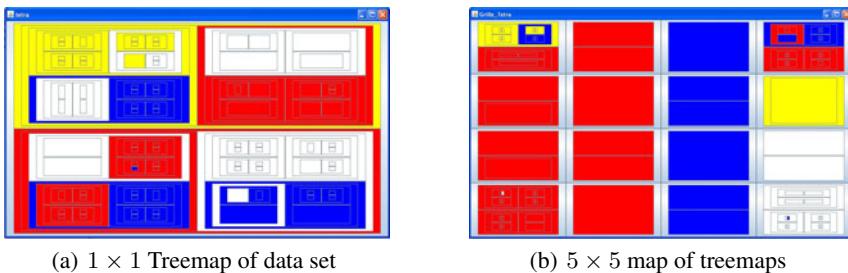


Fig. 4. Tetra Dataset

can conclude on the level and the side where cluster will become mixed. Thus, this visual analysis is done using only 2D visualization unlike SOM method where we can not conclude on which level data is positioned. Concerning numerical results, it is no obvious and no objective to compare our model with algorithm that does not use similar architecture. We remind here that THT model, provides more information than the traditional hierarchical models, K-means or others. In this work we compare the obtained result with SOM model. In this case we adopt the same parameter : map size, initial neighborhood, final neighborhood. Table I reports clustering evaluation criterion obtained with THT and SOM. To evaluate the quality of map clustering, we adopt the approach of comparing results to a "ground truth". We use two criterions for measuring the clustering results. The first one is rand index which measure the percentage of the

Table 1. Competitive results obtained with THT and SOM using the same parameter (map size, initial and final parameter T). *DB* is the davides-bouldin index

Datasets	size.	THT		SOM	
		<i>DB</i>	Rand	<i>DB</i>	Rand
Atom(2)	800	1.4	0.88	1.47	0.51
Anneaux (2)	1000	0.80	0.61	0.90	0.51
ART1(4)	400	0.98	0.81	0.85	0.81
Demi-cercle(2)	600	0.58	0.60	0.67	0.5
Glass(7)	214	1.56	0.70	2	0.65
Hepta(7)	212	0.92	0.92	0.85	0.93
Iris(3)	150	1.06	0.75	1.03	0.75
Lsun(3)	400	0.97	0.71	1.09	0.72
Pima(2)	768	1.09	0.5	2.23	0.43
Target(6)	770	1.4	0.85	1.17	0.58
Tetra(4)	400	0.82	0.81	1.25	0.76
TwoDiamonds(2)	800	0.86	0.60	0.81	0.51

observation pair belonging to the same classes which are assigned to same cluster of the map [13]. The second index is davides bouldin criterion which is used to determine the optimal number K of centroids for K-means [14]. Indeed, as shown in the table [1], we can see that THT method provide similar results and quite comparable to the one obtained by SOM method on the majority of cases. Looking to columns (DB and rand) associated to THT, we can observe that value of DB index is lower using our algorithm and rand index is highest near one for the majority of datasets. Our purpose through this comparison, is not to assert that our method is the best, but to show that THT method can obtain quite the same good results as SOM or other clustering algorithms. Unlike SOM method, THT does not require a posterior processing to analyze the structure of data belonging to clusters (cells). THT also provides simultaneously hierarchical and topological clustering which is more interesting for visualization task.

5 Conclusions et Perspectives

We have developed a new method of hierarchical clustering that provides a local hierarchical clustering of data allowing better visualization of data organization. It generates both 2D self-organization of the trees and a hierarchical organization provided by the constructed tree. Concerning the results we have seen that our approach obtains competitive results on several datasets. The major benefits of our THT approach compared with the standard SOM are the following. First THT uncovers the hierarchical structure of the data allowing the user to understand and analyze large amounts of data. Second, with the various emerging trees at each cell being rather small in size, it is much easier for the user to keep an overview of the various clusters.

As perspectives, the obtained results are preliminary and much work will be done. It is obvious that using trees for data clustering greatly speeds up the learning process, we wish to generalize these algorithms to other kind of structures which may not be trees. The same principles seem to be applicable also to graphs. Finally, it will be necessary

to focus on the visual aspect of our approach. Indeed, we will develop a 2D/3D view of the different trees that result from the hierarchical clustering in order to allow an interactive exploration of data.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Computing Surveys* 31(3), 264–323 (1999)
2. Goss, S., Deneubourg, J.-L.: Harvesting by a group of robots. In: Varela, F., Bourgine, P. (eds.) *Proceedings of the First European Conference on Artificial Life*, pp. 195–204. Elsevier Publishing, Amsterdam (1991)
3. Lumer, E.D., Faieta, B.: Diversity and adaptation in populations of clustering ants. In: Cliff, D., Husbands, P., Meyer, J.A. (eds.) *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pp. 501–508. MIT Press, Cambridge (1994)
4. Deneubourg, J.-L., Goss, S., Franks, N.R., Sendova-Franks, A., Detrain, C., Chretien, L.: The dynamics of collective sorting: robot-like ant and ant-like robots. In: *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pp. 356–365 (1990)
5. Kuntz, P., Snyers, D., Layzell, P.J.: A stochastic heuristic for visualising graph clusters in a bi-dimensional space prior to partitioning. *J. Heuristics* 5(3), 327–351 (1998)
6. Abraham, A., Ramos, V.: Web usage mining using artificial ant colony clustering and linear genetic programming. In: *The Congress on Evolutionary Computation*, Canberra, Australia, December 08–12, pp. 1384–1391. IEEE Press, Los Alamitos (2003)
7. Handl, J., Knowles, J., Dorigo, M.: On the performance of ant-based clustering, pp. 204–213 (2003)
8. Azzag, H., Guinot, C., Oliver, A., Venturini, G.: A hierarchical ant based clustering algorithm and its use in three real-world applications. In: Dullaert, K.S.W., Sevaux, M., Springael, J. (eds.) *European Journal of Operational Research, EJOR* (2006) Special Issue on Applications of Metaheuristics
9. Theraulaz, G., Bonabeau, E., Sauwens, C., Deneubourg, J.-L., Lioni, A., Libert, F., Passera, L., Sol, R.-V.: Model of droplet formation and dynamics in the argentine ant (*linepithema humile mayr*). *Bulletin of Mathematical Biology* (2001)
10. Kohonen, T.: *Self-organizing Maps*. Springer, Berlin (2001)
11. Blake, C.L., Merz, C.L.: UCI repository of machine learning databases. Technical report, University of California, Department of information and Computer science, Irvine, CA (1998) <http://ftp.ics.uci.edu/pub/machine-learning-databases>
12. Schneiderman, B.: Tree visualization with tree-maps: A 2-D space-filling approach. *ACM Transactions on Graphics* 11, 92–99 (1992)
13. Saporta, G., Youness, G.: Concordance entre deux partitions: quelques propositions et expériences. In: *Actes des 8es rencontres de la SFC*, Pointeá Pitre (2001)
14. Davies, D.L., Bouldin, D.W.: A cluster separation measure. *IEEE Transactions on Pattern Recognition and Machine Intelligence* 1(2), 224–227 (1979)
15. Wu, L.H., Hsu, P.Y.: The perceptual eye view: A user-defined method for information visualization, *Human-Computer Interaction. Interaction Platforms and Techniques*, pp. 181–190
16. Mendis, B.S.U., Gedeon, T.D., Botzheim, J., Kóczy, L.T.: Generalised Weighted Relevance Aggregation Operators for Hierarchical Fuzzy Signatures. In: *International Conference on Computational Inteligence for Modelling Control and Automation (CIMCA 2006)*, Sydney, Australia (2006)

Bottom-Up Generative Modeling of Tree-Structured Data

Davide Bacciu¹, Alessio Micheli¹, and Alessandro Sperduti²

¹ Dipartimento di Informatica, Università di Pisa

² Dipartimento di Matematica Pura e Applicata, Università di Padova

Abstract. We introduce a compositional probabilistic model for tree-structured data that defines a bottom-up generative process from the leaves to the root of a tree. Contextual state transitions are introduced from the joint configuration of the children to the parent nodes, allowing hidden states to model the co-occurrence of substructures among the child subtrees. A mixed memory approximation is proposed to factorize the joint transition matrix as a mixture of pairwise transitions. A comparative experimental analysis shows that the proposed approach is able to better model deep structures with respect to top-down approaches.

1 Introduction

Many natural and artificial processes generate data that can be more appropriately modeled using a tree-structured relational representation over a flat encoding based on a vector of numerical features (e.g. phylogenetic trees, XML documents [1], medical data [2]). Dealing with such rich representation entails processing information embedded in a context rather than in isolation. We introduce a *Bottom-Up Hidden Tree Markov Model* (BHTMM, in short) for modeling probability distributions over spaces of trees, that allows learning an hidden generative process for labeled trees which naturally models the dependency between sibling subtrees while routing contextual/structural information to the root node. The *Hidden Recursive Model* (HRM) [3] has long since postulated the opportunity of a bottom-up, or recursive, probabilistic approach, but this has been described only within the scope of a theoretical framework whose realization seemed to be limited to small trees with binary out-degree. In fact, since each node in a tree has at most one parent but a possibly large number of children, the introduction of a bottom-up state transition brings in an explosion of the parameters space whose size grows exponentially with the nodes' outdegree. This has, so far, prevented the development of *practical* bottom-up Markov models for trees, whereas in the neural networks community [34], the bottom-up approach is the prominent model for non-flat data. This is motivated by the inherently recursive nature of hierarchical information, which can be effectively captured by taking a compositional approach that first discovers the less articulated substructures at the bottom of the tree, before tackling with the complexity of deep subtrees. The proposed BHTMM efficiently addresses

such joint state transition problem by resorting to an approach known as *mixed memory* approximation [5] or *switching parent* model. This approximation allows factorizing large state spaces into a mixture of *simpler* distributions and, as such, can be used to break the joint children-to-parent transition matrix into a mixture of pairwise child-to-parent state transition probabilities.

2 Bottom-Up Hidden Tree Markov Model

A rooted tree \mathbf{y}^n is an acyclic directed graph consisting of a set of nodes \mathcal{U}_n , each characterized by an observed label y_u (see Fig. ⑩). Superscript n is used to indicate that \mathbf{y}^n is the n -th sample: bold and plain notation denote structured and scalar terms, respectively. We assume *positional* trees, i.e. where a node's position with respect to its siblings is relevant for information representation, with a finite maximum outdegree L . An observed tree is modeled by a generative process defined by a set of hidden state variables $Q_u \in [1, \dots, C]$, following the same indexing as the observed nodes u , which are responsible for the emission of the observed node labels y_u . Within a *bottom-up* approach, the tree structure is processed from the leaves to the root of the tree, corresponding to a generative process where hidden state transitions are performed from the joint configuration of the child nodes to the corresponding parents.

Fig. ⑩ depicts the Dynamic Bayesian Network (DBN) showing the children-to-parent causal relationships for the generative bottom-up model (denoted as BHTMM in figure). The BHTMM log-likelihood is obtained by marginalizing the hidden state associations $Q_1 = j_1, \dots, Q_{\mathcal{U}_n} = j_{\mathcal{U}_n}$ for the N sample trees

$$\begin{aligned} \log \mathcal{L}_c = & \log \prod_{n=1}^N \prod_{\substack{u' \in \\ \text{leaf}(\mathbf{y}^n)}} \prod_{i=1}^C \{P(Q_{u'} = i) \times P(y_{u'} | Q_{u'} = i)\}^{z_{u'i}^n} \\ & \times \prod_{u \in \mathcal{U}_n \setminus \text{leaf}(\mathbf{y}^n)} \prod_{i=1}^C \{P(y_u | Q_u = i)\}^{z_{ui}^n} \left\{ P(Q_u = i | \bigcap_{l=1}^L Q_{ch_l(u)} = j_l) \right\}^{z_{ui}^n \prod_{l=1}^L z_{ch_l(u)}^n j_l}. \end{aligned} \quad (1)$$

In practice, the sum-marginalization is rewritten as a more tractable product over state assignments, when maximizing the log-likelihood *completed* by hidden indicator variables z_{ui}^n that denote which state i is responsible for the generation of node u . The expression in ⑩ results from the factorization of the joint distribution $P(\mathbf{y}^n, Q_1, \dots, Q_{\mathcal{U}_n})$ following the conditional independence relationships described by the DBN in Fig. ⑩: i.e. each node is conditionally independent of the rest of the tree when the joint hidden state of its direct descendants in the tree is observed. The term $\text{leaf}(\mathbf{y}^n)$ denotes the set of leaves in tree \mathbf{y}^n , $ch_l(u)$ identifies the l -th child of node u and $P(Q_u | \bigcap_{l=1}^L Q_{ch_l(u)})$ is a short form for the children-dependent state transition probability. For the purpose of this paper, we assume strict positional stationarity: hence, we define a null state i_\emptyset that is used to model the absence of the l -th child, i.e. $Q_{ch_l(u)} = i_\emptyset$. The model

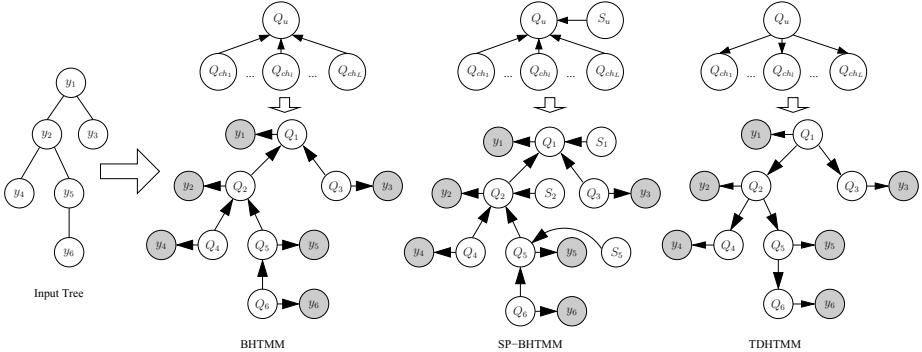


Fig. 1. Conditional dependence assumptions (top) for a BHTMM, SP-BHTMM and TDHTMM, with the corresponding encoding networks (bottom) for an input tree

described by (II) becomes computationally impractical for trees other than binary due to the size of the joint conditional transition distribution, that is order of C^{L+1} . A similar combinatorial problem is observed in factorial HMMs, where a *Mixed Memory* approximation [5] is used to represent the joint L -th order matrix as a convex combination of L first order transitions. The basic idea is to introduce a selector variable S_u , called *switching parent*, that determines the responsibility of the single chains of a factorial HMM to state transition. Here, we seek an approximation to the joint children-parent transition $P(Q_u | \bigcap_{l=1}^L Q_{ch_l(u)})$. We introduce a switching parent $S_u \in \{1, \dots, L\}$ such that $P(Q_u | S_u = l, \bigcap_{l=1}^L Q_{ch_l(u)}) = P(Q_u | Q_{ch_l(u)})$: applying marginalizing yields $P(Q_u | \bigcap_{l=1}^L Q_{ch_l(u)}) = \sum_{l=1}^L P(S_u = l)P(Q_u | Q_{ch_l(u)})$ where we have used the assumption that S_u is independent from each $Q_{ch_l(u)}$. This equation states that the joint transition distribution can be approximated as a mixture of L pairwise distributions where the influence of the l -th children on a state transition to node u is determined by the weight $\varphi_l = P(S_u = l)$. The corresponding graphical model is denoted as SP-BHTMM in Fig. II. The complete log-likelihood of the SP-BHTMM can be obtained by substituting the switching parent formulation for $P(Q_u | \bigcap_{l=1}^L Q_{ch_l(u)})$ into (II) while introducing indicator variables t_{ul}^n for the switching parent, yielding

$$\begin{aligned} \log \mathcal{L}_c = & \log \prod_{n=1}^N \prod_{\substack{u' \in \\ \text{leaf}(\mathbf{y}^n)}} \prod_{i=1}^C \{\pi_i b_i(y_{u'})\}^{z_{u'}^n} \\ & \times \prod_{u \in \mathcal{U}_n \setminus \text{leaf}(\mathbf{y}^n)} \prod_{i,j=1}^C \prod_{l=1}^L \{b_i(y_u)\}^{z_{ui}^n} \{\varphi_l A_{i,j}\}^{z_{ui}^n t_{ul}^n z_{ch_l(u)j}^n} \end{aligned} \quad (2)$$

where $\pi_i = P(Q_{u'} = i)$ is the leaves *prior distribution*, $A_{i,j} = P(Q_u = i | Q_{ch_l(u)} = j)$ is the state transition probability and $b_i(y_u) = P(y_u | Q_u = i)$ is the *emission model*. The actual parametrization of the SP-BHTMM model depends on the

Table 1. Reversed Upwards-Downwards and Viterbi Equations

Upward Recursion (Root and internal nodes)	
$\beta_u(i) = \frac{b_i(y_u) \sum_{l=1}^L \varphi_l \beta_{u, ch_l(u)}(i) P(Q_u = i)}{\sum_{j=1}^C \sum_{l=1}^L b_j(y_u) \varphi_l \beta_{u, ch_l(u)}(j) P(Q_u = j)}$	$\beta_{u, ch_l(u)}(i) = \frac{\sum_{j=1}^C A_{i,j}^l \beta_{ch_l(u)}(j)}{p(Q_u = i)}$
Downward Recursion (Internal and leaf nodes)	
$\epsilon_{u, ch_l(u)}^l(i, j) = \frac{\epsilon_u(i) \beta_{ch_l(u)}(j) \varphi_l A_{i,j}^l}{p(Q_u = i) \sum_{l''=1}^L \varphi_{l''} \beta_{u, ch_{l''}(u)}(i)}$	$\epsilon_u(i) = \sum_{k=1}^C \epsilon_{pa(u), u}^{pos(u)}(k, i)$
Exact Viterbi (Root and internal nodes)	
$\delta_u(i) = \max_{i_1, \dots, i_L} \left\{ \sum_{l=1}^L \varphi_l A_{i,i_l}^l \prod_{v=1}^L b_{i_v}(y_v) \prod_{\substack{v' \in \\ \text{CH}(u)}} \delta_{v'}(i'_v) \right\}$	

stationarity assumptions that are taken: here, we choose a positional probability $A_{i,j}^l$ (i.e. state transition depends on the position l in the subtree) and a switching distribution $\varphi_l = P(S_u = l)$ shared among the hidden states. For such a parametrization to be consistent, also the leaves prior π_i^l has to be positional. Given a maximum out degree L , the resulting model requires $O(LC^2 + L)$ independent parameters, which is far more feasible than the $O(C^{L+1})$ parameters of the *exact* joint transition model.

