

LNCS 5506

Mario Köppen
Nikola Kasabov
George Coghill (Eds.)

Advances in Neuro-Information Processing

15th International Conference, ICONIP 2008
Auckland, New Zealand, November 2008
Revised Selected Papers, Part I

1
Part I



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Mario Köppen Nikola Kasabov
George Coghill (Eds.)

Advances in Neuro-Information Processing

15th International Conference, ICONIP 2008
Auckland, New Zealand, November 25-28, 2008
Revised Selected Papers, Part I

Volume Editors

Mario Köppen

Network Design and Research Center, Kyushu Institute of Technology

680-4, Kawazu, Iizuka, Fukuoka 820-8502, Japan

E-mail: mkoeppen@ieee.org

Nikola Kasabov

Auckland University of Technology

Knowledge Engineering and Discovery Research Institute (KEDRI)

School of Computing and Mathematical Sciences

350 Queen Street, Auckland 10110, New Zealand

E-mail: nkasabov@aut.ac.nz

George Coghill

Auckland University of Technology, Robotics Laboratory

Department of Electrical and Computer Engineering

38 Princes Street, Auckland 1142, New Zealand

E-mail: g.coghill@auckland.ac.nz

Library of Congress Control Number: 2009929832

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

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

ISSN 0302-9743

ISBN-10 3-642-02489-0 Springer Berlin Heidelberg New York

ISBN-13 978-3-642-02489-4 Springer Berlin Heidelberg New York

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

springer.com

© Springer-Verlag Berlin Heidelberg 2009

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12692460 06/3180 5 4 3 2 1 0

Preface

The two volumes contain the papers presented at the ICONIP 2008 conference of the Asia Pacific Neural Network Assembly, held in Auckland, New Zealand, November 25–28, 2008.

ICONIP 2008 attracted around 400 submissions, with approx. 260 presentations accepted, many of them invited. ICONIP 2008 covered a large scope of topics in the areas of: methods and techniques of artificial neural networks, neuromcomputers, brain modeling, neuroscience, bioinformatics, pattern recognition, intelligent information systems, quantum computation, and their numerous applications in almost all areas of science, engineering, medicine, the environment, and business.

One of the features of the conference was the list of 20 plenary and invited speakers, all internationally established scientists, presenting their recent work. Among them: Professors Shun-ichi Amari, RIKEN Brain Science Institute; Shiro Usui, RIKEN Brain Science Institute, Japan; Andrzej Cichocki, RIKEN Brain Science Institute; Takeshi Yamakawa, Kyushu Institute of Technology; Kenji Doya, Okinawa Institute of Science and Technology; Youki Kadobayashi, National Institute of Information and Communications Technology, Japan; Sung-Bae Cho, Yonsei University, Korea; Alessandro Villa, University of Grenoble, France; Danilo Mandic, Imperial College, UK; Richard Duro, Universidade da Coruna, Spain, Andreas König, Technische Universität Kaiserslautern, Germany; Yaochu Jin, Honda Research Institute Europe, Germany; Bogdan Gabrys, University of Bournemouth, UK; Jun Wang, Chinese University of Hong Kong; Mike Paulin, Otago University, New Zealand; Mika Hirvensalo, University of Turku, Finland; Lei Xu, Chinese University of Hong Kong and Beijing University, China; Włodzisław Duch, Nicolaus Copernicus University, Poland; Gary Marcus, New York University, USA.

The organizers would also like to thank all special session organizers for their strong efforts to enrich the scope and program of this conference. The ICONIP 2008 conference covered the following special sessions: “Data Mining Methods for Cybersecurity,” organized by Youki Kadobayashi, Daisuke Inoue, and Tao Ban, “Computational Models and Their Applications to Machine Learning and Pattern Recognition,” organized by Kazunori Iwata and Kazushi Ikeda, “Lifelong Incremental Learning for Intelligent Systems,” organized by Seiichi Ozawa, Paul Pang, Minho Lee, and Guang-Bin Huang, “Application of Intelligent Methods in Ecological Informatics,” organized by Michael J. Watts and Susan P. Worner, “Pattern Recognition from Real-world Information by SVM and Other Sophisticated Techniques,” organized by Ikuko Nishikawa and Kazushi Ikeda, “Dynamics of Neural Networks,” organized by Zhigang Zeng and Tingwen Huang, “Recent Advances in Brain-Inspired Technologies for Robotics,” organized by Kazuo Ishii and Keiichi Horio, and “Neural Information Processing

in Cooperative Multi-Robot Systems,” organized by Jose A. Becerra, Javier de Lope, and Ivan Villaverde.