The SP-BHTMM model is fit by applying Expectation-Maximization to the complete log-likelihood. The introduction of children-to-parent causal relationships generates sibling dependencies in the moral graph, which falsify the assumptions upon which the *standard* algorithms for top-down learning and inference are built [6]. To realize the mixed-memory SP-BHTMM model we derive novel and efficient procedures for parameter learning and inference that account for such dependence inversion, introducing the *reversed upwards-downwards* and the *reversed Viterbi* algorithms. Due to space limitations, we briefly discuss only the resulting update equations that are summarized in Table 1; the reader is referred to the technical report¹ for a detailed derivation. The model parameters (i.e. π_i^l , $A_{i,j}^l$, φ and the parameters of the emission model b_i) are estimated based on the posterior of the indicator variables $\bar{z}_{uijl} = z_{ui}^n t_{ul}^n z_{ch_l(u)j}^n$ stating that node u is in state i while its l -th child is in state j , that is the conditional expected value $\epsilon_{u, ch_l(u)}^l(i, j) = p(Q_u = i, Q_{ch_l(u)} = j, S_u = l | \mathbf{y}^n)$. This posterior is estimated at the E-step by message passing on the structure of the nodes' dependency graph, by exploiting the following marginalized posterior decomposition $\epsilon_u(i) = \frac{p(\mathbf{y}_{1 \setminus u} | Q_u = i)}{p(\mathbf{y}_{1 \setminus u} | \mathbf{y}_u)} p(Q_u = i | \mathbf{y}_u) = \alpha_u(i) \beta_u(i)$, where $\beta_u(i)$ and $\alpha_u(i)$ are the *upward* and *downward probabilities*, respectively. The term \mathbf{y}_u identifies the observed subtree rooted at node u , while $\mathbf{y}_{1 \setminus u}$ denotes the observed tree (i.e. rooted at 1) without the \mathbf{y}_u subtree. Upward probabilities can be computed for the *leaf*

¹ <http://compass2.di.unipi.it/TR/Files/TR-10-08.pdf.gz>

nodes y_u as $\beta_u(i) = \frac{b_i(y_u)\pi_i^{pos(u)}}{\sum_{j=1}^C b_j(y_u)\pi_j^{pos(u)}}$ while the $\beta_u(i)$ values for the *internal* and *root* nodes u are computed as described in Table II. The conditional independence of a node's emission, given the realization of its hidden state, has been used to isolate $b_i(y_i)$, while the switching parents approximation has been exploited to factorize the probability of the hidden state given its subtree. In order to compute $\beta_u(i)$ in the bottom-up pass, we also need to compute and propagate the *auxiliary upward parameters* $\beta_{u, ch_l(u)}(i)$ for each child l as detailed in Table II. As a by-product of the upwards recursion, we can also compute the log-likelihood as the sum of the logs of the denominators of $\beta_u(i)$. As for the downward recursion, rather than computing the $\alpha_u(i)$ values, we can straightforwardly compute and propagate downwards the *smoothed probabilities* $\epsilon_u(i)$. The basis of the recursion is at the root node Q_1 , where $\epsilon_1(i) = p(Q_1 = i | \mathbf{y}_1) = \beta_1(i)$, while the recursive step is computed as shown in Table II ($pos(u)$ denotes the position of node u with respect to its siblings). Demonstrating the factorization of $\epsilon_{u, ch_l(u)}^l(i, j)$ into known probabilities requires a long derivation that cannot be shown due to space limitations. Briefly, the factorization of the pairwise smoothed posterior is based on a Bayesian decomposition of the posterior, followed by the repeated application of the switching parents hypothesis.

The *reversed Viterbi* algorithm determines the most likely hidden states assignment $\mathbf{Q}_1 = \mathbf{x}$ for a given observed tree \mathbf{y}_1 or, equivalently, $\max_{\mathbf{x}} P(\mathbf{y}_1, \mathbf{Q}_1 = \mathbf{x}) = \max_i \left\{ \delta_u(i) \max_{\mathbf{x}_{1 \setminus u}} \left\{ P(\mathbf{y}_{1 \setminus \mathbf{CH}(u)}, \mathbf{Q}_{1 \setminus u} = \mathbf{x}_{1 \setminus u} | Q_u = i) \right\} \right\}$, where the term on the right has been obtained by straightforward application of Bayes theorem on a node u in the tree. The term $\mathbf{CH}(u)$ is the set of children of node u , $\mathbf{y}_{\mathbf{CH}(u)}$ denotes the observed subtrees rooted at each of the child of u , while $\mathbf{y}_{1 \setminus \mathbf{CH}(u)}$ is the observed tree without the child subtrees of node u . The maximization is computed by an upward recursion initialized for each leaf node u' as $\delta_{u'}(i) = \max_{\mathbf{x}_{\mathbf{CH}(u')}} P(Q'_u = i) = \pi_i$, that is equivalent to the prior distribution. For each internal node taken upwards, we use the $\delta_u(i)$ definition in Table II. Such factorization follows from the fact that couples of observed and hidden subtrees are independent from each other when the parent (y_u, Q_u) is not given (formal proof can be obtained by repeated Bayesian decomposition, showing that the Markov blanket of an observed subtree \mathbf{y}_v includes only the corresponding hidden states \mathbf{Q}_v). The $\delta_u(i)$ recursion ends at the root node: at this point, the root can determine its hidden state assignment i_1^* that maximizes the joint probability. The maximization step in the $\delta_u(i)$ recursion requires computing $O(C^L)$ factors for each internal node in the tree, introducing severe feasibility issues when dealing with trees with large out-degree. To this end, we introduce an *approximated Viterbi* algorithm that deals with the joint maximization in $\delta_u(i)$ by treating the contribution of each child node in separation, i.e. allowing a parent node u to separately maximize $\max_{i_l} \left\{ \varphi_l A_{i_l, i_l}^l b_{i_l}(y_{ch_l(u)}) \delta_{ch_l(u)}(i_l) \right\}$ for each children l , using these estimated hidden states to compute $\delta_u(i)$. Such an approximation allows to consistently reduce the memory and computational requirements, that is to $O(L)$ and $O(LC)$, respectively.

The authors couldn't find in literature any BHTMM beside the theoretical model described as part of the HRM framework in [3]. Alternative bottom-up approaches

have been proposed in the context of *Probabilistic Context Free Grammars* (PCFG) and tree automata [7], but their use is limited to binary or ternary trees and to applications such that exist a set of *pre-defined* rewriting rules for the grammar (i.e. natural language processing). The complexity of the bottom-up approaches has motivated the diffusion of the less computationally-intensive *top-down* hidden Markov models (TDHTMM) [8,9], where the generative process flows from the root to the leaves of the tree (see the rightmost model in Fig. II). TDHTMM assumes a first order Markovian dependency such that the current state of a node $ch_l(u)$ depends solely on that of its parent u , entailing that TDHTMM cannot model any form of dependency between siblings. Further, since context is propagated top-down, the root state assignment depends solely on the prior distribution, whereas commonsense suggests that the root is the best node where to convey contextual information concerning the whole tree structure (as in bottom-up models).

3 Experimental Evaluation

The experimental evaluation compares the performance of SP-BHTMM and TDHTMM on synthetic and real datasets used in tree classification tasks (see Table 2 for a summary of the datasets statistics). Separate bottom-up and top-down models are trained for each class, while the Viterbi algorithm determines the maximum likelihood labeling of the fresh test structures. The intended aim is not to show that HTMM models can compete with state-of-the-art discriminative classifiers for structured data, rather, class labels are used within a generative setup to evaluate the ability of SP-BHTMM to capture and model structural information more effectively than a TDHTMM. To this aim we exploit well known measures of clustering quality such as *purity* and *entropy*, where the former measures the degree to which a model generates structures from the same class, while the latter measure the amount of uncertainty and disorder.

The first experiment targets at evaluating the effect of tree depth on a synthetic two-class data set, generated using proprietary software, comprising sequences, represented as binary trees with only left children, and *proper* binary trees (not complete). For each depth $T \in [4, 12]$, we have sampled 5 training and test sets, each comprising 150 sequences and 150 binary trees. Training has been performed for SP-BHTMM and TDHTMM comprising 2 hidden states and using single Gaussian emissions. Figure 2 shows the average purity and entropy as a function of tree depth, obtained both for the exact TDHTMM and SP-BHTMM Viterbi as well as for the approximated bottom-up Viterbi. The plot shows a marked purity deterioration and increased model disorder in TDHTMM

Table 2. Statistics computed for the datasets used in the experimentation

DS	outdegree	max depth	min # nodes	max # nodes	# train trees	# test trees	# classes
Artificial	2	12	2	1513	12197	12164	2
INEX 2005	32	4	4	87	4820	4811	11
INEX 2006	66	8	3	115	6053	6064	18

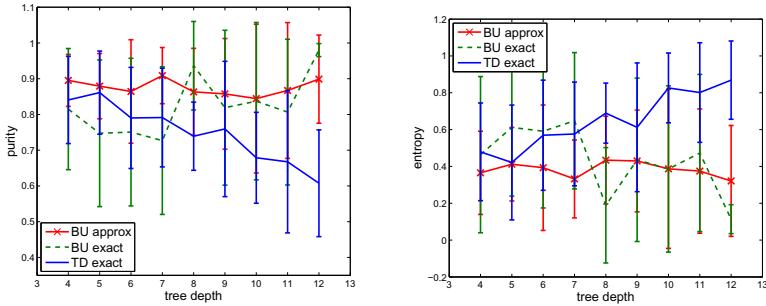


Fig. 2. Results for the Artificial dataset

as tree depth increases, while SP-BHTMM seems to be able to efficiently cope with deep trees, with the approximated Viterbi inference maintaining a superior overall performance also on shallow trees. As tree depth grows, accuracy for the exact Viterbi increases as a longer sequence of approximated state assignments is more likely to produce sub-optimal solutions. Conversely, on shallow trees, it shows an higher uncertainty due to the multiplication of small probability values that makes maximization less reliable.

In the second part, we evaluate the effect of a large outdegree on the two approaches using two data sets from the 2005 and 2006 INEX Competition [1], comprising XML formatted documents represented as trees. The large number of classes in both data sets makes them challenging benchmarks, such that the random classifier baseline for INEX 2005 and INEX 2006 is 9% and 5.5%, respectively. For this task, performance has been evaluated for varying configurations of the models, comprising 2 to 10 hidden states. Training and test trials have been repeated 5 times for each model configuration, and the results averaged. INEX trees are shallow structures which, following previous results, should not affect consistently the TDHTMM performance. These trees are characterized by a large out-degree that prevents the exact bottom-up Viterbi from being applicable already with 2 hidden states, hence results only relate to the approximated Viterbi. Table 3 shows the average purity and entropy of the two models as a function of the hidden state number (average deviation is $\sim 1\%$). In INEX 2005, SP-BHTMM yields to markedly *purer* models as the hidden space size increases, while consistently reducing the model confusion. SP-BHTMM maintains a (less marked but significative) performance advantage also on INEX 2006, achieving better purity and entropy already with $C = 2$ hidden states. The INEX 2006 dataset is a hard benchmark where also discriminative kernel approaches yield to poor classification accuracies, e.g. between 30% and 40%. In particular, INEX 2006 trees have similar label distributions, so that structural information needs to be effectively exploited to be able to separate the 18 classes. In general, results point out that SP-BHTMM is capable of effectively exploiting larger hidden state spaces to capture informative substructures that allow a cleaner and more robust modeling of different structures.

Table 3. Average purity and entropy for INEX 2005 and 2006

Hidden states	INEX 2005				INEX 2006			
	Purity		Entropy		Purity		Entropy	
	TD	BU	TD	BU	TD	BU	TD	BU
C=2	75.81	76.89	19.83	19.26	29.88	32.10	68.21	66.43
C=4	80.99	79.60	16.60	17.60	29.37	30.68	68.40	67.88
C=6	78.52	82.71	18.40	15.54	30.83	31.70	67.55	66.67
C=8	79.73	84.89	16.92	13.99	29.50	30.50	67.64	67.29
C=10	79.26	86.97	17.39	12.65	29.03	31.07	68.30	66.91

4 Conclusion

We have introduced a scalable BHTMM that addresses the computational feasibility problem introduced by the joint children-to-parent state transition by resorting to an approximation based on a finite mixture of multinomials. We have provided efficient learning and inference procedures whose per-node complexity remains bounded by $O(LC^2 + L)$ for the reversed upwards-downwards learning, and by $O(LC)$ for the approximated Viterbi. Experimental results suggest that a bottom-up approach leads to a more efficient processing of the contextual information in the subtrees, which allows for a better propagation of structural information, in particular when dealing with deep trees. Bottom-up processing has an inherent compositional nature, as its hidden states capture structural information according to the recursive nature of hierarchical data, allowing for the development of compositional probabilistic models for the topographic mapping of tree data [10]. An interesting future application concerns the use of BHTMM to learn adaptive metrics for efficiently confronting fuzzy signatures [2] that model data from different real-world domains by fuzzy hierarchical structures.

References

1. Denoyer, L., Gallinari, P.: Report on the XML mining track at INEX 2005 and INEX 2006: categorization and clustering of XML documents. *SIGIR Forum* 41(1), 79–90 (2007)
2. Mendis, B., Gedeon, T.: Polymorphic fuzzy signatures. In: Proc. of the 2010 Int. Conf. on Fuzzy Systems, IEEE, Los Alamitos (2010)
3. Frasconi, P., Gori, M., Sperduti, A.: A general framework for adaptive processing of data structures. *IEEE Trans. Neural Netw.* 9(5), 768–786 (1998)
4. Hammer, B., Micheli, A., Sperduti, A., Strickert, M.: A general framework for unsupervised processing of structured data. *Neurocomputing* 57, 3–35 (2004)
5. Saul, L.K., Jordan, M.I.: Mixed memory markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Mach. Learn.* 37(1), 75–87 (1999)
6. Durand, J., Goncalves, P., Guedon, Y., Rhone-Alpes, I., Montbonnot, F.: Computational methods for hidden Markov tree models—an application to wavelet trees. *IEEE Trans. Signal Process.* 52(9), 2551–2560 (2004)
7. Verdu-Mas, J.L., Carrasco, R.C., Calera-Rubio, J.: Parsing with probabilistic strictly locally testable tree languages. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(7) (2005)

8. Crouse, M., Nowak, R., Baraniuk, R.: Wavelet-based statistical signal-processing using hidden markov-models. *IEEE Trans. Signal Process.* 46(4), 886–902 (1998)
9. Diligenti, M., Frasconi, P., Gori, M.: Hidden tree markov models for document image classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 25(4), 519–523 (2003)
10. Bacciu, D., Micheli, A., Sperduti, A.: Compositional generative mapping of structured data. In: Proc. of the 2010 Int. Joint Conf. on Neural Netw. IEEE, Los Alamitos (2010)

Exploit of Online Social Networks with Community-Based Graph Semi-Supervised Learning

Mingzhen Mo and Irwin King

Department of Computer Science and Engineering,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
`{mzmo, king}@cse.cuhk.edu.hk`

Abstract. With the rapid growth of the Internet, more and more people interact with their friends in online social networks like Facebook¹. Currently, the privacy issue of online social networks becomes a hot and dynamic research topic. Though some privacy protecting strategies are implemented, they are not stringent enough. Recently, Semi-Supervised Learning (SSL), which has the advantage of utilizing the unlabeled data to achieve better performance, attracts much attention from the web research community. By utilizing a large number of unlabeled data from websites, SSL can effectively infer hidden or sensitive information on the Internet. Furthermore, graph-based SSL is much more suitable for modeling real-world objects with graph characteristics, like online social networks. Thus, we propose a novel Community-based Graph (CG) SSL model that can be applied to exploit security issues in online social networks, then provide two consistent algorithms satisfying distinct needs. In order to evaluate the effectiveness of this model, we conduct a series of experiments on a synthetic data and two real-world data from StudiVZ² and Facebook. Experimental results demonstrate that our approach can more accurately and confidently predict sensitive information of online users, comparing to previous models.

Keywords: privacy issue, social network, graph-based semi-supervised learning, community consistency.

1 Introduction

Currently, online social networks are becoming increasingly popular. For example, Facebook currently is utilized by more than 400 million active users and more than 500 billion minutes are spent on it everyday [1]. In these online social networks, people can form social links with others through making friends or joining groups with similar contents.

The security issue of online social networks turns into one of the hot topics, because it affects hundreds of millions users. Online social networks allow people to enable privacy restriction on their profiles. Nevertheless, the friendship and group membership are still visible to the public directly or indirectly. In other words, the public friendship or group information, which online social networks claim to be safe, becomes the

¹ <http://www.facebook.com>

² <http://www.studivz.net>

potential threat to users' privacy. [5][6][10] demonstrate that this information can leak a large quantity of sensitive information.

Recently, Semi-Supervised Learning (SSL) has become a useful technique to exploit unknown information. Compared to supervised learning, SSL has the advantage of avoiding high cost in labeling training data by utilizing large amount of unlabeled data. Thus, SSL can be applied on predicting or learning knowledge from the websites which contain massive unlabeled data, e.g., hidden or sensitive information.

As a technique to exploit hidden information, SSL suits well with the scenario that online social networks contain little public information and a large number of hidden ones [7]. In SSL learning model, the public information can be considered as labeled data and that hidden as unlabeled data. According to the statistics, on average 70% users in Facebook have incomplete profiles. It illustrates that labeled data are far fewer than the unlabeled data.

Especially, graph-based SSL further fits well the online social networks with graph structures. First, graph-based SSL is good at modeling objects with graph structures, in which relationship information is easily expressed by edges and their weights. Second, the learning procedure of graph-based SSL is spreading known information to unknown area to predict the result. That is very similar to the cases in the real world, e.g., online social networks: we expand our networks from existing friends to unacquainted persons and from familiar groups to strange communities. Hence, graph-based SSL is rather suitable for exploiting online social networks.

This paper proposes a novel graph-based SSL model with community consistency. There are several graph-based learning models were proposed before, e.g., basic graph learning with harmonic function [12], which mainly considers the local consistency, and Local and Global Consistency (LGC) graph leaning [11]. Now, we propose a novel graph learning model considering not only local consistency and global consistency but also community consistency. The relationship between this model and the previous ones is shown in Fig. 1.

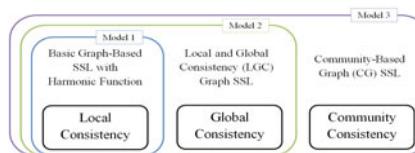


Fig. 1. The Relationship of Three Graph-based SSL Models

This novel SSL exploit model is evaluated on a synthetic dataset 'TwoMoons' and two real-world datasets from StudiVZ and Facebook, comparing with two previous graph-based SSL models and a Supervised Learning model. The evaluation criterion contains accuracy and weighted accuracy, which is defined to measure the confidence of predictions.

The contributions of this paper include the following:

- **A graph-based semi-supervised learning with community consistency is firstly proposed.** With the additional consistency in the objective, this learning model describes the real world more accurately and achieves better learning results.
- **This paper provides two algorithms for the Community-based Graph (CG) SSL exploit model: a closed form algorithm and an iterative algorithm.** The closed form algorithm has a very simple formula to obtain the prediction result, while the iterative algorithm could deal with large-scale datasets.

2 CG SSL Exploit Model and Algorithms

2.1 CG SSL Exploit Model

Preparation. Similar to [7], we define a social network as an undirected graph $G(V, E)$. In $G(V, E)$, every vertex (user) has feature vector $P_i = (p_i^1, p_i^2, \dots, p_i^{n_f})$ and every edge (relationship) has weighted value $W_{i,j} = (w_{i,j}^{fd}, w_{i,j}^{gp}, w_{i,j}^{nk})$. n_f is the total number of features; $w_{i,j}^{fd}$ is a weight for friendship, $w_{i,j}^{gp}$ for group membership, $w_{i,j}^{nk}$ for network relationship and $0 \leq P_i, W_{i,j} \leq 1$. In the whole graph, there are l vertices labeled as \bar{Y}_{label} and u vertices needed to predict their labels $\hat{Y}_{unlabel}$. So our objective is to let the prediction result agree with the true labels $\bar{Y}_{unlabel}$.

Definition 1 (Community). We define a community as a group of users $V_c \in V$, who have strong connection with other users in one or more groups and networks in an online social network. They may not be friends, even have not similar profiles.

According to the definition, we prepare community data. First, we construct all the communities according to the network and group information in online social networks (details in Section 2.3). Then, we can express all communities in a weight matrix. In a community $C_i, i \in N^+$, there are n_i^c members strongly connecting with each other, $v_{j_1}, v_{j_2}, \dots, v_{j_{n_i^c}}$. We could express their relationship in a $(l+u) \times (l+u)$ matrix W_i^c . Then, for all communities, $C_1, C_2, \dots, C_i, \dots, C_{n_c}$, we have a separate matrix for each of them, $W_1^c, W_2^c, \dots, W_i^c, \dots, W_{n_c}^c$. The community weight matrix is $W^c = \sum_{i=1}^{n_c} W_i^c$, where n_c is the total number of communities in the data sample.