Another feature of ICONIP 2008 was that it was preceded by the First Symposium of the International Neural Network Society (INNS) on New Directions in Neural Networks (NNN 2008), held November 25–25, 2008. This symposium was on the topic “Modeling the Brain and Neurvous systems,” with two streams: Development and Learning and Computational Neurogenetic Modeling. Among the invited speakers were: A. Villa, J. Weng, G. Marcus, C. Abraham, H. Kojima, M. Tsukada, Y. Jin, L. Benuskova. The papers presented at NNN 2008 are also included in these two volumes.

ICONIP 2008 and NNN 2008 were technically co-sponsored by APNNA, INNS, the IEEE Computational Intelligence Society, the Japanese Neural Network Society (JNNS), the European Neural Network Society (ENNS), the Knowledge Engineering and Discovery Research Institute (KEDRI), Auckland University of Technology, Toyota USA, Auckland Sky City, and the School of Computing and Mathematical Sciences at the Auckland University of Technology. Our sincere thanks to the sponsors!

The ICONIP 2008 and the NNN 2008 events were hosted by the Knowledge Engineering and Discovery Research Institute (KEDRI) of the Auckland University of Technology (AUT). We would like to acknowledge the staff of KEDRI and especially the Local Organizing Chair Joyce DMello, the Web manager Peter Hwang, and the publication team comprising Stefan Schliebs, Raphael Hu and Kshitij Doble, for their effort to make this conference an exciting event.

March 2009

Nikola Kasabov
Mario Köppen
George Coghill

ICONIP 2008 Organization

ICONIP 2008 was organized by the Knowledge Engineering and Discovery Research Institute (KEDRI) of the Auckland University of Technology (AUT).

Conference Committee

General Chair	Nikola Kasabov
Program Co-chairs	Mario Köppen, George Coghill, Masumi Ishikawa
Publicity Chairs	Shiro Usui, Bill Howel, Ajit Narayanan, Suash Deb
Plenary Chairs	Takeshi Yamakawa, Andreas König, Tom Gedeon
Panels Chairs	Robert Kozma, Mario Fedrizzi, M.Tsukada, Stephen MacDonell
Tutorial Chairs	Sung-Bae Cho, Martin McGinnity, L. Benuskova
Special Session Chairs	Soo-Young Lee, Richard Duro, Shaoning Pang
Poster Session Chairs	Bernadete Ribeiro, Qun Song, Frances Joseph
Workshop Chairs	Mike Paulin, Irwin King, Kaori Yoshida, Napoleon H. Reyes
Demonstrations Chairs	Sue Worner, Russel Pears, Michael Defoin-Platel
Local Organizing Chair	Joyce Mello
Technical Support Chair	Peter Hwang

Track Chairs

- Neurodynamics: Takeshi Aihara, Tamagawa University, Japan
Cognitive Neuroscience: Alessandro Villa, UJF Grenoble, France
Brain Mapping: Jagath Rajapakse, Nanyang Technological University, Singapore
Neural Network Learning Paradigms: Nik Kasabov, Auckland University of Technology, New Zealand
Kernel Methods and SVM: Bernadete Ribeiro, University of Coimbra, Portugal
Ensemble Methods for Neural Networks: Andre C.P.L.F. de Carvalho, University of Sao Paulo, Brazil
Information Algebra: Andrzej Cichocki, RIKEN, Japan
Neural Networks for Perception: Akira Iwata, Nagoya Institute of Technology, Japan
Neural Networks for Motoric Control: Minho Lee, Kyungpook National University, Korea

VIII Organization

Neural Networks for Pattern Recognition: Paul Pang, Auckland University of Technology, New Zealand
Neural Networks for Robotics: Richard Duro, University of a Coruña, Spain
Neuromorphic Hardware: Leslie S. Smith, University of Stirling, UK
Embedded Neural Networks: Andreas Koenig, University of Kaiserslautern, Germany
Neural Network-Based Semantic Web, Data Mining and Knowledge Discovery: Irwin King, The Chinese University of Hong Kong, Hong Kong
Computational Intelligence: Włodzisław Duch, Nicolaus Copernicus University, Poland
Bioinformatics: Sung-Bae Cho, Yonsei University, Korea
Neural Paradigms for Real-World Networks: Tom Gedeon, The Australian National University, Australia
Quantum Neural Networks: Mika Hirvensalo, University of Turku, Finland
Neural Network Implementation in Hardware and Software: George Coghill, Auckland University of Technology, New Zealand
Biologically Inspired Neural Networks: Nik Kasabov, Auckland University of Technology, New Zealand

International Technical Committee

Abbass, Hussein	Coghill, George	Hayashi, Hatsuo
Abe, Shigeo	Cohen, Avis	Hikawa, Hiroomi
Aihara, Takeshi	Dauwels, Justin	Hirvensalo, Mika
Alippi, Cesare	de Lope, Javier	Honkela, Antti
Ando, Ruo	de Souto, Marcilio	Horio, Keiichi
Andras, Peter	Dorronsoro, Jose	Huang, Kaizhu
Asoh, Hideki	Dourado, Antonio	Ikeda, Kazushi
Ban, Tao	Duch, Włodzisław	Inoue, Daisuke
Bapi, Raju	Duro, Richard	Ishida, Fumihiko
Barczak, Andre Luis Chautard	Elizondo, David	Iwata, Kazunori
Barros, Allan Kardec	Erdi, Peter	Iwata, Akira
Becerra Permuy, José Antonio	Fukumura, Naohiro	Kadone, Hideki
Behera, Laxmidhar	Fung, Wai-keung	Kanoh, Shin'ichiro
Behrman, Elizabeth	Furukawa, Tetsuo	Kasabov, Nikola
Beliczynski, Bartłomiej	fyfe, colin	Kim, Kyung-Joong
Carvalho, Andre C.P.L.F. de	Garcez, Artur	Kimura, Shuhei
Chang, Jyh-Yeong	Gedeon, Tom	King, Irwin
Cho, Sung-Bae	Grana, Manuel	Kitajima, Tatsuo
Choi, Seungjin	Gruen, Sonja	Koenig, Andreas
Chung, I-Fang	Guo, Shangqing	Koeppen, Mario
Cichocki, Andrzej	Hagiwara, Katusyuki	Kondo, Toshiyuki
	Hammer, Barbara	Kurita, Takio
	Hartono, Pitoyo	Kurogi, Shuichi
	Hayashi, Akira	Lai, Weng Kin

Lee, Minho	Pang, Paul	Takenouchi, Takashi
Lendasse, Amaury	Patel, Leena	Tambouratzis, Tatiana
Lim, CP	Peters, Jan	Tanaka, Yoshiyuki
Liu, Ju	Phillips, Steven	Tang, Ke
Liu, Shih-Chii	Rajapakse, Jagath	Tateno, Katsumi
Lu, Bao-Liang	Reyes, Napoleon	van Schaik, Andre
Ludermir, Teresa	Ribeiro, Bernardete	Villa, Alessandro
Mandziuk, Jacek	Rueckert, Ulrich	Villaverde, Ivan
Matsui, Nobuyuki	Sakai, Ko	Wada, Yasuhiro
Mayr, Christian	Sato, Shigeo	Wagatsuma, Hiroaki
McKay, Bob	Sato, Naoyuki	Watanabe, Keigo
Meier, Karlheinz	Schemmel, Johannes	Watanabe, Kazuho
Mimura, Kazushi	Setiono, Rudy	Watts, Michael
Miyamoto, Hiroyuki	Shibata, Tomohiro	Wu, Jianhua
Molter, Colin	Shimazaki, Hideaki	Xiao, Qinghan
Morie, Takashi	Shouno, Hayaru	Yamaguchi, Nobuhiko
Morita, Kenji	Silva, Catarina	Yamauchi, Koichiro
Nakajima, Koji	Small, Michael	Yi, Zhang
Nakauchi, Shigeki	Smith, Leslie S.	Yoshimoto, Junichiro
Nguyen-Tuong, Duy	Spaanenburg, Lambert	Zhang, Zonghua
Nishii, Jun	Stafylopatis, Andreas	Zhang, Liming
Nishikawa, Ikuko	Suematsu, Nobuo	Zhang, Liqing
Ohnishi, Noboru	Suh, Il Hong	Zhang, Byoung-Tak
Omori, Toshiaki	Sum, John	
Ozawa, Seiichi	Suykens, Johan	