Model Building Up. In this part, we show the process of building up the new graph-based SSL exploit model step by step. First we construct the local data $W^g = (1 - \gamma)W^p + \gamma W^{fd}$, where W^p is a similarity matrix of personal information, W^{fd} for friendship and $0 < \gamma < 1$. Then, let $D_{ii}^g = \sum_{j=1}^{l+u} W^g(i, j), i \in \{1, \dots, l, l+1, \dots, l+u\}$ and D^g be the $(l+u) \times (l+u)$ diagonal matrix by placing D_{ii}^g on the diagonal. Now the unnormalized Laplacian matrix L^g is defined as $L^g = D^g - W^g$. Similarly, L^c for community information is constructed from W^c and D^c . Finally, based on the Local and Global Consistency (LGC) graph-based learning [11], we add the constrain of communities and formulate the problem as

$$\min_{\hat{Y} \in Y_{label}^{l+u}} \text{tr}\{\hat{Y}^\top L^g \hat{Y} + \mu_1(\hat{Y} - \bar{Y})^\top (\hat{Y} - \bar{Y}) + \mu_2 \hat{Y}^\top L^c \hat{Y}\}, \quad (1)$$

where the predicted result $\hat{Y} = (\bar{Y}_{label}, \bar{Y}_{unlabel})^\top$, real label $\bar{Y} = (\bar{Y}_{label}, \bar{Y}_{unlabel})^\top$ and $\mu_1, \mu_2 > 0$. $Y_{label} = \{0, 1\}^{n_{label}}$, where n_{label} is the number of different labels. With this step, we have built up a complete CG SSL exploit model to solve the problem.

2.2 Algorithms

In this section, we propose two methods to solve the optimization problem we have formulated before. The first one is a closed form algorithm. Utilizing this method, the exact final result can be obtained directly. The other one is an iterative algorithm, by which we could compute an approximate result. This would be a time-consuming method, but it is able to deal with large-scale datasets.

To simplify the problem, we relax it and solve it. By the definition of this model in Eq. (1), we realize that this is an integer programming problem, which is hard to solve in the consideration of computational complexity. Thus, we relax the feasible region from discrete $\{0, 1\}^{(l+u) \times n_{label}}$ to continuous $\{[0, 1]\}^{(l+u) \times n_{label}}$.

Closed Form Algorithm. Here we first develop a regularization framework for the optimization problem formulated before. Rewriting the objective function associated with F replacing \hat{Y} in Eq. (1), $F \in \{[0, 1]\}^{(l+u) \times n_{label}}$, we have

$$\mathcal{Q}(F) = \frac{1}{2} \left(\sum_{i,j=1}^{l+u} W_{ij}^g \left\| \frac{1}{\sqrt{D_{ii}^g}} F_i - \frac{1}{\sqrt{D_{jj}^g}} F_j \right\|^2 + \mu_1 \sum_{i=1}^{l+u} \|F_i - y_i\| + \mu_2 \sum_{i,j=1}^{l+u} W_{ij}^c \left\| \frac{1}{\sqrt{D_{ii}^c}} F_i - \frac{1}{\sqrt{D_{jj}^c}} F_j \right\|^2 \right), \quad (2)$$

where μ_1, μ_2 are regularization parameters and $\mu_1, \mu_2 \geq 0$. Here the first term (local consistency) and the third term (community consistency) is normalized with $\sqrt{D_{ii}^g}$ and $\sqrt{D_{ii}^c}$. $\frac{1}{2}$ is for the convenience of differentiation and does not affect the classification result. By mathematical deriving, the optimal solution is $F^* = (1 - \alpha - \beta)(I - \alpha S - \beta C)^{-1} Y$, where $S = D^{g^{-1/2}} W^g D^{g^{-1/2}}$, $C = D^{c^{-1/2}} W^c D^{c^{-1/2}}$, $Y = (\bar{Y}_{label}, 0)$ and $\alpha = \frac{1}{1+\mu_1+\mu_2}$, $\beta = \frac{\mu_2}{1+\mu_1+\mu_2}$.

We need to design a strategy to make a final decision from F^* . Because we relax the problem before we solve it, the answer F^* is only the probability of unlabeled data belonging to labels, instead of the final result. $F^*(i, j)$ means the probability of the i -th vertex belonging to the j -th label. Thus, we may choose the label with the largest probability as the final label of a vertex, $\hat{y}_i = \arg \max_{1 \leq j \leq n_{label}} F^*(i, j)$. According to this strategy, the closed form formular is clearly equivalent to

$$F^* = (I - \alpha S - \beta C)^{-1} Y, \quad (3)$$

where $0 < \alpha \leq 1$, $0 \leq \beta < 1$ and $0 \leq \alpha + \beta \leq 1$.

Thus, we could develop a very simple closed form algorithm to solve the problem according to the Eq. (3).

Algorithm 1. Closed Form Algorithm for Community-Based Graph SSL

Input: Graph matrix W^g and community matrix W^c .

- 1: Construct the matrices S & C .
- 2: Predict the probability F^* of every label by Eq. (3).
- 3: Decide the final labels $\hat{y}_i = \arg \max_{j \leq n_{label}} F^*(i, j)$.

Output: Predicting labels \hat{Y} .

Iterative Algorithm. Because of the need of processing large-scale dataset and the drawback of the closed form algorithm, we proposed an iterative algorithm.

Algorithm 2. Iterative Algorithm for Community-Based Graph SSL

Input: Graph matrix W^g and community matrix W^c .

1: Initialize $F(0) = Y = (\bar{Y}_{label}, 0)$.

2: Construct the matrices S & C .

3: **repeat**

4: $F(t+1) = \alpha SF(t) + \beta CF(t) + (1 - \alpha - \beta)Y$.

5: **until** $|F(t) - F(t-1)| < \varepsilon$

6: Decide the final labels $\hat{y}_i = \arg \max_{j \leq n_{label}} F_{i,j}(t)$.

Output: Predicting labels \hat{Y} .

According to the mathematical deriving, we could obtain the limitation of $F(t)$ is equal to F^* in the Algorithm 1. Moreover, we easily found that the computation in every iteration only contains multiplication and addition of matrix, which have low computational complexity comparing to the computation of inverse matrix in the closed form algorithm.

2.3 Community Generation

In this section, we discuss the details of generating all the possible communities based on the groups and networks information in online social networks. First, we define the “distance” d between any two user v_i and v_j , $d_{i,j} = e^{-\|\langle w_{i,j}^{gp}, w_{i,j}^{nk} \rangle\|}$. According to this, we utilize a clustering method K -mean to generate communities C_1, C_2, \dots, C_{n_c} .

In fact, many other methods can be utilized to generate communities, e.g., Gaussian Mixture Model (GMM) and Graph Cut. But no matter what method is applied to generate communities, the CG SSL exploit model is still in effect.

3 Experiments

In the experiments, we employ both novel SSL exploit model and other three learning models as comparison, including two graph-based SSL and a supervised learning model, to predict the labels on a synthetic dataset and exposing which universities users come from on two real-world datasets. The results are evaluated in terms of accuracy and weighted accuracy on these three datasets.

3.1 Dataset Description

We describe the details of three datasets in this part. Table 1 gives detail statistics of these three datasets.

TwoMoons Dataset. ‘TwoMoons’ is a simple dataset only with 2 classes and 200 vertices distributing in 2D space. The distribution of the original data is shown in Table 2. Based on this, friendship information (local similarity) and community information

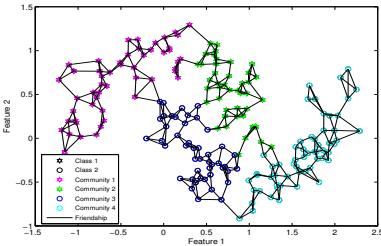


Fig. 2. The Synthetic Friendship and Networks Information of TwoMoons Dataset

Table 1. Statistics of TwoMoons, StudiVZ and Facebook Datasets

Dataset	Vertices	Edges	Groups	Networks	Classes
TwoMoons	200	381	0	4	2
StudiVZ	1,423	7,769	406	0	6
Facebook	10,410	45,842	61	78	3

Table 2. Statistics of Data Distribution on TwoMoons Dataset

Class	Class 1	Class 2
Size of Class	95	105

Table 3. Statistics of Data Distribution on StudiVZ Dataset

University	LMU Muenchen	Uni Wien	Uni Bayreuth
Size of Class	128	79	98
University	Uni Frankfurt am Main	TU Wien	(Others)
Size of Class	74	70	974

(community similarity) are artificially generated. These two kinds of synthetic information are shown in Fig. 2.

StudiVZ Dataset. The dataset has sufficient information of users' profiles and groups. Based on crawled data, we build a graph which contains 1,423 vertices and 7,769 edges. Data distribution is shown in Table 3.

Facebook Dataset. The dataset has sufficient number of vertices and all kinds of relational information, thus it is similar to the situation of the real world. Comparing with StudiVZ dataset, Facebook dataset has much more missing values in personal profile and more group information.

3.2 Data Preprocessing

For two real-world datasets, a series of data preprocessing such as feature selection, data cleaning and data translation are conducted before running algorithms.

Feature Selection. For users' profile information, we select top three features for which most people provide information. For relational information, a number of small groups and networks are removed. Apart from that, some networks whose names explicitly reveal universities' names, such as "LMU Muenchen", are removed manually.

Table 4. Statistics of Data Distribution on Facebook Dataset

University	CUHK	HKUST	(Others)
Size of Class	68	1,583	8,759

Data Translation. We need to translation some data into the proper forms. For example, we translate home town to its longitude and latitude values through Google maps API³ to calculate the similarity. Moreover, missing data are filled with average value of existed data and noise data are treated as missing ones. Cosine similarity is applied between any two profile vectors. If both of the users fail to provide at least 50% information, we set the cosine similarity with mean value.

3.3 Experiment Process

Labeled Data Selection. Labeled data are selected randomly with two constrains: 1. each class must have labeled data; 2. the numbers of labeled data in all classes are similar. The second point suggests an assumption that we do not know the distribution of all classes when labeling data.

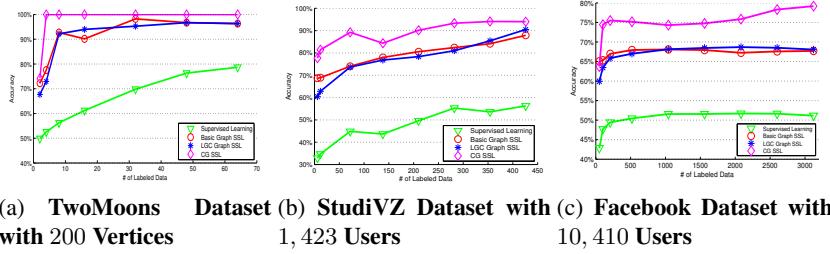
Evaluation Criterion. We mainly utilize the accuracy to measure the results of learning and a Weighted Accuracy (WA) measurement would assist us to analyst the confidence of the learning results. We define WA as $\frac{\sum_{i \in V_c} F^*(i, \hat{y}_i)}{\sum_{i \in V_c} F^*(i, \hat{y}_i) + \sum_{i \in V_{inc}} F^*(i, \hat{y}_i)}$, where V_c is a set containing all the vertices whose predictions are correct and V_{inc} contains all incorrect-prediction vertices.

3.4 Experiment Results

Table 5, 6 and 7 give the results of experiments, from which various algorithms' performance can be evaluated. Figure 3(a), 3(b) and 3(c) describe the accuracy of prediction with TwoMoons, StudiVZ and Facebook datasets respectively. What's more, the results of supervised learning are provided for comparison.

TwoMoons. Figure 3(a) shows the predicting results on the synthetic dataset ‘TwoMoons’. First, the accuracy of graph-based SSL models is obviously better than that of supervised learning. Second, Consistencies make the learning models stabler. The global consistency makes the LGC SSL stabler - the learning accuracies would keep enhancing along with the increasing of the number of labeled data. Moreover, the community consistency keeps the CG SSL stably better than other graph-based SSL models. Third, the community information does help in prediction in term of accuracy. In Fig. 2 we observe that some vertices have strong local similarity (friendship) with each other, but actually they do not belong to the same class. Without the help from community information, basic graph SSL and LGC SSL always incorrectly predict the classes of 6 to 8 vertices (Table 5), even if the percentage of labeled data is pretty high. The experiment on this synthetic dataset illustrates that the CG SSL could really improve the learning result in some ideal conditions.

³ <http://code.google.com/apis/maps/>

**Fig. 3.** Accuracy of Prediction on Three Datasets**Table 5.** Accuracy & Weighted Accuracy of Learning on TwoMoons Data with 200 Vertices

# of Labeled Data	Labeled %	Supervised Learning	Basic Graph SSL (weighted acc.)	LGC SSL (weighted acc.)	CG SSL (weighted acc.)
2	1.00%	49.9%	72.22% (73.77%)	67.71% (98.24%)	74.24% (100.00%)
4	2.00%	52.55%	77.55% (83.34%)	72.96% (98.35%)	100.00% (100.00%)
8	4.00%	56.25%	92.71% (95.18%)	92.19% (98.95%)	100.00% (100.00%)
16	8.00%	61.20%	90.16% (93.05%)	93.99% (99.23%)	100.00% (100.00%)
32	16.00%	69.82%	98.22% (98.80%)	95.27% (99.38%)	100.00% (100.00%)
48	24.00%	76.32%	96.71% (97.78%)	96.71% (99.34%)	100.00% (100.00%)
64	32.00%	78.68%	96.32% (97.50%)	96.32% (99.17%)	100.00% (100.00%)

Table 6. Accuracy & Weighted Accuracy of Learning on StudiVZ Data with 1,423 Users

# of Labeled Data	Labeled %	Supervised Learning	Basic Graph SSL (weighted acc.)	LGC SSL (weighted acc.)	CG SSL (weighted acc.)
6	0.42%	32.89%	68.67% (68.71%)	60.41% (59.89%)	77.66% (70.91%)
12	0.84%	34.73%	68.89% (68.95%)	62.79% (62.76%)	81.40% (84.17%)
72	5.06%	44.86%	74.02% (74.26%)	73.65% (73.88%)	89.23% (92.06%)
138	9.70%	43.66%	77.98% (78.37%)	76.81% (77.36%)	84.35% (83.71%)
210	14.76%	49.63%	80.54% (81.05%)	78.32% (78.98%)	90.15% (91.40%)
282	19.82%	55.93%	82.38% (82.91%)	80.98% (81.67%)	93.30% (98.91%)
354	24.88%	53.60%	84.10% (84.62%)	85.41% (85.94%)	94.06% (95.06%)
426	29.94%	56.27%	87.86% (88.29%)	90.47% (90.83%)	94.00% (91.52%)

StudiVZ. Figure 3(b) gives similar results. First, all graph-based SSL models outperform supervised learning. Second, the performance of CG SSL with 138 labeled data is worse than that with only 72 labeled data. We conjecture that it is due to the instability of the clustering technique for generating communities. Although we could tend to the optional predicting result, the randomness of clustering still exists and affects the stability of the final learning results.

Facebook. Figure 3(c) illustrates various algorithms' performance on Facebook dataset. First of all, in most cases the results of SSL methods are still superior to supervised learning. Second, even if there are only a few labeled data, CG SSL method can still make good predictions. The last point is that there is little instability in CG SSL model. The accuracy of learning with 10.00% labeled data is a little worse than that with only 4.99% labeled data. This would be caused by the same reason as in the experiment on the StudiVZ dataset.

Comparing with StudiVZ dataset, the learning results of CG SSL on Facebook dataset are less accurate. This is probably due to the existing of many missing values in Facebook dataset. However, the difference between CG SSL and other two graph-based SSL models is more obvious on Facebook dataset. We conjecture the reason is that there

Table 7. Accuracy & Weighted Accuracy of Learning on Facebook Data with 10, 410 Users

# of Labeled Data	Labeled %	Supervised Learning	Basic Graph SSL (weighted acc.)	LGC SSL (weighted acc.)	CG SSL (weighted acc.)
51	0.49%	42.86%	65.05% (58.72%)	59.93% (53.29%)	63.71% (62.05%)
102	0.98%	47.69%	65.32% (59.68%)	63.45% (61.53%)	74.44% (74.83%)
507	1.99%	49.43%	66.97% (61.67%)	65.88% (69.43%)	75.49% (76.52%)
519	4.99%	50.45%	67.94% (65.07%)	67.00% (82.27%)	75.17% (72.51%)
1041	10.00%	51.56%	68.09% (66.19%)	68.14% (88.85%)	74.32% (75.69%)
1560	14.99%	51.57%	67.93% (66.35%)	68.47% (91.29%)	74.76% (78.14%)
2082	20.00%	51.69%	67.24% (65.89%)	68.68% (92.43%)	75.84% (79.94%)
2601	24.99%	51.65%	67.58% (65.31%)	68.51% (92.81%)	78.30% (84.96%)
3123	30.00%	51.16%	67.71% (65.31%)	68.05% (92.59%)	79.17% (80.32%)

is more effective community information on Facebook dataset and the distribution of community information is more relative and helpful to the predicted attribute.

Summary. In terms of both accuracy and confidence (or certainty), CG SSL exploit model performs better than other two graph-based learning models in most cases and its advantage is amplified gradually when the number of labeled data increases.

4 Related Work

Since the online social networks began to thrive, there has been a growing interest in the security of users' privacy under the current privacy protection. Among the previous work, the exposures using machine learning with public profile and relation information attract a large amount of attention and have great significance in the security of online social networks [7][10]. The exposures employing machine learning methods include supervised learning and unsupervised learning at the beginning.

Based on the characteristic of semi-supervised learning, it has attracted many researchers to study in the last decades. Semi-supervised learning can be divided into several typical kinds of models, including generative model [4], co-training method [3], graph-based methods [12], SVM [8][9], etc.

Graph-based semi-supervised learning methods model objects as weighted undirected graphs. Blum and Chawla [2] pose semi-supervised learning as a graph mincut problem. Another graph-based semi-supervised learning algorithm proposed in [12] is the harmonic function that is a function which has the same values as given labels on the labeled data and satisfied the weighted average property on the unlabeled data. Based on [12], [11] proposes the Local and Global Consistency graph-based method which improves harmonic function method.

5 Conclusions

Community-based Graph SSL model describes the real world exactly. With the help of community consistency, this model illustrates the further relationship among all users in the real world. Moreover, this paper provides two algorithms to solve the problem. In contrast with previous graph-based SSL models, CG SSL predicts the sensitive information of online social networks with higher accuracy and confidence. Thus, the privacy exposure problem in online social networks becomes more serious and the security of users' information is no longer secure.

References

1. <http://www.facebook.com/press/info.php?statistics>
2. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: Proc. of ICML, pp. 19–26. Citeseer (2001)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual Conference on Computational Learning Theory, p. 100. ACM, New York (1998)
4. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
5. Getoor, L., Taskar, B.: Introduction to statistical relational learning. MIT Press, Cambridge (2007)
6. He, J., Chu, W., Liu, Z.: Inferring privacy information from social networks. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975, pp. 154–165. Springer, Heidelberg (2006)
7. Mo, M., Wang, D., Li, B., Hong, D., King, I.: Exploit of Online Social Networks with Semi-Supervised Learning. In: Proc. of IJCNN, pp. 1893–1900. IEEE, Los Alamitos (2010)
8. Vapnik, V.: Structure of statistical learning theory. Computational Learning and Probabilistic Reasoning, 3 (1996)
9. Xu, Z., Jin, R., Zhu, J., King, I., Lyu, M., Yang, Z.: Adaptive Regularization for Transductive Support Vector Machine. In: Proc. of NIPS, pp. 2125–2133 (2009)
10. Zheleva, E., Getoor, L.: To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In: Proc. of WWW, pp. 531–540. ACM, New York (2009)
11. Zhou, D., Bousquet, O., Lal, T., Weston, J., Scholkopf, B.: Learning with local and global consistency. In: Proc. of NIPS, pp. 595–602. MIT Press, Cambridge (2004)
12. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: Proc. of ICML (2003)

Hierarchical Lossless Image Coding Using Cellular Neural Network

Seiya Takenouchi¹, Hisashi Aomori¹, Tsuyoshi Otake², Mamoru Tanaka³,
Ichiro Matsuda¹, and Susumu Itoh¹

¹ Department of Electrical Engineering, Tokyo University of Science,
2641 Yamazaki, Noda-shi, Chiba 278–8510 Japan

² Department of Software Science, Tamagawa University,
6–1–1 Tamagawagakuen, Machida, Tokyo, 194–8610 Japan

³ Department of Information and Communication Sciences, Sophia University,
7–1 Kioi-cho, Chiyoda-ku, Tokyo 102–8554 Japan

Abstract. In this paper, a novel hierarchical lossless image coding scheme using the cellular neural network (CNN) is proposed. The coding architecture of the proposed method is based on the lifting scheme that is one of the scalable coding framework for still images, and its coding performance strongly depends on the prediction ability. To cope with this spontaneously characteristic, an image interpolation is modeled by an optimal problem that minimizes the prediction error. To achieve the high accuracy prediction with a low coding rate, two types of templates are used for dealing with the local structure of the image, and the CNN parameters are decided by the minimum coding rate learning. In the coding layer, the arithmetic coder with context modeling is used for obtaining a high coding efficiency. Experimental results in various standard test images suggest that the coding performance of our proposed method is better than that of conventional hierarchical coding schemes.