Additional Referees

Pong Meau Yeong, Hua Nong Ting, Sim Kok Swee, Yap Keem Siah, Shahrel Azmin Suandi, Tomas Henrique Bode Maul, Nor Ashidi Mat Isa, Haidi Ibrahim, Tan Shing Chiang, Dhanesh Ramachand Ram, Mohd Fadzli Mohd Salleh, Khoo Bee Ee

Sponsoring Institutions

Asia Pacific Neural Network Assembly (APNNA)
 International Neural Network Society (INNS)
 IEEE Computational Intelligence Society
 Japanese Neural Network Society (JNNS)
 European Neural Network Society (ENNS)
 Knowledge Engineering and Discovery Research Institute (KEDRI)
 Auckland University of Technology (AUT)
 Toyota USA
 Auckland Sky City
 School of Computing and Mathematical Sciences at the Auckland University of Technology

INNS NNN 2008 Organization

INNS NNN 2008 was organized by the Knowledge Engineering and Discovery Research Institute (KEDRI) of the Auckland University of Technology (AUT).

Conference Committee

General Chair	Juyang Weng
Program Chair	Nikola Kasabov
Program Co-chairs	Mario Koeppen, John Weng, Lubica Benuskova
Local Organizing Chair	Joyce DMello
Technical Support Chair	Peter Hwang
Publishing Committee	Stefan Schliebs, Kshitij Dhoble, Raphael Hu
Symposium 1 Co-chairs	Juyang Weng, Jeffrey L. Krichmar, Hiroaki Wagatsuma
Symposium 2 Co-chairs	Lubica Benuskova, Alessandro E.P. Villa, Nikola Kasabov

Program Committee

Alessandro E.P. Villa	Juyang Weng
Anil Seth	Kazuhiko Kawamura
wCharles Unsworth	Lubica Benuskova
Chris Trengove	Michael Defoin Platel
Cliff Abraham	Michal Cernansky
Danil Prokhorov	Ming Xie
Frederic Kaplan	Peter Jedlicka
Hiroaki Wagatsuma	Rick Granger
Igor Farkas	Rolf Pfeifer
James Wright	Roman Rosipal
Jason Fleischer	Xiangyang Xue
Jeffrey L. Krichmar	Zhengyou Zhang
Jiri Sima	

Reviewers

Danil Prokhorov	Kaz Kawamura
Ming Xie	Juyang Weng
Frederic Kaplan	Lubica Benuskova
Hiroaki Wagatsuma	Igor Farkas

Chris Trengove
Charles Unsworth
Roman Rosipal
Jiri Sima
Alessandro Villa
Peter Jedlicka
Michal Cernansky
Michael Defoin-Platel

Xiangyang Xue
Wickliffe Abraham
James Wright
Zhengyou Zhang
Anil Seth
Jason Fleischer
Nikola Kasabov

Sponsoring Institutions

Auckland University of Technology (AUT)
Asia Pacific Neural Network Assembly (APPNA)
International Neural Network Society (INNS)
Knowledge Engineering and Discovery Research Institute (KEDRI)
IEEE Computational Intelligence Society

Table of Contents – Part I

I INNS Symposium “New Directions in Neural Networks”

Integrative Probabilistic Evolving Spiking Neural Networks Utilising Quantum Inspired Evolutionary Algorithm: A Computational Framework	3
<i>Nikola Kasabov</i>	
A Spiking Network of Hippocampal Model Including Neurogenesis	14
<i>Yusuke Tabata and Masaharu Adachi</i>	
NeuroEvolution Based on Reusable and Hierarchical Modular Representation	22
<i>Takumi Kamioka, Eiji Uchibe, and Kenji Doya</i>	
A Common-Neural-Pattern Based Reasoning for Mobile Robot Cognitive Mapping	32
<i>Aram Kawewong, Yutaro Honda, Manabu Tsuboyama, and Osamu Hasegawa</i>	
Identifying Emotions Using Topographic Conditioning Maps	40
<i>Athanasios Pavlou and Matthew Casey</i>	
A Gene Regulatory Model for the Development of Primitive Nervous Systems	48
<i>Yaochu Jin, Lisa Schramm, and Bernhard Sendhoff</i>	
Real-Time Epileptic Seizure Detection on Intra-cranial Rat Data Using Reservoir Computing	56
<i>Pieter Buteneers, Benjamin Schrauwen, David Verstraeten, and Dirk Stroobandt</i>	
Learning of Subgoals for Goal-Oriented Behavior Control of Mobile Robots	64
<i>Sang Hyoung Lee, Sanghoon Lee, Il Hong Suh, and Wan Kyun Chung</i>	
Coding Mechanisms in Hippocampal Networks for Learning and Memory	72
<i>Yasuhiro Fukushima, Minoru Tsukada, Ichiro Tsuda, Yutaka Yamaguti, and Shigeru Kuroda</i>	
Developmental Stereo: Topographic Iconic-Abstract Map from Top-Down Connection	80
<i>Mojtaba Solgi and Juyang Weng</i>	

An Analysis of Synaptic Transmission and its Plasticity by Glutamate Receptor Channel Kinetics Models and 2-Photon Laser Photolysis	88
<i>Hiroshi Kojima and Shiori Katsumata</i>	
A Biologically Inspired Neural CPG for Sea Wave Conditions/Frequencies	95
<i>Leena N. Patel and Alan Murray</i>	
Feature Subset Selection Using Differential Evolution	103
<i>Rami N. Khushaba, Ahmed Al-Ani, and Adel Al-Jumaily</i>	
Topology of Brain Functional Networks: Towards the Role of Genes	111
<i>Mária Markošová, Liz Franz, and Ľubica Beňušková</i>	
Hybrid Design Principles and Time Constants in the Construction of Brain-Based Robotics: A Real-Time Simulator of Oscillatory Neural Networks Interacting with the Real Environment via Robotic Devices...	119
<i>Hiroaki Wagatsuma</i>	

II Neurodynamics

First Spiking Dynamics of Stochastic Neuronal Model with Optimal Control	129
<i>Yongjun Wu, Jianhua Peng, and Ming Luo</i>	
BCM and Membrane Potential: Alternative Ways to Timing Dependent Plasticity	137
<i>Johannes Partzsch, Christian Mayr, and Rene Schüffny</i>	
A Novel Hybrid Spiking Neuron: Response Analysis and Learning Potential	145
<i>Sho Hashimoto and Hiroyuki Torikai</i>	
Event-Related Desynchronisation/Synchronisation of Spontaneous Motor Actions	153
<i>Somnuk Phon-Amnuaisuk</i>	
Competition between Synapses Located in Proximal and Distal Dendrites of the Dentate Granule Cell through STDP	161
<i>Yukihiro Nonaka and Hatsuo Hayashi</i>	
An Analysis of the Autonomic Cardiac Activity by Reducing the Interplay between Sympathetic and Parasympathetic Information	169
<i>Fausto Lucena, D.S. Brito, Allan Kardec Barros, and Noboru Ohnishi</i>	
On Similarity Measures for Spike Trains	177
<i>Justin Dauwels, François Vialatte, Theophane Weber, and Andrzej Cichocki</i>	

Relationship between an Input Sequence and Asymmetric Connections Formed by Theta Phase Precession and STDP	186
<i>Naoyuki Sato and Yoko Yamaguchi</i>	
Analysis of Microelectrographic Neuronal Background in Deep Brain Nuclei in Parkinson Disease	194
<i>Hsiao-Lung Chan, Ming-An Lin, Tony Wu, Pei-Kuang Chao, Shih-Tseng Lee, and Peng-Chuan Chen</i>	
Gating Echo State Neural Networks for Time Series Forecasting	200
<i>Štefan Babinec and Jiří Pospíchal</i>	
A Novel Artificial Model of Spiral Ganglion Cell and Its Spike-Based Encoding Function	208
<i>Hiroyuki Torikai and Toru Nishigami</i>	
Evolution of Neural Organization in a Hydra-Like Animat	216
<i>Ben Jones, Yaochu Jin, Xin Yao, and Bernhard Sendhoff</i>	
Improved Sparse Bump Modeling for Electrophysiological Data	224
<i>François-Benoit Vialatte, Justin Dauwels, Jordi Solé-Casals, Monique Maurice, and Andrzej Cichocki</i>	
Classify Event-Related Motor Potentials of Cued Motor Actions	232
<i>Somnuk Phon-Amnuaisuk</i>	
A Neural Network Based Hierarchical Motor Schema of a Multi-finger Hand and Its Motion Diversity	240
<i>Eiichi Inohira, Shiori Uota, and Hirokazu Yokoi</i>	