Keywords: discrete-time cellular neural network, hierarchical coding, context modeling.

1 Introduction

Lossless image coding methods are necessary for high quality image applications such as remote sensing and medical diagnosis. In general, their designing algorithms are based on the power minimizing problem with respect to the prediction residuals [1]. However, the aim of the lossless image coding is to achieve the high compression ratio. Hence, the reduction of the coding ratio as well as power of the prediction residuals is very important. Actually, the effectiveness of the rate minimization designing concept was validated in [2]. Moreover, owing to the development of imaging devices such as mobile phone, digital photo frame, television, etc., the importance of the scalable image coding framework like JPEG2000 becomes high.

The cellular neural network (CNN) [3] which is one of artificial neural networks, has been applied to many applications such as binary image processing, image compression, pattern recognition, and so on [4][5][6]. In our previous works [6], the hierarchical lossless image coding algorithm using the CNN were proposed. From the viewpoint of the

optimal lossless coding, however, two important points were not considered in these works. That is, the single template can not perform the optimal prediction considering the local structure of the image, and the objective function of CNN that evaluates the distortion between the original image and the prediction image, is not suitable for the rate minimization coding. Since the coding rate using CNNs prediction can not be formulated, it is very difficult to design the CNNs by using the rate minimization model directly.

In this paper, a novel hierarchical lossless image coding scheme using the CNN is proposed. In our method, two types of templates are used for dealing with the local structure of the image, and the CNN parameters are decided by the minimum coding rate learning. Therefore, the proposed method can optimize not only the distortion but also a coding rate. Moreover, the context modeling in the coding layer is utilized to achieve a high coding efficiency. By the computer simulation using various standard test images, it is confirmed that the coding performance of proposed algorithm is better than that of conventional hierarchical lossless coding methods.

2 JPEG2000

2.1 The Lifting Scheme

The JPEG2000 which is the latest international still image coding standard, is based on the lifting scheme [7] that is a new generation wavelet transform framework. Fig. 1 shows the typical lifting scheme, and it is composed of following three steps. The input image X is divided into even polyphase components X_e and odd polyphase components X_o . In the predict step, the odd polyphase components X_o are predicted using the neighboring even polyphase components via the predict filter. Then prediction residuals d is given by

$$d = X_o - P(X_e), \quad (1)$$

where $P()$ is the predict filter. Next, the even polyphase components X_e are updated by

$$c = X_e + U(d), \quad (2)$$

where $U()$ is the update filter. The updated coefficients c become the input of the next stage, and these lifting processes are repeated iteratively. Finally, the updated coefficients of final lifting stage and all prediction residuals are transmitted to the entropy coding layer called EBCOT.

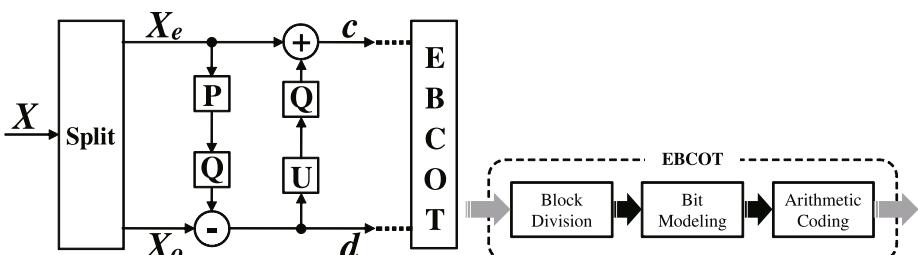


Fig. 1. The integer lifting scheme (encoder)

Fig. 2. EBCOT

2.2 The Entropy Coding

The EBCOT is the entropy coding system of the JPEG2000. In this encoding process shown in Fig. 2, the wavelet coefficients are divided into code blocks. Each code block is processed independently, and multilevel wavelet coefficients in a code block are transformed into binary bit-planes. All bit-planes are encoded by a context-driven binary arithmetic coder called the MQ-coder.

3 Hierarchical Lossless Image Coding Using CNN

The proposed hierarchical lossless image coding system is shown in Fig. 3. This system is composed of the basic predict lifting scheme and an entropy coding layer. At the split step, the input image U is divided into even polyphase components U_e and odd polyphase components U_o . The sampling lattice is the quincunx subsampling as shown in Fig. 4 and one level corresponds to two subsampling patterns. The prediction for odd polyphase components in each stage is designed using the CNN having two types of templates for dealing with the local structure of the image. Moreover, the CNN parameters are decided by the minimum coding rate learning. Therefore, the proposed method can optimize not only the distortion but also a coding rate. Then, the prediction residuals e are obtained. In the encoder, abovementioned processes are applied iteratively. The even polyphase components of final lifting stage and all prediction residuals are transmitted to the entropy coding layer using context modeling. In the entropy coding layer, each prediction residuals are encoded from the final stage. In addition, since the resolution of the input image is shrunked to 1/4, various version of the input image can be obtained.

3.1 CNN Image Interpolation with Minimum Coding Rate Learning

Discrete Time CNN. Fig. 5 shows the block diagram of the discrete time CNN (DT-CNN) with output template. The state equation is described as

$$x_{ij}(t+1) = \sum_{C(k,l) \in N_r(i,j)} B(i, j; k, l) u_{kl} + \sum_{C(k,l) \in N_r(i,j)} A(i, j; k, l) y_{kl}(t) + T_h, \quad (3)$$

$$y_{ij}(t) = f(x_{ij}(t)), \quad (4)$$

$$\tilde{u}_{ij} = \sum_{y(k,l) \in N_r(i,j)} G_{out}(i, j; k, l) y_{kl}, \quad (5)$$

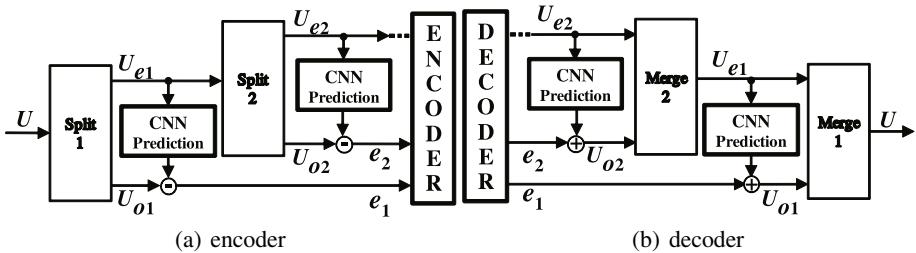
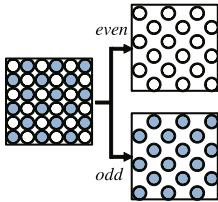
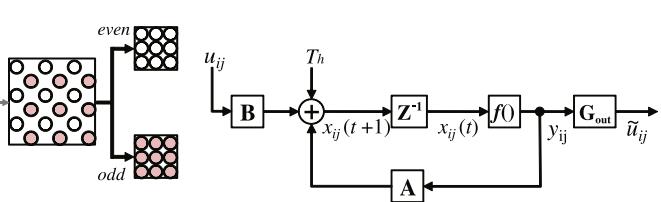


Fig. 3. The proposed hierarchical encoder and decoder

**Fig. 4.** The Quincunx sampling lattice**Fig. 5.** The cell of DT-CNN with output template

where $x_{ij}(t)$, $y_{ij}(t)$, u_{ij} , T_h , $f()$, $A(i, j; k, l)$, $B(i, j; k, l)$, and $G_{out}(i, j; k, l)$ are the internal state, the output, the input of the cell (basic processing unit of CNN), the threshold, the quantization function, the feed-back template, the feed-forward template, and the output template, respectively. The r -neighborhood $N_r(i, j)$ is defined by $N_r(i, j) = \{C(k, l) | \max\{|k - i|, |l - j|\} \leq r\}$. Also, the Lyapunov energy function of DT-CNN [3,5] is defined in matrix form by

$$E(t) = -\frac{1}{2}\mathbf{y}^T(\mathbf{A} - \delta\mathbf{I})\mathbf{y} - \mathbf{y}^T\mathbf{B}\mathbf{u} - \mathbf{T}_h^T\mathbf{y}, \quad (6)$$

where δ is the possible constant value to determine the quantizing region, \mathbf{I} is the unit matrix, and \mathbf{a}^T is the transposed matrix of \mathbf{a} . If \mathbf{A} is symmetric and its center value is larger than zero, then the Lyapunov function becomes a monotonic decreasing [3,5].

The Image Interpolation by CNN. The even polyphase images such as Fig. 4 are interpolated using the local structure driven CNNs prediction, where two types of templates are adaptively used by the edge strength information from the Sobel filter (see Fig. 6). Since the coding rate using the CNNs prediction can not be formulated, it is very difficult to design the CNNs by using the rate minimization model directly. Therefore, the CNNs are designed by the minimization model of distortion between the original image and the prediction image. Then, the CNN parameters are decided by the minimum coding rate learning. Hence, the proposed method can optimize not only the distortion but also a coding rate. Let \mathbf{G} be the 2D Gaussian filter, then the distortion function is defined by

$$\text{dist}(\mathbf{y}, \mathbf{u}) = \left\| \frac{1}{2}\mathbf{y}^T(\mathbf{G}\mathbf{y} - \mathbf{u}) \right\|. \quad (7)$$

By comparing Eqs. (6) and (7), the templates of a space variant DT-CNN can be determined as

$$\mathbf{A} = A(i, j; k, l), \quad C(k, l) \in N_r(i, j) \quad (8)$$

$$= \begin{cases} 0 & \text{if } k = i \text{ and } l = j, \\ -\frac{1}{2\pi\sigma^2} \exp\left(-\frac{(k-i)^2 + (l-j)^2}{2\sigma^2}\right) & \text{otherwise,} \end{cases}$$

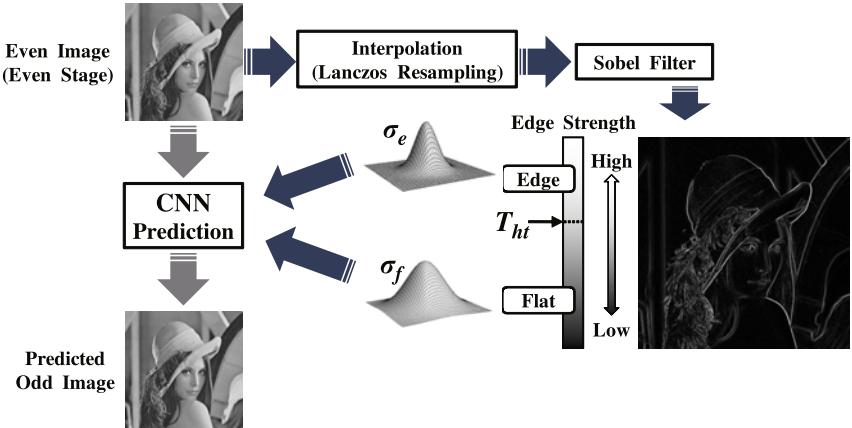


Fig. 6. Adaptive CNN prediction considering local structure

$$\mathbf{B} = B(i, j; k, l), \quad C(k, l) \in N_r(i, j) \quad (9)$$

$$= \begin{cases} 1 & \text{if } k = i \text{ and } l = j, \\ 0 & \text{otherwise,} \end{cases}$$

$$T_h = O, \quad (10)$$

where O is the zero matrix. Then, the interpolated image \tilde{u} can be obtained by using the equilibrium solution of the DT-CNNs through output template G_{out} that is defined in each stages. In the odd stages, the G_{out} is given by

$$\begin{aligned} G_{out}^o &= G_{out}^o(i, j; k, l), \quad C(k, l) \in N_r(i, j) \\ &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(k-i)^2 + (l-j)^2}{2\sigma^2}\right), \end{aligned} \quad (11)$$

Also, in the even stages, the G_{out} is spatially extended version of G_{out}^o as

$$\begin{aligned} G_{out}^e &= G_{out}^e(i, j; k, l), \quad C(k, l) \in N'_r(i, j) \\ &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(k-i)^2 + (l-j)^2}{2\sigma^2 d_m^2}\right), \end{aligned} \quad (12)$$

where $N'_r(i, j) = \{C(k, l) | \max\{|k - i|, |l - j|\} \leq rd_m\}$. For adaptive template switching in order to deal with the local structure of an input image, the local structure of an input image is classified into a flat part and an edge part by a threshold T_{ht} that is an edge strength value. The r -neighborhood and the standard deviation of Gaussian filter in the flat part are larger than that of the edge part. Moreover, these parameters are learned by following processes for minimizing the cost function $\text{cost}(\sigma_f, \sigma_e, T_{ht})$ that evaluates the coding rate of prediction residuals.

- Step:1 σ_e and σ_f are set to 0.8 in odd stages, and are set to 0.5 in even stages. T_{ht} is set to the mean value of edge strength of an even polyphase components.
- Step:2 Assuming that $\sigma_e = \sigma_f$ and T_{ht} are constants, then σ_f is given by

$$\sigma_f = \underset{\sigma_f}{\operatorname{argmin}} \text{cost}(\sigma_e, \sigma_f, T_{ht}).$$
- Step:3 Assuming that σ_f and T_{ht} are constants, then σ_e is decided by

$$\sigma_e = \underset{\sigma_e}{\operatorname{argmin}} \text{cost}(\sigma_e, \sigma_f, T_{ht}).$$
- Step:4 Assuming that σ_f and σ_e are constants, then T_{ht} is obtained by

$$T_{ht} = \underset{T_{ht}}{\operatorname{argmin}} \text{cost}(\sigma_e, \sigma_f, T_{ht}).$$
- Step:5 If all parameters are unchanged, go to step 6, otherwise return to step 2.
- Step:6 End.

3.2 The Entropy Coding with Context Modeling

The prediction residuals in each stages are arithmetic encoded with the context modeling. The context shows the distribution model of prediction residuals, and it can be classified by using feature quantity U_m given by

$$U_m = \sum_{k=1}^N \frac{1}{\delta_k} |e_{d,k}|, \quad (13)$$

where δ_k is the Manhattan distance between the target pixel and each neighborhood pixels, $e_{d,k}$ is the prediction error, d is the number of reference stages, and N is the number of reference pixels. The prediction errors are encoded from the highest stage. For high accuracy context estimation, U_m in middle stages can be calculated using pre-encoded prediction errors as shown in Fig. 7. Then U_m is quantized into sixteen groups by the threshold $T_{hm}(n)$ ($n = 1, 2, \dots, 15$) (see Fig. 8). Since each group is determined by the encoded prediction error $e_{d,k}$ and the threshold T_{hm} , the probability model can be switched each pixel. The probability models $P(e)$ can be determined using the generalized Gaussian function by

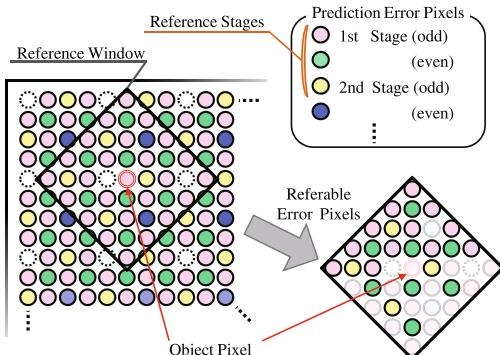


Fig. 7. Calculating the feature quantify U_m ($d = 3, N = 40$)

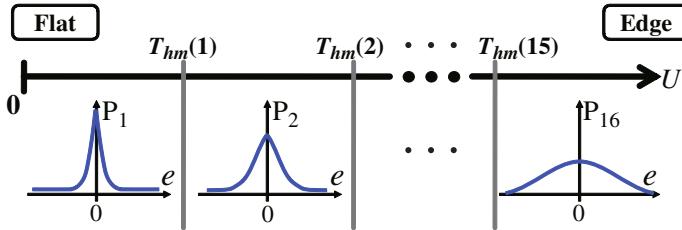


Fig. 8. The context modeling

$$P(e) = \frac{c \cdot \eta(c, \sigma)}{2\Gamma(1/c)} \exp \left\{ -|\eta(c, \sigma) \cdot e|^c \right\}, \quad (14)$$

$$\eta(c, \sigma) = \frac{1}{\sigma} \sqrt{\frac{\Gamma(3/c)}{\Gamma(1/c)}}, \quad (15)$$

where $\Gamma()$ is the Gamma function, and c is a shape parameter which controls shapness of the probability model. The shape parameter of each group is experimentally decided in order to suit well to the actual distribution.

4 Simulation Results

To evaluate the coding performance of our proposed lossless image coding algorithm, we implemented the coder and decoder by software. We applied our system to twenty

Table 1. Lossless image coding performance [bits/pel]

Image	single-template	proposed method	JPEG2000
Aerial	5.379	5.060	5.192
Airfield	5.784	5.591	5.834
Airplane	4.376	4.009	4.015
Baboon	6.299	5.960	6.109
Barbara	5.480	4.909	4.845
Boat	5.154	4.891	4.880
Couple	5.101	4.813	4.839
Crowd	4.416	4.022	4.197
Elaine	5.064	4.934	4.944
Goldhill	4.984	4.729	4.836
Lax	6.018	5.885	5.962
Lenna	4.805	4.553	4.685
Lennagrey	4.465	4.184	4.306
Milkdrop	4.120	3.803	3.768
Peppers	4.857	4.592	4.631
Sailboat	5.285	4.992	5.152
Shapes	3.619	2.776	1.928
Tiffany	4.530	4.298	4.223
Tiffanygrey	5.093	4.728	4.812
Water	4.415	4.330	4.177
Average	4.962	4.653	4.667

512×512 8-bit standard grayscale test images. The performance of the proposed method was compared with the basic CNN coding system with single probability model and JPEG2000 reference software. For the simulation, the coding factor is decided experimentally; the number of lifting level $L = 6$ (12 stages), the r -neighborhood of a flat region $r_f = 2$, the r -neighborhood of an edge region cell $r_e = 1$. The initial values of σ_f and σ_e are 0.8 in the odd stage and 0.5 in the even stage. Also, in the entropy encoding layer, the reference pixels $N = 40$ and the reference stages $d = 3$. Table 1 shows the coding performance of each method. As shown in Table 1, the average coding rate of the proposed method outperforms that of conventional methods.

5 Conclusion

In this paper, we proposed a novel hierarchical lossless image coding scheme using the CNN. In our method, two types of templates are used for dealing with the local structure of the image. Since the CNN parameters are decided by the minimum coding rate learning, the proposed method can optimize not only the distortion but also a coding rate. Moreover, in the coding layer, the context modeling helps to obtain a high coding efficiency. Experimental results in various standard test images suggest that our proposed method outperforms the conventional hierarchical coding methods including JPEG2000.

Acknowledgment

This research was partially supported by the Grant-in-Aid for Young Scientists (B) of Ministry of Education, Culture, Sports, Science and Technology, Japan (MEXT).