III Cognitive Neuroscience

Biological Plausibility of Spectral Domain Approach for Spatiotemporal Visual Saliency	251
<i>Peng Bian and Liming Zhang</i>	
A “Global Closure” Effect in Contour Integration	259
<i>Kazuhiko Sakamoto, Hidekazu Nakajima, Takeshi Suzuki, and Masafumi Yano</i>	
Modeling of Associative Dynamics in Hippocampal Contributions to Heuristic Decision Making	267
<i>Miki Hirabayashi and Hirodata Ohashi</i>	
Tracking with Depth-from-Size	275
<i>Chen Zhang, Volker Willert, and Julian Eggert</i>	
Training Recurrent Connectionist Models on Symbolic Time Series	285
<i>Michal Černánský and Ľubica Beňušková</i>	

Computational Modeling of Risk–Related Eye Movement of Car Drivers	293
<i>Masayoshi Sato, Yuki Togashi, Takashi Omori, Koichiro Yamauchi, Satoru Ishikawa, and Toshihiro Wakita</i>	
Robust Detection of Medial-Axis by Onset Synchronization of Border-Ownership Selective Cells and Shape Reconstruction from Its Medial-Axis	301
<i>Yasuhiro Hatori and Ko Sakai</i>	
Synaptic Cooperation and Competition in STDP Learning Rule	310
<i>Shigeru Kubota and Tatsuo Kitajima</i>	
An Exemplar-Based Statistical Model for the Dynamics of Neural Synchrony	318
<i>Justin Dauwels, François Vialatte, Theophane Weber, and Andrzej Cichocki</i>	
Towards a Comparative Theory of the Primates’ Tool-Use Behavior	327
<i>Toshisada Mariyama and Hideaki Itoh</i>	
Artifact Removal Using Simultaneous Current Estimation of Noise and Cortical Sources	336
<i>Ken-ichi Morishige, Dai Kawawaki, Taku Yoshioka, Masa-aki Sato, and Mitsuo Kawato</i>	
Significance for Hippocampal Memory of Context-Like Information Generated in Hippocampal CA3c	344
<i>Toshikazu Samura, Motonobu Hattori, Shinichi Kikuchi, and Shun Ishizaki</i>	
Bio-signal Integration for Humanoid Operation: Gesture and Brain Signal Recognition by HMM/SVM-Embedded BN	352
<i>Yasuo Matsuyama, Fumiya Matsushima, Youichi Nishida, Takashi Hatakeyama, Koji Sawada, and Takatoshi Kato</i>	
Interpreting Dopamine Activities in Stochastic Reward Tasks	361
<i>Akiyo Asahina, Jun-ichiro Hirayama, and Shin Ishii</i>	
Epileptogenic ECoG Monitoring and Brain Stimulation Using a Multifunctional Microprobe for Minimally Invasive Brain Microsurgery	369
<i>Toshitaka Yamakawa, Takeshi Yamakawa, Michiyasu Suzuki, and Masami Fujii</i>	
Generalization in Linguistic Category Formation Based on the Prototype Effect	377
<i>Tamami Sudo and Ken Mogi</i>	

Model of the Activity of Hippocampal Neurons Based on the Theory of Selective Desensitization	384
<i>Atsuo Suemitsu, Yasuhiro Miyazawa, and Masahiko Morita</i>	
EEG-Based Classification of Brain Activity for Brightness Stimuli	392
<i>Qi Zhang</i>	
Steady State Visual Evoked Potentials in the Delta Range (0.5-5 Hz) ...	400
<i>François-Benoit Vialatte, Monique Maurice, Justin Dauwels, and Andrzej Cichocki</i>	
Using Optimality to Predict Photoreceptor Distribution in the Retina	408
<i>Travis Monk and Chris Harris</i>	
Optical Imaging of Plastic Changes Induced by Fear Conditioning in the Auditory Cortex of Guinea Pig	416
<i>Yoshinori Ide, Johan Lauwereyns, and Minoru Tsukada</i>	
Possibility of Cantor Coding by Spatial Input Patterns	423
<i>Yasuhiro Fukushima, Minoru Tsukada, Ichiro Tsuda, Yutaka Yamaguti, and Shigeru Kuroda</i>	
A Neural Network Model for a Hierarchical Spatio-temporal Memory ...	428
<i>Kiruthika Ramanathan, Luping Shi, Jianming Li, Kian Guan Lim, Ming Hui Li, Zhi Ping Ang, and Tow Chong Chong</i>	
Time-Varying Synchronization of Visual ERP during Sentences Identification	436
<i>Minfen Shen, Jialiang Chen, and K.H. Ting</i>	
Neural Mechanism of Synchronous Firing of Inferior Temporal Cortex in Face Perception	444
<i>Kazuhiro Takazawa and Yoshiki Kashimori</i>	