References

1. Golchin, F., Paliwal, K.: Classified adaptive prediction and entropy coding for lossless coding of images. In: Proc. International Conference on Image Processing, vol. 3, pp. 110–113 (October 1997)
2. Matsuda, I., Mori, H., Itoh, S.: Lossless coding of still images using minimum-rate predictors. In: Proc. International Conference on Image Processing, vol. 1, pp. 132–135 (2000)
3. Chua, L.O., Yang, L.: Cellular neural networks: Theory. IEEE Trans. Circuits and Syst. 35(10), 1257–1272 (1988)
4. Venetianer, P.L., Roska, T.: Image compression by cellular neural networks. IEEE Trans. Circuits Syst. 45, 205–215 (1998)
5. Yokosawa, K., Nakaguchi, T., Tanji, Y., Tanaka, M.: Cellular neural networks with output function having multiple constant regions. IEEE Trans. Circuits Syst. 50(7), 847–857 (2003)
6. Aomori, H., Kawakami, K., Otake, T., Takahashi, N., Tanaka, M.: Separable 2d lifting using discrete-time cellular neural networks for lossless image coding. IEICE Trans. on Fundamentals E88-A(10), 2607–2614 (2005)
7. Sweldens, W.: The lifting scheme: A custom-design construction of biorthogonal wavelets. Appl. Comp. Harmon. Anal. 3(2), 186–200 (1996)

Multivariate Decision Tree Function Approximation for Reinforcement Learning

Hossein Bashashati Saghezchi and Masoud Asadpour

Department of Electrical and Computer Engineering, University of Tehran, Iran
`{bashashati, asadpour}@ut.ac.ir`

Abstract. In reinforcement learning, when dimensionality of the state space increases, making use of state abstraction seems inevitable. Among the methods proposed to solve this problem, decision tree based methods could be useful as they provide automatic state abstraction. But existing methods use univariate, therefore axis-aligned, splits in decision nodes, imposing hyper-rectangular partitioning of the state space. In some applications, multivariate splits can generate smaller and more accurate trees. In this paper, we use oblique decision trees as an instance of multivariate trees to implement state abstraction for reinforcement learning agents. Simulation results on mountain car and puddle world tasks show significant improvement in the average received rewards, average number of steps to finish the task, and size of the trees both in learning and test phases.

Keywords: Reinforcement Learning, Decision Tree, State Abstraction.

1 Introduction

Reinforcement learning is a simple and popular method in robot learning in which, an agent learns by trial and error. The agent perceives the environment's state through its sensors, selects an action, and gets a scalar feedback, as a reinforcement signal. The agent tries to maximize its expected reward.

The simplest approaches to reinforcement learning use lookup tables; However, in case of dealing with high dimensional state spaces, a relatively large numbers of trials is required (this problem is called the curse of dimensionality).

Number of learning trials can be reduced, to some extent, by implementing state abstraction techniques. Among the proposed methods for state abstraction, function approximation methods [1] are one of the most popular methods; however possibility of divergence is their main drawback [2]. Some other approaches, automatically abstract states to non-uniform partitions [3], [4], [5], [6], [7], [8]. They model the whole world with a single abstract state in their first phase, and then split the abstract states recursively when they find level of abstraction is too high. Among these methods we employ decision trees techniques [5], [6], [7], [8], as they are more human readable and provide hierarchical state abstraction.

However, decision tree-based abstraction methods usually generate univariate splits in their decision nodes. This means they divide the state space to hyper-rectangular partitions. Many real world tasks need multivariate splits, thus creating

linear (in any direction) or even non-linear shapes in the state space. In such cases, it is obvious that approximating these concepts using linear axis-aligned splits might increase the size of trees and decrease their accuracy.

In this paper, we use linear multivariate splits instead of axis-aligned splits. We extend the application of oblique classifier trees as linear multivariate trees and apply them to state abstraction in reinforcement learning.

The paper is organized in six sections. After introduction, related works will be explained. Then, our method and the splitting criteria we use are described. Afterwards, the selected problems, simulation results and discussions will come. Then conclusion and future works will be mentioned.

2 Related Works

Decision tree has been used to approximate the value function in reinforcement learning. The G-Learning algorithm [4] employed decision tree to discretize the state space with variable resolutions. In this method, the world is firstly considered as a single state and then the algorithm tries to split the states recursively when necessary. However, this method was designed to work with binary inputs. Here, T-Test is performed on historical data to determine when to split a node. In U-Tree method [5], [6], G-learning was extended to work in continuous state spaces.

The U-Tree algorithm tries to partition the state space incrementally. It starts by considering the whole state space as a leaf of the decision tree. Then it recursively tries to split the leaf nodes to generate a better tree. This algorithm consists of two major phases: sampling and splitting phases. In sampling phase the agent acts as a standard reinforcement learning agent and by using the decision tree, translates its perception from the environment to a leaf node which corresponds to a state. In this phase, it tries to gather a history of the transitions in the state space, i.e. $T_i(s, a, r, s')$, in which s is the current state, a current action, r the reward and s' is the next state. A value is assigned to each transition, T_i .

In the splitting phase, the algorithm inspects all leaf nodes to see whether it can be split into two new leaves. This splitting is based on some criteria, which compare the values of transitions in each new leaf and returns a numerical value indicating the difference between the two distributions. If the largest of these values is greater than a given threshold, the split is performed. In [5], [6], [7], [8] some of these splitting criteria are introduced. The criteria which we use in this paper are introduced in 3.1.

The aim of the methods described above is to find the line which optimizes the splitting criterion, i.e. the best axis-aligned split. Based on our discussion in previous section, we chose oblique classifiers for internal nodes of the decision tree. The main problem here is that there are $2^d \cdot (n + d)$ candidate oblique hyperplanes to split a leaf node, where d is the dimension of the state space (i.e. number of features) and n is number of samples. This makes the problem of finding the optimal oblique hyperplane NP-hard [12], whereas finding the optimal axis-aligned split only requires $n \cdot d$ iterations over the candidate lines.

There are many methods in the supervised learning literature which are designed to find an oblique split. Most of these methods are designed to work with labeled data [9]. These methods use some classes of splitting criteria that are based on the number of misclassified samples, or the distance of the samples from the separating hyperplane,

etc. In contrast, there are two other methods, namely the CART [10] and the OC1 [11] algorithms, which find a hyperplane, split without having any knowledge about the labeling of the dataset. These methods can be used with any type of splitting criteria, and therefore their resulting hyperplane might be suboptimal.

Considering the U-Tree algorithm, which only uses value of the transitions for splitting a leaf node and has therefore no labeled samples in each leaf node, we need to use one of the algorithms in the latter group. Finding an oblique split with OC1 takes $O(dn^2 \log n)$, but CART suffers from unbounded execution time, i.e. there is no upper bound on its execution time [11]. Knowing that, finding an axis-aligned split takes $O(dn^2)$, OC1 takes at most $\log n$ times longer.

3 U-Tree with Linear Splits

Our method makes use of the OC1 [11] algorithm in the splitting phase of the U-Tree algorithm, but the original U-Tree algorithm uses a simple search in the space of candidate axis-aligned hyperplanes. The OC1 algorithm iterates over two major phases. In the first phase, a deterministic hill climbing algorithm is used to compute the coefficients of a hyperplane split which maximizes the splitting criteria. In the second phase a random perturbation is applied to the hyperplane vector to improve the impurity measure.

For instance, assume the initial hyperplane is $a_1x_1 + a_2x_2 + \dots + a_nx_n + a_0 = 0$. To determine the best value for a coefficient, e.g. a_1 , the algorithm considers the values of all the other coefficients, i.e. a_2, \dots, a_n , as fixed and computes for each data sample the value for that coefficient, i.e. a_1 . After sorting the computed coefficients, the measure is calculated for the midpoint of each two adjacent ones, and the initial coefficient is replaced by the midpoint with minimum impurity measure.

In the randomization phase, the algorithm chooses a random direction and tries to perturb the line in that direction. If this process improves the impurity measure of the hyperplane split then the last split is replaced with the new one and the whole process continues for a fixed number of iterations.

3.1 Splitting Criteria

Many splitting criteria have already been proposed in literature e.g. [5],[6],[7],[8]. Some, like Kolmogorov-Smirnov (KS) test are general methods and some like the ones introduced in [7] are specialized tests that are more efficient in computation time as well as learning performance in reinforcement learning. In this paper, we use the two specialized criteria abbreviated as VAR (Variance Reduction) and SANDS (State Abstraction for Non-Deterministic Systems):

- **VAR**

This criterion minimizes the mean squared error of the action values on either side of the split:

$$VAR(\bar{S}) = \max_a \hat{\rho}_1^a \hat{\rho}_2^a (\hat{\mu}_1^a - \hat{\mu}_2^a)^2 \quad (1)$$

$\hat{\rho}_i^a$: The fraction of samples that belong to either sides of the trial split.

$\hat{\mu}_i^a$: Value of action a averaged on the samples on either sides of the trial split.

- SANDS

In this criterion, the expected reward return is maximized by finding a splitting point in such a way that the policies, as well as the Q-values, is well differentiated.

$$SANDS(\bar{S}) = \max_a \sum_{i=1}^2 (\hat{\pi}_i^a - \hat{\pi}^a) \hat{\mu}_i^a \hat{\rho}_i^a \quad (2)$$

$\hat{\pi}^a$: Current estimation of probability of action a .

$\hat{\pi}_i^a$: Probability of action a estimated from samples in either sides of the trail split.

4 Problem Domains

Our method is compared in simulation with univariate splits on two standard reinforcement learning tasks proposed: Mountain Car and Puddle World.

In Mountain Car (Fig.1-left), an agent should drive a car to reach the top of a mountain. As the car engine is not powerful enough to overcome gravity, agent should learn how to drive backwards up the hill in order to produce sufficient inertia to climb up to the goal before losing its speed. The agent's state is its position, x , and velocity, v , where $-1.2 \leq x \leq 0.5$ and $-0.07 \leq v \leq 0.07$. The agent is initialized at the bottom of the hill. Each time step, the agent is punished with -1 except if it reaches the goal. Three possible actions are positive, negative and zero throttle.

In Puddle World (Fig.1-right), an agent that starts from the bottom left corner, learns to navigate safely to the upper right corner. Meanwhile, it should take care of puddles. The agent's state is expressed by its x and y positions. Each time step, the agent is punished with -1 until it finds the goal state. If the agent goes into a puddle, it gets an additional punishment, -400 times the depth of the puddle. The agent has four possible actions: moving up, down, left and right by 0.05 plus a small amount of Gaussian noise with the standard deviation 0.01.

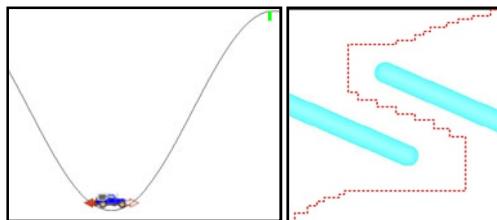


Fig. 1. The mountain car and puddle world domains used in the simulations

5 Results and Discussion

For the mountain car problem, we run each method for 10,000 episodes. Episodes start from the position shown in Fig.1-left and end if the car reaches the goal or a

maximum number of steps, here 5000, are passed. The results are averaged over 10 different runs. For the puddle world problem, we run the algorithms for 15,000 episodes with the maximum of 1500 steps in each episode. The results are averaged over 15 runs. For our simulations we have used rl-glue simulator [14].

We use Watkin's $Q(\lambda)$ algorithm with ϵ -greedy action selection, the learning parameters are $\alpha=0.2$, $\lambda=0.9$ and $\gamma=0.95$. In the mountain car ϵ is initialized to 0.5 and is decayed by multiplying to 0.99983. In the puddle world, ϵ is initialized to 1 and is decayed by multiplying to 0.99982. These multiplying factors are chosen so that ϵ becomes approximately 0.08 during the final episodes of the training phase.

The last 1000 episodes are dedicated to test phase, in which learning stops and the agent acts based on the greedy policy, i.e. selects the action with highest value.

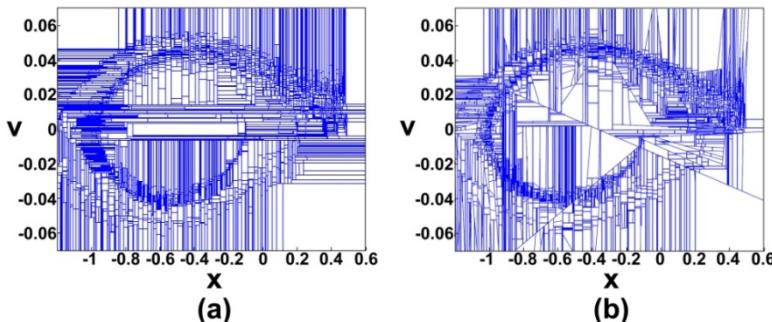


Fig. 2. A sample of partitioning the state space with (a) univariate and (b) oblique decision trees in the Mountain Car domain

Fig. 2 shows a sample of partitioning the state space with univariate and oblique decision trees for the mountain car problem. The learning curve for the mountain car domain is shown in Fig. 3 (a). Table 1 compares oblique trees with univariate splits in terms of the average received reward per episode in test phase and the number of leaves in the learned decision trees. Bigger average rewards indicate less number of actions from the initial state to goal, and fewer number of leaves indicates smaller trees and more efficient state abstractions (if it comes with big rewards). The results show that oblique trees are slightly better than their univariate counterparts in terms of both received rewards and speed of convergence. Since the Mountain Car problem is a bit simple, the difference between the received rewards is not clear enough, even though the number of states decreases.

Table 1. Mountain Car results

Methods	Average Reward in test phase	Number of States
Oblique: SANDS	-81.196 ± 2.399	3430 ± 465
univariate: SANDS	-82.339 ± 0.956	3603 ± 168
Oblique: VAR	-80.434 ± 1.360	2915 ± 114
univariate: VAR	-80.525 ± 0.472	3070 ± 78

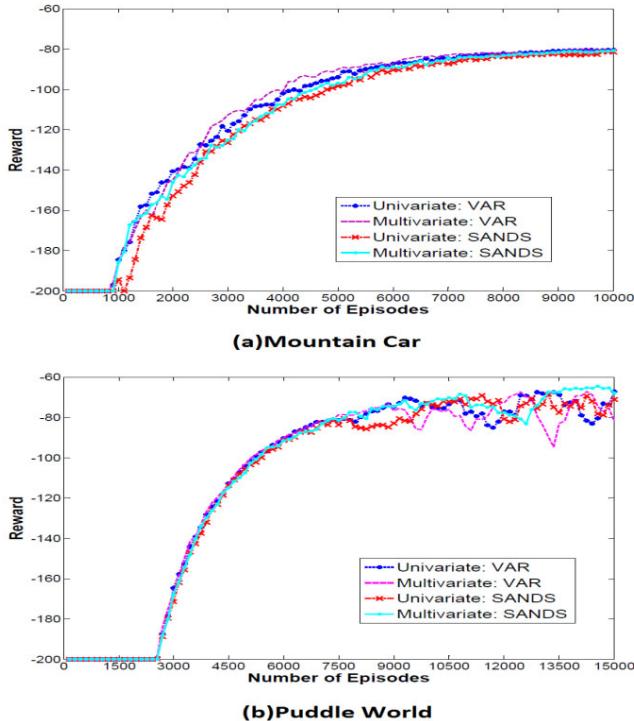


Fig. 3. The Learning Curves for (a) Mountain Car, and (b) Puddle World

Table 2. Puddle World results

Methods	Average Reward in test phase	Number of States
Oblique: SANDS	-65.708 ± 3.204	2861 ± 59
univariate: SANDS	-74.037 ± 5.285	2882 ± 28
Oblique: VAR	-73.563 ± 3.447	3138 ± 35
univariate: VAR	-76.892 ± 16.126	3134 ± 33

Fig. 4 shows a sample of partitioning the state space with univariate and oblique decision trees in Puddle World problem. Fig. 3(b) shows the amount of received rewards per episode for the Puddle World problem. Table 2 includes numerical data on the average reward (per episode) in test phase and the number of leaves of the learned trees. Fig. 3(b) shows that learning speeds of oblique and univariate trees are approximately equal. Table 2 indicates that oblique trees have been improved over the univariate trees in terms of the average reward in test phase, while sizes of the trees have remained almost the same.

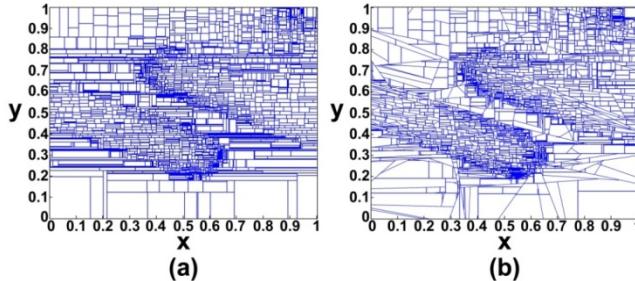


Fig. 4. A sample of partitioning the state space with (a) univaraite and (b) oblique decision trees in the Puddle World domain

6 Conclusion and Future Works

In this paper, we used a combination of U-Tree method and OC1 algorithm to generate oblique decision trees for state abstraction in reinforcement learning. Our results show that using the proposed method instead of its univariate counterparts for state abstraction tends to generate smaller and more efficient decision trees in relatively difficult domains, while its time complexity is at most $\log n$ times worse than univariate decision trees.

Our proposed method does not guarantee finding an optimal oblique split for internal nodes of the decision tree. This problem can be solved by introducing other splitting criteria, which transform the optimization problem of finding the oblique split to a convex optimization problem. As the resulting trees are still relatively large, the method might not be suitable in complex learning tasks. Adding pruning methods as well as locally modifying the oblique splits in each node might result in smaller and more accurate trees.

References

1. Tesauro, G.: Temporal difference learning and td-gammon. *Communications of the ACM* 38(3), 58–68 (1995)
2. Boyan, J.A., Moore, A.W.: Generalization in reinforcement learning: Safely approximating the value function. In: Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.) *Advances in Neural Information Processing Systems*, Cambridge, MA, vol. 7 (1995)
3. Moore, A.W.: The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. In: Cowan, J.D., Tesauro, G., Alspector, J. (eds.) *Advances in Neural Information Processing Systems*, vol. 6, pp. 711–718. Morgan Kaufmann Publishers, Inc., San Francisco (1994)
4. Chapman, D., Kaelbling, L.P.: Input generalization in delayed reinforcement learning: An algorithm and performance comparisons. In: *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pp. 726–731 (1991)
5. McCallum, A.: Reinforcement with Selective Perception and Hidden State. PhD thesis, Computer Science Dept., University of Rochester (1995)

6. Uther, W.R., Veloso, M.M.: Tree Based Discretization for Continuous State Space Reinforcement Learning. In: Proceedings of the National Conference on Artificial Intelligence, pp. 769–775 (1998)
7. Asadpour, M.: Behavior Design in Microrobots: Hierarchical Reinforcement Learning under Resource Constraints. PHD thesis, EPFL, Lausanne, Switzerland (2006)
8. Pyeatt, L.D., Howe, A.E.: Decision tree function approximation in reinforcement learning. In: Proceedings of the Third International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Models, pp. 70–77 (2001)
9. Brodley, C.E., Utgoff, P.E.: Multivariate decision trees. Machine Learning 19, 45–77 (1995)
10. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Technical report, Wadsworth International, Monterey, CA (1984)
11. Murthy, S.K.: On Growing Better Decision Trees from Data, Ph.D. dissertation, Johns Hopkins Univ., Dept. Comput. Sci., Baltimore, MD (1995)
12. Heath, D., Kasif, S., Salzberg, S.: Learning Oblique Decision Trees. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence, Chambery, France, pp. 1002–1007. Morgan Kaufmann, San Francisco (1993)
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
14. Tanner, B., White, A.: RL-Glue: Language-Independent Software for Reinforcement-Learning Experiments. Journal of Machine Learning Research, 2133–2136 (2009)

Improving Hierarchical Document Signature Performance by Classifier Combination

Jieyi Liao, B. Sumudu U. Mendis, and Sukanya Manna

School of Computer Science, Australian National University

{u4720443,sumudu.mendis,sukanya.manna}@anu.edu.au

Abstract. We present a classifier-combination experimental framework for part-of-speech (POS) tagging in which four different POS taggers are combined in order to get a better result for sentence similarity using Hierarchical Document Signature (HDS). It is important to abstract information available to form humanly accessible structures. The way people think and talk is hierarchical with limited information presented in any one sentence, and that information is always linked together to further information. As such, HDS is a significant way to represent sentences when finding their similarity. POS tagging plays an important role in HDS. But POS taggers available are not perfect in tagging words in a sentence and tend to tag words improperly if they are either not properly cased or do not match the corpus dataset by which these taggers are trained. Thus, different weighted voting strategies are used to overcome some of these drawbacks of these existing taggers. Comparisons between individual taggers and combined taggers under different voting strategies are made. Their results show that the combined taggers provide better results than the individual ones.

Keywords: Part-of-Speech Taggers, Different Tagging methods, Hierarchical Document Signature.

1 Introduction

Sentences are considered to be the same if they share the same or similar meaning. There are different methods to find sentence similarities, and HDS is one of them. Since HDS uses the word level similarity to propagate to the sentence level, it is essential to identify these words based on their proper parts of speech; as it is known that the same word in different parts of speech may mean differently. Thus it is necessary to formulate a method which disambiguates the interpretational uncertainties and delivers more contextually appropriate word similarity - taking into account different measures, such as - graph based measures and glosses of the contextual senses,. Previous work in this area used neural networks [3], fuzzy relational maps [4] and fuzzy probability [5]. Hierarchical Document Signature (HDS) [6] [7] was chosen. It organizes a sentence into a hierarchical structure that helps to compare two or more sentences based on their part-of-speech (POS) using fuzzy aggregation. As such, its result is effected by the result of POS tagging directly.

It is known that each type of learning method has its own bias. It has been shown that, when the errors are uncorrelated to a sufficient degree, a combined classifier can often produce a better result than any individual one [8] [9]. This is a reflection of the bias/variance tradeoff [10], with a combined classifier providing a better balance between bias and variance if the separate classifiers are different in functionality. In this way, combining different taggers can make the classifier more robust to the quirks of each tagger's particular bias. Also, using information about each individual tagger's behaviour may be able to, in principle, fix the collective errors.

In this approach, we use different individual taggers and combine them under different voting strategies to see whether combined taggers can provide better results for sentence similarity using HDS. The remainder of this paper is organized as followed: section 2 gives brief introduction of the four individual taggers and compares our work to some previous work, section 3 describes the main algorithms used in this approach, section 4 analyses the experimental results in detail.

2 Related Work on Taggers

Four well-known English language taggers are chosen based on availability and published accuracies. In this paper, accuracy refers to the percentage of data tagged correctly by a given tagger when compared to the pretagged result. The OpenNLP POS tagger [11], based on the maxent maximum entropy package [18], gives an accuracy > 96% ; the Lingpipe tagger [12], based on a statistical model, gives detailed document and allows flexible usage; the Hunpos tagger [13], which is an open source reimplementation of TnT [14], gives an accuracy of 96.58%; finally, the CRFTagger, which is the Conditional Random Fields POS tagger built upon FlexCRFs, gives an accuracy of 97% [15].

There are some combined POS taggers such as [16] [17] [19] and [20]. Their results have shown that the combined POS taggers performed better than the individual taggers. Our work is different from [16] [17] because their experiments were based on Swedish or Telugu rather than English. Our approach builds upon the work on [19] [20]'s by using new individual taggers. Additionally it tests the result on both the word and sentence levels (using HDS).

3 Method

3.1 Preprocessing

Mapping for Tags. It is shown that different individual taggers may use different labels (tags) to represent the same POS. The four individual taggers in this approach use different sets of tags (tagsets). For the convenience of future use all combined taggers in this approach use The Penn Treebank POS tagset [21] as a standard. Therefore, it is important to match different tags with the same meaning in the preprocessing phase.

Mapping for Chunks. It is also shown that different taggers not only approximate different tags to the same word, but also divide a sentence into different

numbers of words in some cases. To solve this, a step of checking chunks is executed before voting where a sentence is divided into different chunks based on “space”.

For example, given a sentence:

The population of south-east Greenland is 7.5 million.

After the dividing step in this preprocessing part, the chunks are:

The/ population/ of/ south-east/ Greenland/ is/ 7./ .5/ million.

While some taggers may divide the sentence into different chunks as below:

The/ population/ of/ south/ -/ east/ Greenland/ is/ 7/ ./ 5/ million.

Let *Chunk-standard* be the set of chunks produced by this preprocessing and *Chunk-individual* be the set of chunks produced by the chunker in a certain individual tagger. The pseudocode representation of *chunk_mapping* is shown in Algorithm 1.

Algorithm 1. *chunk_mapping*

```

number_standard_chunk ← number of chunks in Chunk-standard
number_individual ← number of the chunks in Chunk-individual
i ← 0
s ← 0
for s < number_standard_chunk do
    word_stand ← Chunk-standard[s]
    word_individual ← Chunk-individual[i]
    if word_stand and word_individual is the same then
        Use the tag of word_individual to vote
        s ++
        i ++
    else
        Use the tag of word_individual to vote
        s ++
        word_stand ← Chunk-standard[s]
        while word_stand and word_individual are not the same do
            i ++
            word_individual ← Chunk-individual[i]
        end while
    end if
end for

```

In this approach, only the tag of the first part of the combined words is taken into account. Take the word “south-east” to illustrate. Assume it is divided into three chunks by a tagger. This means the tagger suggest three separate tags for the same word. In this case, the *Chunk_mapping* will match these three chunks (“south”, “-”, and “east”) to one word (“south-east”). Only the first suggestion (the tag for “south”) is considered when voting (section 3.2).

This process may lose some information from the individual taggers. This will be discussed to a greater depth in section 4.

3.2 Voting Schemes/Methods

There are many ways to combine the results of the component taggers and select a single tag from the set proposed by these taggers. The most straightforward and simple selection method is an n-way vote. Each component tagger votes for the tag of its own result and the tag with majority votes of the votes is selected as the final choice.

In this approach, experiments using two different voting methods are conducted. One is *equal voting*, the other is *weighted voting*. The following subsections will discuss these two voting methods separately.

Equal Voting. In the first experiment we use the democratic option, which is to give each tagger one vote (equal voting). In the equal voting system, votes from different component taggers are equally weighted. This allows all the taggers to have an equal chance to influence the final tag. As a result the equal voting classifier is fair and without bias to any component tagger. As such, it can be used to decide the weights for different component taggers in the experiments in weighted voting classifiers.

Weighted Voting. Equal voting is a fair strategy. But weighted voting can usually provide better results as it takes the qualities of the taggers into account. Also, it is more likely to disambiguate the result from “draw”, say two taggers vote a word as a noun, while the other two vote it as a verb.

In this subsection, two different weighted voting methods will be discussed: single weighted voting and multiple weighted voting.

In single weighted voting method, weights are assigned based on the accuracy of each tagger. Assuming a dataset in which data is correctly tagged is available, the accuracy of each tagger can be calculated.

In the method above, weights of the component taggers are based on their overall performance. Nevertheless, a tagger may have better performance in tagging certain kinds of words. For example, assume that tagger *A* achieves a better result than tagger *B* when tagging verbs. When it comes to adjectives, tagger *B* performs better than tagger *A*. If tagger *A* has a higher accuracy based on the standard in the previous method, then tagger *A* always has a higher weight when it is obviously not a good choice in tagging adjectives.

To overcome this, a vector of weights (multiple weighted voting method) is proposed: for each component tagger and for each tag, an accuracy is calculated. It also takes false positives into account to calculate appropriate accuracies. With a larger dataset the accuracies are likely to be less extreme, making a large dataset highly important to getting an accurate weight. These accuracies provide details on the performances in tagging different kinds of words. A n-dimension weight which contains the whole set of accuracy is assigned to each component tagger. The pseudocode for multiple weighted voting method is shown in Algorithm 2.

Algorithm 2. Multiple weighted voting method

```

 $n \leftarrow$  number of words in correctly tagged dataset
 $t \leftarrow$  number of tags in tagset
 $c \leftarrow$  tag for each word in correctly tagged dataset
 $i = 0$ 
while  $i < t$  do
     $number\_correct[i] = 0$ 
     $out\_of\_correct[i] = 0$ 
     $i++$ 
end while
 $i = 0$ 
while  $i < n$  do
     $out\_of\_correct[c[i]]++$ 
    if tagger's tag ==  $c[i]$  then
         $number\_correct[tagger's.tag]++$ 
    else
         $out\_of\_correct[tagger's.tag]++$ 
    end if
     $i++$ 
end while
 $i = 0$ 
while  $i < t$  do
     $accuracy[i] = \frac{number\_correct[i]}{out\_of\_correct[i]}$ 
     $i++$ 
end while

```

4 Experimental Results and Discussion

4.1 Experimental Results at Word Level

Experiments in this subsection are based on CoNLL2000 [22] dataset. Table 1¹ shows the accuracies of individual taggers and their combinations. It shows that the combined taggers yield a substantial improvement over any individual tagger.

The average accuracy of combined taggers is 97.27% compared to 93.23% of the individual taggers. This large improvement implies that combining taggers makes a classifier more robust to each tagger's particular bias and helps compensates for the weakness of each tagger. The taggers using weighted voting method give better performances than the one using equal voting method. This indicates that the weights allow the taggers to become more robust to each

¹ EqualVoting is the tagger using equal voting method; SingleW1 is the tagger using single weighted voting method and use the result of equal vote tagging method as a standard when obtaining the weight; MultiW1 is the tagger using multiple weighted voting method and use the result of equal vote tagging method as a standard when obtaining the weight; SingleW2 is the tagger using single weighted voting method and use the pretagged result from the dataset as a standard; MultiW2 is the tagger using multiple weighted voting method and the pretagged result from the dataset as a standard.

Table 1. Accuracy of the taggers

Tagger	Lingpipe	OpenNLP	CRFTagger	Hunpos
accuracy	82.72%	97.08%	96.85%	96.25%
Tagger	EqualVoting	SingleW1	MultiW1	SingleW2
accuracy	97.18%	97.25%	97.29%	97.26% MultiW2 97.37%

individual tagger's particular bias. The tagger using multiple weighted voting method outperforming the one using single weighted voting method meets my hypothesis. From this it is fair to conclude that additional information granted by the multiple weights was successfully used to allow the taggers to help compensate for one another. Inspite of using different standards when calculating the weights, the taggers using single weighted method have very similar accuracy. This implies that there is little advantage in using the dataset's tagged result rather than that of the equal voting. This further implies that the equal voting result is sufficient to determine the weights for the simple weighted voting method. There is however a more significant improvement between taggers using multiple weighted voting method, indicating that the multiple weighted method is able to make more use of the dataset's tagged result. This could be explained by the greater ability to fine-tune the weights. Compare the result of the best individual tagger (97.08%) and the best combined tagger (97.37%), the improved efficiency is 12.7%. From this it is fair to conclude that it is advantageous to train the multiple weighted voting method with a dataset's true tags.

4.2 Experimental Results at Sentence Level

The experiments in this subsection are conducted based on Microsoft Research Paraphrase Corpus 1.0 [23]. Table 2 shows the results for each tagger using HDS. We use the same evaluation standards (Recall, Precision, F1) as the ones in [7]. Note that as there are no available tagged results from this dataset, the result of the equal voting method is assumed to be good enough to use as a standard to calculate the weights for each individual tagger. It shows that the combined taggers yield a substantial improvement in recall over any individual tagger. This reinforces the conclusion that combining taggers make a classifier more robust to each tagger's particular bias and helps compensates for the weakness of each tagger. Among the combined taggers, the tagger using the equal voting method outperformed the others, which is opposite to our prediction.

This interesting result may be due to the following reasons. First, HDS only takes proper nouns (PP, PPS), common nouns (NN, NNS), adjectives (JJ, JJS) and verbs (VB, VBS, etc) into account and ignores other kinds of words. As it can be seen in the simple weighted voting method, when assigning weight, the method takes all kinds of words, including symbols, into account and gives a weight based on the average performance. Therefore, the tagger which has the highest weight may not be the one which performs best in tagging the kinds of words mentioned above. This can also explain why the combined tagger using the multiple weighted method does not yield any improvement over the one

using the single weighted method. If the performance of a tagger in tagging nouns, verbs and adjectives is similar to the average performance which takes all the tags into account, then those two different weighting methods do not make much difference in the result when using HDS evaluation. This is because the information on tagging other tags does not affect the performance in this case. Second, as the results generated by the tagger using the equal vote tagging method is assumed to be 100% correct when calculating the weights, the calculated weights for the multiple weighted voting method is not optimal, as has been shown by the experiments in the previous subsection. It would be better if a dataset which is manually tagged and checked is used as the standard in the weight finding process. In that case, the result of the combined tagger using the weighted voting method is more likely to perform better than the one using the equal voting method. The last but not least possible explanation is that during the preprocessing, more than one suggestions may be given by a tagger when tagging compound words, but only the first one is taken into account. The loss of information may result in an incorrect tag, and effects the final result of HDS.

Table 2. Result for each tagger using HDS

Tagger	Recall	Precision	F1
CRFTagger	0.9088	0.7022	0.7923
Hunpos	0.8853	0.7033	0.7839
Lingpipe	0.8853	0.7201	0.7942
OpenNLP	0.8971	0.7028	0.7881
EqualVoting	0.9147	0.7052	0.7964
SingleW1	0.9118	0.7029	0.7939
MultiW1	0.9118	0.7029	0.7939

5 Conclusion

The results in this approach indicate that the combined taggers outperform any of the individual taggers. The combined tagger using multiple weighted voting method gives the highest accuracy at word level which is 97.37% compared to 97.08% from the best component tagger. Our research concludes that combining different taggers is a good strategy to disambiguate the uncertainties in POS tagging and delivers more contextually appropriate word similarity. We also conclude that the use of properly tagged datasets for the calculation of weights provides little improvement on single weighted voting methods, but would have a more significant impact on multiple weighted voting methods.

References

1. Fellbaum, C.: Wordnet: an Electronic Lexical Database. Bradford Books (1998)
2. Gasperin, C., Gamallo, P., Agustini, A., Lopes, G., Lima, V.: Using syntactic contexts for measuring word similarity. In: Proceedings of the Workshop on Semantic Knowledge Acquisition and Categorisation, Helsinki, Finland (2001)

3. Gedeon, T.D., Mital, V.: Information Retrieval in Law using a Neural Network Integrated with Hypertext. In: Proceedings International Joint Conference on Neural Networks, Singapore, pp. 1819–1824 (1991)
4. Koczy, L.T., Gedeon, T.D., Koczy, J.A.: Fuzzy tolerance relations and relational maps applied to information retrieval. *Fuzzy Sets and Systems* 126(1), 49–61 (2002)
5. Huang, Z., Gedeon, T. D.: Information Retrieval Estimation Via Fuzzy Probability. World Automation Congress (WAC), Budapest, Hungary (2006)
6. Manna, S., Mendis, B., Gedeon, T.: Hierarchical document signature: A specialized application of fuzzy signature for document computing (2009)
7. Manna, S., Gedeon, T.: Hierarchical Document Signature for Semantic Analysis. In: WCCI 2010, FUZZ-IEEE 2010, Barcelona (2010)
8. Ali, K.M., Pazzani, M.J.: Error Reduction through Learning Multiple Descriptions. *Machine Learning* 24(3), 173–202 (1996)
9. Chan, P.K., Stolfo, S.J.: A Comparative Evaluation of Voting and Meta-Learning of Partition Data. In: 12th International Conference on Machine Learning (1995)
10. Gedeon, T.D., Wong, P.M., Harris, D.: Balancing Bias and Variance: Network Topology and Pattern Set Reduction Techniques. In: Sandoval, F., Mira, J. (eds.) IWANN 1995. LNCS, vol. 930, pp. 551–558. Springer, Heidelberg (1995)
11. OpenNLP,
[http://opennlp.sourceforge.net/api/opennlp/tools/lang/english/
PosTagger.html](http://opennlp.sourceforge.net/api/opennlp/tools/lang/english/PosTagger.html)
12. Lingpipe Tagger,
<http://alias-i.com/lingpipe/demos/tutorial/posTags/read-me.html>
13. Halacsy, P., Kornai, A., Oravercz, C.: HunPos -an open source trigram tagger. In: Proceedings of the Demo and Poster Session of the 45th Annual Meeting of the ACL, pp. 209–212 (2007)
14. Brants, T.: TnT-A Statistic Part-of-Speech Tagger. In: Proceedings of ANLP-NAACL Conference (2000)
15. CRFTagger, <http://crftagger.sourceforge.net>
16. Sjöbergh, J.: A Comparative Evaluation of Voting and Meta-Learning of Partition Data. In: Proceedings of RANLP-2003, Borovets, Bulgaria (2003)
17. Rama Sree, R.J., Kusuma kumari, P.: Combining Pos Taggers For Improved Accuracy To Create Telugu Annotated Texts For Information Retrieval. In: ICUDL 2007, Carnegie Mellon University, Pittsburgh, USA-ULIB, (2007)
18. OpenNLP MAXENT, <http://maxent.sourceforge.net>
19. van Halteren, H.: Comparison of Tagging Strategies, a Prelude of Democratic Tagging. *Research in Humanities Computing* 4, 207–215 (1996)
20. Van Halteren, H., Zavrel, J., Daelemans, W.: Improving Data Driven Worldclass Tagging by System Combination. In: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (1998)
21. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the penn treebank. *Computational Linguistics* 19 (1993)
22. CoNLL2000 dataset, www.cnts.ua.ac.be/conll2000/chunking/
23. Microsoft Research Paraphrase Corpus,
[http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-
85bc-a2f65cd28042/](http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042/)

The Discovery of Hierarchical Cluster Structures Assisted by a Visualization Technique

Ke-Bing Zhang¹, Mehmet A. Orgun¹, Yanchang Zhao², and Abhaya C. Nayak¹

¹ Department of Computing, Macquarie University, Sydney, NSW 2109, Australia

{kebing.zhang, mehmet.orgun, abhaya.nayak}@mq.edu.au

² Centrelink Australia

yanchang.zhao@centrelink.gov.au

Abstract. Hierarchical clustering is very versatile in real world applications. However, due to the issue of higher computational complexity from which automated hierarchical clustering algorithms suffer, the user can hardly correct possible misclassifications from the tree-structured nature of clusters. Visualization is a powerful technique for data analysis, however, most of the existing cluster visualization techniques are mainly used for displaying clustering results. In order for the user to be directly involved in the process of discovering nested cluster structures, we introduce a visualization technique, called HOV³, to detect clusters and their internal cluster structure. As a result, our approach provides the user an effective method for the discovery of nested cluster structures by visualization.

Keywords: Visualization, cluster analysis, hierarchical cluster discovery.

1 Introduction

Cluster analysis aims at finding groups in data by given similarity measurements. There are two main categories of clustering approaches, (1) partitioning methods and (2) hierarchical methods [7]. In hierarchical clustering, clusters can have data points or representatives of lower-level clusters which may be produced by different similarity measurements i.e., centroid distance, average distance, single-link distance and complete-link distance. Hierarchical methods are more versatile than partitioning methods, but they have higher computational complexity (of between $O(N^2)$ to $O(N^3)$). And also, Kaufman and Rousseeuw commented that “*a hierarchical method suffers from the defect that it can never repair what was done in previous steps*” [7]. Another issue is that, the outcomes of most existing hierarchical clustering algorithms are unstable due to their sensitivity to noise and outliers in data. As a consequence, hierarchical clustering algorithms and methods are not always effective to discover the nested cluster structure of real-world applications.