IV Bioinformatics

Clustering of Spectral Patterns Based on EMD Components of EEG Channels with Applications to Neurophysiological Signals Separation ...	453
<i>Tomasz M. Rutkowski, Andrzej Cichocki, Toshihisa Tanaka, Anca L. Ralescu, and Danilo P. Mandic</i>	
Consensus Clustering Using Spectral Theory	461
<i>Mariá C.V. Nascimento, Franklin M.B. de Toledo, and André C.P.L.F. de Carvalho</i>	

On the Synchrony of Morphological and Molecular Signaling Events in Cell Migration	469
<i>Justin Dauwels, Yuki Tsukada, Yuichi Sakumura, Shin Ishii, Kazuhiro Aoki, Takeshi Nakamura, Michiyuki Matsuda, François Vialatte, and Andrzej Cichocki</i>	
MISCORE: Mismatch-Based Matrix Similarity Scores for DNA Motif Detection	478
<i>Dianhui Wang and Nung Kion Lee</i>	
Ensembles of Pre-processing Techniques for Noise Detection in Gene Expression Data	486
<i>Giampaolo L. Libralon, André C.P.L.F. Carvalho, and Ana C. Lorena</i>	
FES Position Control of Forearm Using EOG	494
<i>Ken Suetsugu, Yoshihiko Tagawa, Tomohisa Inada, and Naoto Shiba</i>	
Reduction of FPs for Lung Nodules in MDCT by Use of Temporal Subtraction with Voxel-Matching Technique	504
<i>Yoshinori Itai, Hyoungseop Kim, Seiji Ishikawa, Shigehiko Katsuragawa, and Kunio Doi</i>	
Improved Mass Spectrometry Peak Intensity Prediction by Adaptive Feature Weighting	513
<i>Alexandra Scherbart, Wiebke Timm, Sebastian Böcker, and Tim W. Nattkemper</i>	
An Improved Genetic Algorithm for DNA Motif Discovery with Public Domain Information	521
<i>Xi Li and Dianhui Wang</i>	
A Hybrid Model for Prediction of Peptide Binding to MHC Molecules	529
<i>Ping Zhang, Vladimir Brusic, and Kaye Basford</i>	

V Special Session: Data Mining Methods for Cybersecurity

An Evaluation of Machine Learning-Based Methods for Detection of Phishing Sites	539
<i>Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi</i>	
Detecting Methods of Virus Email Based on Mail Header and Encoding Anomaly	547
<i>Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi</i>	

Faster Parameter Detection of Polymorphic Viral Code Using Hot List Strategy	555
Ruo Ando	
G-Means: A Clustering Algorithm for Intrusion Detection	563
<i>Zhonghua Zhao, Shanqing Guo, Qiuliang Xu, and Tao Ban</i>	
Anomaly Intrusion Detection for Evolving Data Stream Based on Semi-supervised Learning	571
<i>Yan Yu, Shanqing Guo, Shaohua Lan, and Tao Ban</i>	
An Incident Analysis System NICTER and Its Analysis Engines Based on Data Mining Techniques	579
<i>Daisuke Inoue, Katsunari Yoshioka, Masashi Eto, Masaya Yamagata, Eisuke Nishino, Jun'ichi Takeuchi, Kazuya Ohkouchi, and Koji Nakao</i>	
Multi-Layered Hand and Face Tracking for Real-Time Gesture Recognition	587
<i>Farhad Dadgostar, Abdolhossein Sarrafzadeh, and Chris Messom</i>	
Towards a Reliable Evaluation Framework for Message Authentication in Web-Based Transactions Based on an Improved Computational Intelligence and Dynamical Systems Methodology	595
<i>Dimitrios Alexios Karras and Vasilios C. Zorkadis</i>	

VI Special Session: Computational Models and Their Applications in Machine Learning and Pattern Recognition