Clustering is an iterative process to discover the optimum (or an optimal) cluster structure in data for a particular application based on the experts’ assessment. Visualization can play an important role in cluster analysis as it can leverage human decision making during the process of clustering. A hierarchical cluster structure is often visualized as a dendrogram in graphics. But a dendrogram is only suitable to display a

small number of observations [12]. Several visualization techniques have been developed to display tree-structured clusters. However, instead of being an interactive mechanism to assist the user in the process of hierarchical cluster discovery, those techniques focus on representing clustering results. This motivates us to develop a novel approach for assisting the user on the discovery of hierarchical cluster structures by visualization.

We have employed a visual technique, *Hypothesis Oriented Verification and Validation by Visualization (HOV³)* [21] to assist the user on partitioning data during cluster exploration [19]. This research builds on the work in [19] further to address hierarchical cluster discovery, where a dataset is firstly projected by HOV³ on a 2D space to have several well-separated groups, and then this process is applied to each (sub)group of the dataset and its internal subgroups recursively. As a result, the user would have tree-structured clusters of the dataset.

The rest of this paper is organized as follows. Section 2 briefly introduces HOV³ and its features as the background of this study. Section 3 gives an overview of the HOV³-based visual approach on discovering clusters in a hierarchy. Section 4 demonstrates the effectiveness of our approach on a benchmark dataset. Section 5 presents a short review of the existing approaches to hierarchical clustering and hierarchical cluster visualization as the related work of this research. Finally, section 6 concludes this paper with a brief summary of our contributions.

2 An Overview of the HOV³ Technique

HOV³ is a visualization technique to project high dimensional data to a two dimensional space [21]. For completeness, we briefly discuss the features of HOV³ below.

2.1 Data Projection by HOV³

The idea of HOV³ is that it extends the perspective of traditional orthogonal X-Y coordinates to a higher dimensional space [21]. In particular, HOV³ divides a 2D circle into n equal sectors with n coordinate axes, where each axis represents a dimension and all axes share the initials at the centre of the circle. First, the data in each dimension are normalized over the interval of $[0, 1]$. Then the values of all axes are mapped to orthogonal X-Y coordinates which share the initial point on a 2D Star coordinates space [6]. Thus, an n -dimensional data item is represented by a point in the 2D circle. Fig.1 illustrates an example of mapping an 8-dimensional item $a = (0.25, 0.75, 0.40, 0.19, 0.85, 0.90, 0.60, 0.40)$ to 2D space by HOV³, where point “ a ” is the mapped location.

To project high-dimensional data onto a two-dimensional surface, we adopt the Polar Coordinates representation. Thus any dimensional vector can be easily transformed to the orthogonal axes X and Y. We use analytic geometry to describe the

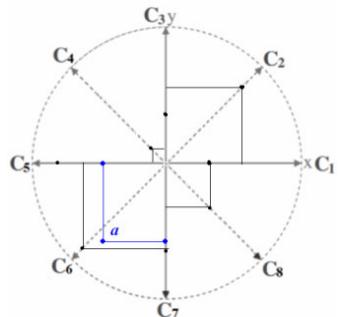


Fig. 1. Projecting an 8-dimensional item to 2D Star coordinates space.

HOV3 model mathematically. The element-wise product of vector A and vector B (mathematically written as $A \circ B$) expresses the difference between them, and its geometrical meaning captures the data distribution plotted by vector A against vector B (and vice versa). Therefore the element-wise product between a dataset and a measure vector in HOV3 can be geometrically viewed as a data distribution plotted by a matrix against the measure vector in the HOV3 space.

For more details of our work, we introduce the complex number system into our study. Let $z = x + iy$, where i is the imaginary unit. According to the Euler formula: $e^{ix} = \cos x + i \sin x$. Let $z_0 = e^{2\pi i/n}$ such that $z_0^1, z_0^2, z_0^3, \dots, z_0^{n-1}, z_0^n$ (with $z_0^n = 1$) divide the unit circle on the complex 2D plane into n equal sectors. Then given a non-zero measure vector M in \mathbb{R}^n and a family of vectors P_j , the projection of P_j against M is mathematically written as:

$$P_j(z_0) = \sum_{k=1}^n [(d_{jk} - \min_k d_k) / (\max_k d_k - \min_k d_k) \cdot z_0^k \cdot m_k] \quad (1)$$

where $\min_k d_k$ and $\max_k d_k$ represent the minimal and maximal values of the k th coordinate respectively; and $(d_{jk} - \min_k d_k) / (\max_k d_k - \min_k d_k)$ is the normalized value of d_{jk} ; m_k is the k th variable of measure M .

As shown in equation (1), the projection of HOV³ can be viewed as a mapping from high-dimensional real space to 2D complex number space ($\mathbb{R}^n \rightarrow \mathbb{C}^2$) with measure M , where the real and imaginary parts are mapped to the X axis and the Y axis respectively. In our study, if data points a and b are closely plotted by HOV³ with a measure vector m , then it is easy for the user to interpret the result by analyzing m . This is because HOV³ provides a quantitative mechanism on the plot of data.

2.2 Features of HOV³

As we explained above, the features of HOV³ are summarized as follows:

Data Order Independence - The location of a high dimensional data item projected by HOV³ is only decided by its values formed in a vector, see equation (1), but not related to its order in the dataset. This feature is important, since most existing hierarchical clustering algorithms are sensitive to the order of data [7]. Therefore, cluster discovery based on HOV³ can find truly hierarchical clusters

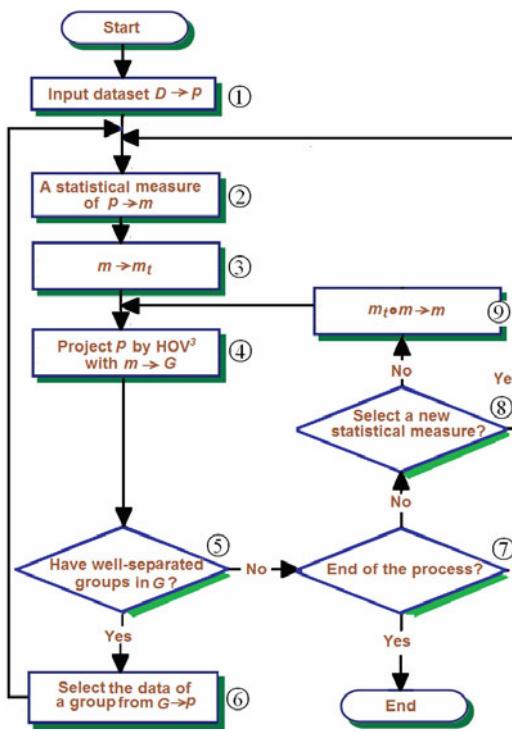
Enhanced Separation - We have shown that, if there are several data point groups that can be roughly separated by applying a measure vector m in HOV³ to a dataset, then multiple applications of the projection in HOV³ with the same measure vector m to the dataset would lead to the groups being more condensed, in other words, we have a good separation of the groups [22]. This feature is significant for cluster exploration by HOV³, since a clear geometrical separation of data points provides the user an intuitive insight on the cluster structure.

Linear Time Complexity - Equation (1) is a standard form of linear transformation of n variables, where m_k is the coefficient of the k th variable of P_j . Since the projection process of HOV³ has only linear time complexity, HOV³ is very effective for the interactive exploration of very large datasets.

Quantitative Exploration - The HOV³ technique adopts the quantified user domain knowledge as a prediction to discover clusters [21]. The quantitative mechanism of HOV³ avoids the randomness and subjectivity of the user during the process of cluster exploration by visualization.

3 Discovering Hierarchical Cluster Structures by HOV³

This work only considers the divisive hierarchical clustering process, i.e., clustering from whole observations to each data item, since the top-down exploration approach is a more natural cognitive process for humans. A divisive hierarchical clustering model builds a sequence of partitions in which each partition is nested within the next-level partition in the sequence. According to Shneiderman's visualization mantra of “Overview, zoom and filter, details on demand” [14], we first apply HOV³ to project a dataset with its statistical summaries in order to obtain the groupings of the dataset. Then this process is performed recursively to each group and its offspring subgroups for discovering lower level nested groups until a termination criterion is reached. To explain our approach, we give an outline of the process of the discovery of hierarchical clusters by HOV³ into a flowchart as explained follows. We present an example to demonstrate how HOV³ works on discovering hierarchical clusters and exhibit its effectiveness in the following section.



- (1) store a dataset D to variable p ;
- (2) select a statistical measure, such as mean, standard deviation, covariance matrix, etc., to m which is used as a measure in HOV³ to project dataset p ;
- (3) prepare a copy of m ;
- (4) G is the data distribution projected with measure m ;
- (5) if there exist well-separated groups in G , then go to the next step; otherwise go to step (7);
- (6) select/circle a group and store the data of the group to p ; then go back to step (2) to repeat the loop;
- (7) make a decision on whether to terminate the process; if yes, end the process; otherwise go to the next step;
- (8) if there a need to select a new statistical measure of p , then go to step (2), else go to the next step;
- (9) choose a measure to apply such as multiple times of m , for example, double times, triple times of m ;

Fig. 2. The flowchart of the discovery of hierarchical clusters by HOV³

4 The Experiments

In this section, we discuss several experiments on the discovery of nested clusters by HOV³ and demonstrate its effectiveness. We implemented the experiments in MATLAB running under Windows XP Professional, where we directly employ the statistical summaries of data as the measurements to project the data by HOV³. Divisive hierarchical cluster exploration can be performed in HOV³ by drilling down the nested groups in a top-down fashion. We present a worked example to demonstrate this process as follows.

Breast-cancer-wisconsin dataset has 11 attributes and 699 instances¹. We omit the first attribute of the dataset is *ID number*, which is not related with data analysis, and use the data with its other attributes (*breast-cancer-wisconsin-10*). Its original data distribution plotted by HOV³ is given in Fig. 3. We first employ the mean value of the dataset, $b_mean = [0.37975, 0.23716, 0.24527, 0.20076, 0.24622, 0.32761, 0.27086, 0.20744, 0.06549, 0.34478]$, as a statistical prediction to project the dataset. The resulting data distribution is plotted in Fig.4, where no clear groups can be observed.

Then we project the dataset by HOV³ by using four times the element-wise product of b_mean ; the projected result is shown in Fig.5. We can observe two groups of data points. We group them as two clusters (cluster 1 has 457 instances, and cluster 2 has 242 items) as shown in Fig.6. By analyzing the instances of clusters, we have found that the clusters are distinguished by their “class” values, where value “2” represents benign, and “4” means malignant.

We next choose cluster C1 and remove the attribute “class” from it for further cluster exploration by HOV³. Thus C1 now has 9 attributes and 457 instances. The original data distribution of C1 is shown in Fig. 7, where we cannot recognize any groups. Then we employ the first row of C1’s covariance matrix, $C1cov1 = [0.0571, 0.0080, 0.0104, 0.0071, 0.0043, 0.0005, 0.0053, 0.0071, -0.0007]$ as a statistical prediction to detect clusters of sub-clusters of C1 by HOV³. Its data distribution is presented in Fig. 8, where several groups can be observed, but the boundary of the groups in Fig. 8 is uncertain.

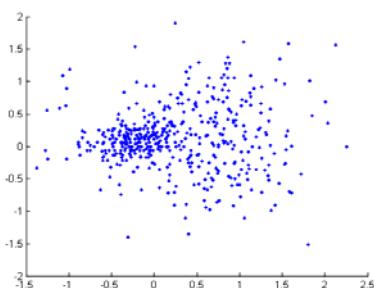


Fig. 3. the original data distribution of breast-cancer-wisconsin-10 projected by HOV3

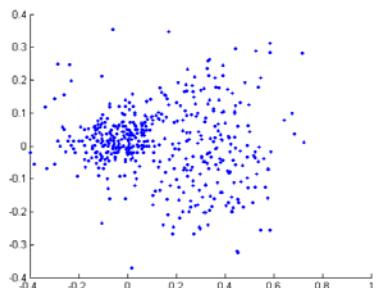


Fig. 4. the data distribution of breast-cancer-wisconsin-10 projected by HOV3 with its mean as the statistical prediction

¹ Available from <http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/>

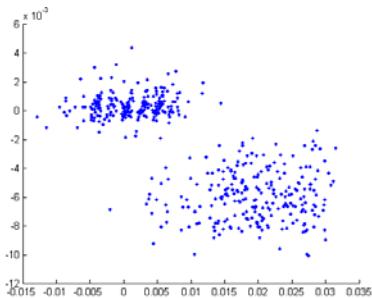


Fig. 5. The data distribution of *breast-cancer-wisconsin-10* projected by HOV^3 with four times mean as the prediction

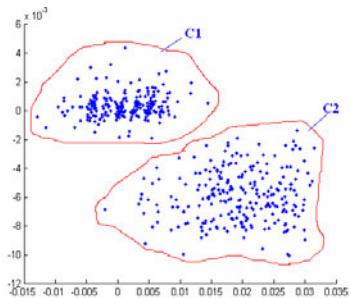


Fig. 6. The clustered distribution of data in Fig. 4 by the user's intuition

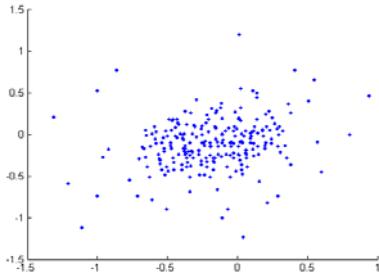


Fig. 7. The data distribution of the cluster C1 in *breast-cancer-wisconsin-10* projected by HOV^3

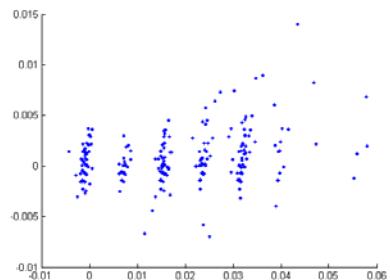


Fig. 8. The data distribution of the cluster C1 in *breast-cancer-wisconsin-10* projected by HOV^3 with $C1\text{cov}1$ as the statistical prediction

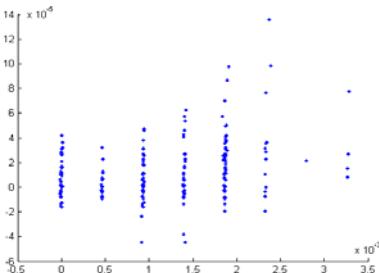


Fig. 9. The data distribution of the cluster C1 in *breast-cancer-wisconsin-10* projected by $\text{HOV}3$ with twice $C1\text{cov}1$ as the prediction

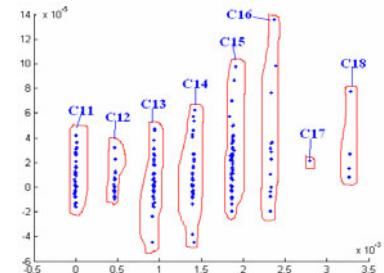


Fig. 10. The clustered distribution of data in Fig. 9 by the user's intuition

With the enhanced separation feature of HOV^3 , we apply $C1\text{cov}1$ twice to C1, and its data distribution is given in Fig. 9, where 8 groups can be clearly observed. We mark them as 8 sub-clusters of C1, as shown in Fig. 10. By analyzing the sub-clusters, we find that they are mainly grouped by the values of the attribute “*Clump Thickness*” in C1, i.e., the first attribute values in a sub-cluster are the same, because we used the first row of the covariance matrix of C1, $C1\text{cov}1$ as the statistical prediction.

Also, we can find that, in subgroup C16 (Fig. 10), the upper three data points are clustered into C16, rather than being clustered into C15 by automated hierarchical clustering algorithms based on the geometrical calculation, for example, distance-based clustering. In HOV³, these three points may be divided further as subgroups of C16. As mentioned above, the data points clustered by HOV³, as shown in Fig. 10, can be given a more reasonable explanation by a domain expert on this structure compared with the clustering results produced by automated hierarchical clustering algorithms. Moreover, the user can do the same detection to C2 or to the sub-clusters in C1 iteratively to perform further (hierarchical) cluster discovery.

We have also applied this approach to several other benchmark datasets, such as *shuttle* and *image* in a systematic study [20]. We have in particular compared the clustering results produced by HOV³ with several datasets, *auto-mpg*, *parkinson's disease*, and *wine* accessed from the website <http://archive.ics.uci.edu/ml/machine-learning-databases/>, to those produced by several commonly used partitioning clustering algorithms [22, 19]. The clustering results of HOV³ are better than those of several automated clustering algorithms based on the experiments we have conducted. This means that HOV³ can have the local optimization of clustering in each level of the hierarchical clustering process. We believe that, as the above experiments with our approach show, the user would be able to interpret the clustering results more easily. This is important, because an essential purpose of data mining is to provide reasonable and meaningful data mining results.

5 Related Work

Several approaches have been proposed, such as BIRCH [23], CURE [4] CHAMELEON [8] and LIMBO [1] in order to deal with the issues of hierarchical clustering as mentioned in Section 1. However, these approaches still suffer from more or less those same issues. For example, the computational complexity of CURE and CHAMELEON are still high. BIRCH and LIMBO have difficulty in choosing the proper parameters for their algorithms. A novel algorithm has been introduced to detect hierarchical clusters from fuzzy data [5], which is also tolerant and flexible to deal with noise and outliers. However, it is slow to handle very large datasets. A survey of recent works on hierarchical clustering algorithms can be found in the literature [7, 17].

Visualization is an important technique for data analysis. It is believed that, the combined strength of visualization and data mining could enrich both approaches and enable more successful solutions. Many approaches have been proposed to visualize the hierarchy of high-dimensional data. However, most of them simply regard visualizing data as a layout problem and focus on representing data in a more readable image/graphics, and not objectively revealing the actual relationships among data. A recent survey of data visualization techniques can be found in the literature [3].

Proposed 3D techniques are intuitive and expressive to represent the nested model of data. For example, H-BLOB [10] visualizes clusters into nested blobs in a 3D hierarchical structure. Yang [18] projects clusters in 3D animations. However, the complex 3D graph formation limits their capability on the interactive cluster discovery of very large-size data by visualization. The output of hierarchical clustering is a tree-structured graphics, named dendrogram. Several techniques have been proposed

to visualize hierarchical clusters in dendograms, for example, XmdvTool [16], HEC [13], Xgobi/Ggobi [11], TreeView [2], maxdView [9] and GrapClus [15]. However, dendrogram visualization is only suitable to visualize small sets of data. With its linear time computational complexity, Star Coordinates is a good choice for detecting cluster structures by visualization [6]. However, it is hard to interpret the grouping results produced by Star Coordinates because its cluster detection is based on the user's random adjustments of certain parameters.

6 Conclusions

In this work, we have applied a visualization technique called HOV³ to discover hierarchical cluster structures in high dimensional data. In the discovery process, the user first projects the original dataset into several groups based on the enhanced separation feature of HOV³ using the statistical features of the dataset. Then this process is applied to each group and its offspring subgroups recursively to any depth/level of hierarchy. The outcome of this process is a nested cluster tree. The HOV³ technique can be either employed on its own to discover hierarchical clusters or can be combined with an automated hierarchical clustering algorithm to assist the user in correcting the data points misclassified by the clustering algorithm. As our experiments have shown, HOV³ is an effective visualization method on the discovery of hierarchical clusters.

References

1. Andritsos, P., Tsaparas, P., Miller, R.J., Sevcik, K.C.: Limbo: scalable clustering of categorical data. In: Naves, G., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Sonnerat, N. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 123–146. Springer, Heidelberg (2004)
2. Eisen M.: Cluster and TreeView Manual (2007), <http://rana.lbl.gov/manuals/ClusterTreeView.pdf>
3. Fry, B.: Visualizing Data, Exploring and Explaining Data with the Processing Environment. O'Reilly Media, Sebastopol (2008)
4. Guha, S., Rastogi, R., Shim, K.: CURE: An efficient clustering algorithm for large databases. In: Proceedings of ACM SIGMOD Conference 1998, pp. 73–84 (1998)
5. GhasemiGol, M., Yazdi, H.S., Monsefi, R.: A New Hierarchical Clustering Algorithm on Fuzzy Data (FHCA). International Journal of Computer and Electrical Engineering 2(1), 1793–8163 (2010)
6. Kandogan, E.: Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In: Proc. of ACM SIGKDD Conference, pp. 107–116 (2001)
7. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data, An Introduction to Cluster Analysis. John Wiley and Sons, Inc., Chichester (2005)
8. Karypis, G., Han, E.-H.S., Kumar, V.: Chameleon: hierarchical clustering using dynamic modeling. IEEE Computer 32(8), 68–75 (1999)
9. <http://www.bioinf.manchester.ac.uk/microarray/maxd/maxdView/overview.html>

10. Sprenger, T.C., Brunella, R., Gross, M.H.: H-BLOB: A Hierarchical Visual Clustering Method Using Implicit Surfaces. In: Proc. of the conference on Visualization 2000, pp. 61–68. IEEE Computer Society Press, Los Alamitos (2000)
11. Swayne, D.F., Cook, D., Buja, A.: XGobi: Interactive dynamic data visualization in the X Window System. *Journal of Computational and Graphical Statistics* 7(1), 113–130 (1998)
12. Schonlau, M.: Visualizing non-hierarchical and hierarchical cluster analyses with clustergrams. *Journal of Computational Statistics* 19(1), 95–111 (2004)
13. Seo, J., Shneiderman, B.: Interactively Exploring Hierarchical Clustering Results. *IEEE Computer* 35(7), 80–86 (2002)
14. Shneiderman, B.: Inventing discovery tools: Combining information visualization with data mining. *Information Visualization* 1, 5–12 (2002)
15. Todd, C.S., Toth, T.M., Robert, B.-F.: GraphClus, a MATLAB program for cluster analysis using graph theory. *Journal of Computers and Geosciences* 35, 1205–1213 (2009)
16. Ward, M.O.: XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In: Proceedings of IEEE Conference on Visualization 1994, pp. 326–333 (1994)
17. Xu, R., Wunsch, D.C.: Clustering. John Wiley and Sons, Inc., Chichester (2009)
18. Yang, L.: Visual Exploration of Large Relational Data Sets through 3D Projections and Footprint Splatting. *IEEE Transactions on Knowledge and Data Engineering* 15(6), 1460–1471 (2003)
19. Zhang, K.-B., Huang, M.L., Orgun, M.A., Nguyen, Q.V.: A Visual Method for High-dimensional Data Cluster Exploration. In: Chan, J.H. (ed.) ICONIP 2009, Part II. LNCS, vol. 5864, pp. 699–709. Springer, Heidelberg (2009)
20. Zhang, K.-B., Orgun, M.A., Busch, P.A., Nayak, A.C.: A Top-down Approach for Hierarchical Cluster Exploration by Visualization. In: The Sixth International Conference on Advanced Data Mining and Applications (ADMA 2010), Xian, China, November 19–21 (2010) (accepted)
21. Zhang, K.-B., Orgun, M.A., Zhang, K.: HOV³, An Approach for Cluster Analysis. In: Li, X., Zaïane, O.R., Li, Z.-h. (eds.) ADMA 2006. LNCS (LNAI), vol. 4093, pp. 317–328. Springer, Heidelberg (2006)
22. Zhang, K.-B., Orgun, M.A., Zhang, K.: A Prediction-based Visual Approach for Cluster Exploration and Cluster Validation by HOV³. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 336–349. Springer, Heidelberg (2007)
23. Zhang, T., Ramakrishna, R., Livny, M.: An Efficient Data Clustering Method for Very Large Database. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 103–114 (1996)

Author Index

- Abe, Shigeo II-108
Ahi, Sercan Taha II-1
Amemiya, Yoshihito I-188
Ando, Ruo II-267, II-337
Aomori, Hisashi I-679
Araki, Osamu I-231
Asada, Minoru II-193
Asadpour, Masoud I-687
Asai, Tetsuya I-188
Asthana, Akshay II-485
Atsumi, Masayasu I-247
Azzag, Hanane I-652
- Bacciu, Davide I-660
Ban, Sang-Woo I-207, II-185
Ban, Tao II-143, II-259, II-267
Barczak, Andre L.C. I-438, II-291
Barros, Allan Kardec I-503
Basirat, Amir H. II-275
Becerra, J.A. I-567
Bellas, F. I-567
Bennani, Younès II-310, II-367
Biswas, Renuka II-590
Blachnik, Marcin II-700
Boedecker, Joschka II-193
Both, Fiemke I-270
Bouganis, Christos-Savvas II-501
Browne, Matthew II-209
Bui, Michael II-470
Byung, Kang II-337
- Caamaño, P. I-567
Cabanes, Guénaël II-367
Cao, Jian I-520
Cavalcante, Andre I-503
Chan, Jonathan H. II-250
Chen, Qingcai I-575
Chen, Ye II-283
Chen, Yin-Ju I-512, I-634
Chetty, Girija II-557, II-590
Cheung, Peter Y.K. II-501
Chi, Zheru I-239
Chin, Teo Chuan II-606
Cho, Sung-Bae II-234, II-650
- Choe, BongWhan II-650
Choi, Seungjin II-34
Chu, Pei-Hui I-512, I-634
Cichocki, Andrzej I-262, II-26,
II-34, II-74
Connie, Tee II-606
Constantinides, A.G. II-177
Creighton, Doug II-727
- Davies, Sergio I-58
De Souza, Alberto F. II-509
Dhall, Abhinav II-485
Diederich, Joachim I-330
Ding, Yuxin II-692
Dinh, Thanh Vu I-262
Dokos, Socrates I-140
Dong, Li II-692
Doya, Kenji I-215
Dozono, Hiroshi II-329, II-383
Duan, Fuqing II-525
Duch, Włodzisław I-282
Duro, R.J. I-567
Dürr, Volker II-676
- Ebrahimpour, Reza I-470
Ejiri, Ayato II-415
Elfwing, Stefan I-215
Elsayed, Saber M. I-585
Essam, Daryl L. I-585
Eto, Masashi I-290
- Fehervari, Tamas I-171
Feng, Dagan I-239
Filho, Hélio Perroni II-509
Fogassi, Leonardo II-17
Foon, Neo Han II-606
Fooproateepshri, Rerkchai II-549
Fujita, Kazuhisa I-148
Fukushima, Kunihiko II-574
Fukuzaki, Ryutaro I-82
Funase, Arao II-74
Fung, Chun Che II-152
Furber, Steve I-58
Furuhashi, Takeshi II-50

- Galluppi, Francesco I-58
 Garro, Beatriz A. II-201
 Gedeon, Tom I-298, II-66, II-124
 Ghaemi, Mohammad Sajjad I-470
 Glette, Kyrre I-540, I-642
 Goecke, Roland II-485
 Goodwin, Julian II-590
 Grozavu, Nistor II-310
 Gunasekara, Nuwan II-91
 Guo, Ping II-525, II-533
 Guo, Shanqing II-143, II-259
 Guo, William W. II-708
- Hada, Takahiro I-405
 Hamada, Toshiyuki I-223
 Hara, Satoshi I-422
 Hara, Shigeomi II-329, II-383
 Harada, Hidetaka II-321
 Hasegawa, Mikio I-49, I-66
 Hasegawa, Osamu II-344
 Hattori, Motonobu II-598
 Hawick, Ken A. I-438
 Hellbach, Sven II-676
 Herwik, Stanislav II-17
 Higashi, Hiroshi II-26
 Hirayama, Jun-ichiro I-371
 Hirose, Akira II-415
 Ho, Shiu-Hwei II-716
 Høvin, Mats I-540, I-642
 Hoogendoorn, Mark I-196, I-270
 Horiguchi, Yasuhito II-668
 Horio, Yoshihiko I-49
 Hoshino, Eiichi I-255
 Hosino, Tikara I-446
 Hosoya, Haruo I-1, I-33
 Hsu, Wen-Chuin II-462
 Huang, Chung-Hsien II-462
 Huang, Kaizhu I-494
 Hyon, Sang-Ho I-347
 Hyvärinen, Aapo I-371
- Ichisugi, Yuuji I-33
 Ijichi, Hirofumi I-107
 Ikeguchi, Tohru I-116
 Imam, Tasadduq II-116
 Imran, Nomica II-300
 Inoue, Daisuke I-290
 Iqbal, Ahmad Ali I-307
 Ishii, Shin I-371
 Ishikawa, Masumi I-609
- Islam, Tanvir I-82
 Itoh, Susumu I-679
 Itou, Shinsuke II-383
 Iwasa, Kaname I-74
 Iwasaki, Yuishi I-17
 Iwata, Akira I-74, II-407
 Iwata, Kazunori I-478
- Jalali, Sepehr II-541
 Jang, Young-Min I-207
 Jaruskulchai, Chuleerat I-559
 Jeatrakul, Piyasak II-152
 Jeong, Sungmoon II-185
 Jezzini, Ahmad II-17
 Ji, Yanli II-391
 Jin, Andrew Teoh Beng II-606
 Jitsev, Jenia II-616
 Jones, David Huw II-501
 Jung, Chanwoong II-185
- Kabashima, Takaru II-329
 Kadobayashi, Youki II-143,
 II-267, II-337
 Kambara, Hiroyuki II-1
 Kamei, Keiji I-609
 Kamimura, Ryotaro II-375, II-423
 Kaneko, Kunihiko I-155
 Kasabov, Nikola I-163, II-91, II-283
 Kashimori, Yoshiki I-124
 Kato, Hideyuki I-116
 Kawahara, Yoshinobu I-422
 Kawamura, Tetsuo I-49
 Kawashima, Manabu II-431
 Kawata, Hiroshi II-42
 Khan, Asad I. II-275, II-300
 Khosravi, Abbas II-727
 Kidode, Masatsugu II-668
 Kim, Hyung Chan I-290
 Kim, Kyung-Joong II-234
 King, Irwin I-397, I-669
 Kisban, Sebastian II-17
 Kitahara, Kunio II-415
 Kitamura, Takuya II-108
 Klein, Michel C.A. I-270
 Kobayashi, Takumi I-462, II-99
 Koike, Yasuharu II-1
 Kondo, Keiichi II-565
 Kong, Qi I-601
 Kopeć, Grzegorz II-700
 Krause, André Frank II-676

- Kubota, Naoyuki I-25
 Kugler, Mauricio I-74, II-407
 Kumada, Taichi I-9
 Kuramochi, Masahiro I-17
 Kurikawa, Tomoki I-155
 Kurogi, Shuichi I-363, II-352
 Kuroiwa, Jousuke I-223
 Kurokawa, Hiroaki I-179
 Kuroyanagi, Susumu I-74, II-407
 Kurutach, Werasak II-549
- Labiod, Lazhar II-310
 Lam, Ping-Man II-177
 Le, Trung II-132
 Lebbah, Mustapha I-652
 Lee, Jiann-Der II-462
 Lee, Minho I-207, I-430, II-185
 Lee, Nung Kion II-242
 Lee, Wono I-430
 Lenhardt, Alexander II-58
 Leung, Chi Sing II-160, II-177
 Li, Jianwu II-658
 Li, Lily D. II-708
 Li, Minglu I-520
 Li, Xi II-217
 Liao, Jieyi I-695
 Liao, Shu-Hsien I-512, I-634, II-716
 Lim, Chee Peng II-226
 Lim, Joo Hwee II-541
 Lin, Fengbo II-259
 Lin, Song II-533
 Lin, Tzu-Chao II-82
 Liu, Cheng-Lin I-494
 Liu, Mu-Kun II-82
 Liu, Tao I-314
 Liu, Wenhuang I-322
 Loth, Manuel I-454
 Lovell, Nigel H. I-140
 Lu, Bao-Liang I-601, II-493, II-625
 Lucena, Fausto I-503
- Ma, Wanli II-132
 Mączka, Krystian II-700
 Mak, Shue Kwan II-160
 Manna, Sukanya I-298, I-695
 Manoonpong, Poramate I-414
 Masoudnia, Saeed I-470
 Matsubara, Takamitsu I-347, II-668
 Matsuda, Ichiro I-679
- Matsuda, Yoshitatsu II-633
 Matsuoka, Masaru I-171
 Mayer, N. Michael II-209
 Meechai, Asawin II-250
 Melkumyan, Arman II-684
 Mendis, B. Sumudu U. I-298, I-695, II-124
 Michael, Goh Kah Ong I-532, II-606
 Micheli, Alessio I-660
 Midorikawa, Hiroki II-477
 Miki, Tsutomu II-321
 Min, Jun-Ki II-650
 Mineishi, Shota I-363
 Mo, Mingzhen I-669
 Moghadam, Gita Khalili I-140
 Mogi, Ken I-255
 Morimoto, Jun I-347, I-414
 Morita, Masahiko II-42
 Mouri, Motoaki II-74
 Mu, Yang I-338, II-641
 Murphy, Richard J. II-684
 Mus, Anne O. II-447
- Nagase, Yoshihiro I-124
 Nagi, Tomokazu II-352
 Nahavandi, Saeid II-727
 Nakakuni, Masanori II-329, II-383
 Nakao, Koji I-290
 Nam, Yunjun II-34
 Nayak, Abhaya C. I-703
 Neo, H.F. I-532
 Nishida, Takeshi II-352
 Nuntalid, Nuttapod I-163
- Obst, Oliver II-193
 Odaka, Tomohiro I-223
 Ogura, Hisakazu I-223
 Ohnishi, Noboru I-503
 Okumura, Keiji I-91, I-486
 Okuno, Hirotsugu I-171
 Oliver, Gareth II-66
 Ong, Sim Heng II-541
 Orgun, Mehmet A. I-703
 Osana, Yuko I-405, II-477
 Otake, Tsuyoshi I-679
 Otsu, Nobuyuki I-462, II-99
 Otsuka, Makoto I-215
- Pang, Shaoning II-91, II-283
 Papliński, Andrew P. II-360

- Paul, Oliver II-17
 Phan, Anh Huy I-262
 Phoomvuthisarn, Suronapee II-549
 Powell, Adam II-501
 Preux, Philippe I-454
 Prieto, A. I-567
 Prom-on, Santitham II-250
- Rahman, Ashfaqur I-551
 Rast, Alexander I-58
 Reyes, Napoleon H. II-291
 Ripon, Kazi Shah Nawaz I-540, I-642
 Ritter, Helge II-58
 Ruther, Patrick II-17
 Rutkowski, Tomasz II-26
- Saghezchi, Hossein Bashashati I-687
 Saitoh, Fumiaki II-399
 Sakai, Ko I-99, II-565
 Sakamoto, Kazuhiro I-9
 Samma, Ali II-226
 Samma, Hussein II-226
 Sarker, Ruhul A. I-585
 Sasaki, Hiroaki I-132
 Sato, Seitaro I-363
 Sato, Yasuomi D. I-91, II-616
 Satoh, Shunji I-132
 Schliebs, Stefan I-163
 Seidl, Karsten II-17
 Sekine, Masatoshi II-439
 Seneviratne, Aruna I-307
 Serventi, Francesca Ugoletti II-17
 Sharma, Dharmendra II-132
 Shen, Furao II-344
 Shi, Yongquan II-143
 Shiino, Masatoshi I-91, I-486
 Shimada, Atsushi II-391, II-431
 Shin, Heesang II-291
 Shirai, Haruhiko I-223
 Shu, Xinyi I-322
 Sim, K.S. I-532
 Singh, Alok I-626
 Singh, Monica II-557
 Song, Insu I-330
 Song, Jungsuk I-290
 Song, Mingli II-641
 Sootanan, Pitak II-250
 Sossa, Humberto II-201
 Sperduti, Alessandro I-660
 Stratton, Peter I-41
- Suaning, Gregg I-140
 Suemitsu, Atsuo II-42
 Sum, John II-160, II-177
 Sun, Shiliang I-355, II-9
 Sundar, Shyam I-626
 Susnjak, Teo I-438
 Suwa, Izumi I-223
 Szymański, Julian I-282
- Tajiri, Yasuyuki II-108
 Takahashi, Hiromu II-50
 Takatsuka, Masahiro II-470
 Takenouchi, Seiya I-679
 Takeuchi, Yoshinori I-503
 Takumi, Ichi II-74
 Tamukoh, Hakaru II-439
 Tanaka, Fumihide II-42
 Tanaka, Mamoru I-679
 Tanaka, Toshihisa II-26
 Tang, Buzhou I-575
 Tang, Dalai I-25
 Tang, Maolin I-618
 Tang, Yan-Ming II-493
 Taniguchi, Rin-ichiro II-391, II-431
 Tao, Dacheng I-338, I-388, II-641
 Tao, Yanyun I-520
 Tee, C. I-532
 Tekawa, Masaaki II-598
 Teo, C.C. I-532
 Tham, Jo Yew II-541
 Tickle, Kevin II-116
 Tjondronegoro, Dian II-582
 Torikai, Hiroyuki I-107
 Torresen, Jim I-540, I-642
 Tovar, Gessyca Maria I-188
 Tran, Dat II-132
 Treerattanapitak, Kiatichai I-559
 Treur, Jan I-196, I-270
 Tsuzuki, Hirofumi II-407
 Tu, Wenting II-9
- Uchibe, Eiji I-215
 Umiltà, Maria Alessandra II-17
 Usui, Shiro I-132
- van der Wal, C. Natalie I-196
 Van Dijck, Gert II-17
 Van Hulle, Marc M. II-17
 van Wissen, Arlette I-196
 Vázquez, Roberto A. II-201

- Verma, Brijesh I-551
Vo, Tan II-124
von Bünau, Paul I-422
von der Malsburg, Christoph II-616

Wagatsuma, Nobuhiko I-99
Wai, Yau-Yau II-462
Wang, Dianhui II-217, II-242
Wang, Dingyan I-397
Wang, Fengyu II-259
Wang, Jiun-Jie II-462
Wang, Kuanquan I-380
Wang, Liang II-525
Wang, Ling II-692
Wang, Xiaolong I-575
Wang, Xuan I-575
Washio, Takashi I-422
Washizawa, Yoshikazu II-26
Watanabe, Kenji I-462
Watanabe, Sumio II-399
White, Matthew II-557
Whymark, Greg II-708
Wieczorek, Tadeusz II-700
Wiles, Janet I-41
Wilke, Robert G.H. I-140
Wong, Kok Wai II-152
Wörgötter, Florentin I-414
Wu, Guohua II-692
Wu, Horng Jason II-209
Wu, Tian-Xiang II-625

Xie, Bo I-338, II-641
Xu, Chi I-239
Xu, Qiuliang II-143
Xu, Xue II-517
Xu, Zhijie I-355

Yabuwaki, Ryosuke II-108
Yagi, Tetsuya I-171
Yamaguchi, Kazunori II-633
Yamaguchi, Nobuhiko II-454
Yang, Gang II-517
Yang, Gelan II-517
Yang, Shih-Ting II-462
Yang, Wei I-380
Yang, Yujiu I-322
Yano, Masafumi I-9
Yao, Xin I-551
Yao, Yao II-533
Yeh, Chien-Ting II-82
Yonekawa, Masato I-179
Yoshihara, Masahiro I-179
Yoshikawa, Tomohiro II-50
Yoshimura, Natsue II-1
Yu, Hui II-344
Yuan, Qixia II-259
Yusoh, Zeratal Izzah Mohd I-618

Zang, Xu II-168
Zdunek, Rafal I-262
Zhang, Hong I-593
Zhang, Jianming II-517
Zhang, Ke-Bing I-703
Zhang, Ligang II-582
Zhang, Qing II-658
Zhao, Hai I-601
Zhao, Qibin II-34
Zhao, Xiaoyu I-239
Zhao, Yanchang I-703
Zhong, Guoqiang I-494
Zhou, Tianyi I-388
Zuo, Wangmeng I-380