A Neuro-GA Approach for the Maximum Fuzzy Clique Problem	605
<i>Sanghamitra Bandyopadhyay and Malay Bhattacharyya</i>	
Hybrid Feature Selection: Combining Fisher Criterion and Mutual Information for Efficient Feature Selection.....	613
<i>Chandra Shekhar Dhir and Soo Young Lee</i>	
Sensibility-Aware Image Retrieval Using Computationally Learned Bases: RIM, JPG, J2K, and Their Mixtures	621
<i>Takatoshi Kato, Shun'ichi Honma, Yasuo Matsuyama, Tetsuma Yoshino, and Yuuki Hoshino</i>	
An Analysis of Generalization Error in Relevant Subtask Learning	629
<i>Keisuke Yamazaki and Samuel Kaski</i>	
Intelligent Automated Guided Vehicle with Reverse Strategy: A Comparison Study	638
<i>Shigeru Kato and Kok Wai Wong</i>	

Neural Networks for Optimal Form Design of Personal Digital Assistants	647
<i>Chen-Cheng Wang, Yang-Cheng Lin, and Chung-Hsing Yeh</i>	
Firing Rate Estimation Using an Approximate Bayesian Method	655
<i>Kazuho Watanabe and Masato Okada</i>	
Sampling Curve Images to Find Similarities among Parts of Images	663
<i>Kazunori Iwata and Akira Hayashi</i>	
Improving the State Space Organization of Untrained Recurrent Networks	671
<i>Michal Černánský, Matej Makula, and Ľubica Beňušková</i>	
Online Multibody Factorization Based on Bayesian Principal Component Analysis of Gaussian Mixture Models	679
<i>Kentarou Hitomi, Takashi Bando, Naoki Fukaya, Kazushi Ikeda, and Tomohiro Shibata</i>	
Experimental Study of Ergodic Learning Curve in Hidden Markov Models	688
<i>Masashi Matsumoto and Sumio Watanabe</i>	
Design of Exchange Monte Carlo Method for Bayesian Learning in Normal Mixture Models	696
<i>Kenji Nagata and Sumio Watanabe</i>	
Image Filling-In: A Gestalt Approach	707
<i>Jun Ma</i>	
Sports Video Segmentation Using a Hierarchical Hidden CRF	715
<i>Hirotaka Tamada and Akira Hayashi</i>	
Learning Manifolds for Bankruptcy Analysis	723
<i>Bernardete Ribeiro, Armando Vieira, João Duarte, Catarina Silva, João Carvalho das Neves, Qingzhong Liu, and Andrew H. Sung</i>	
Information Geometry of Interspike Intervals in Spiking Neurons with Refractories	731
<i>Daisuke Komazawa, Kazushi Ikeda, and Hiroyuki Funaya</i>	
Convulsive Blind Speech Separation by Decorrelation	737
<i>Fuxiang Wang and Jun Zhang</i>	

VII Special Session: Recent Advances in Brain-Inspired Technologies for Robotics

Cognitive Representation and Bayesian Model of Spatial Object Contexts for Robot Localization	747
<i>Chuho Yi, Il Hong Suh, Gi Hyun Lim, Seungdo Jeong, and Byung-Uk Choi</i>	
Learning of Action Generation from Raw Camera Images in a Real-World-Like Environment by Simple Coupling of Reinforcement Learning and a Neural Network	755
<i>Katsunari Shibata and Tomohiko Kawano</i>	
Brain-Inspired Emergence of Behaviors Based on the Desire for Existence by Reinforcement Learning	763
<i>Mikio Morita and Masumi Ishikawa</i>	
A Neural Network Based Controller for an Outdoor Mobile Robot	771
<i>Masanori Sato, Atsushi Kanda, and Kazuo Ishii</i>	
Depth Perception Using a Monocular Vision System	779
<i>Xuebing Wang and Kazuo Ishii</i>	
Trajectory Planning with Dynamics Constraints for an Underactuated Manipulator	787
<i>Yuya Nishida and Masahiro Nagamatu</i>	
Neural Networks That Mimic the Human Brain: Turing Machines versus Machines That Generate Conscious Sensations	794
<i>Alan Rosen and David B. Rosen</i>	

VIII Special Session: Lifelong Incremental Learning for Intelligent Systems

A Vector Quantization Approach for Life-Long Learning of Categories	805
<i>Stephan Kirstein, Heiko Wersing, Horst-Michael Gross, and Edgar Körner</i>	
An Integrated System for Incremental Learning of Multiple Visual Categories	813
<i>Stephan Kirstein, Heiko Wersing, Horst-Michael Gross, and Edgar Körner</i>	
A Neural Network Model for Sequential Multitask Pattern Recognition Problems	821
<i>Hitoshi Nishikawa, Seiichi Ozawa, and Asim Roy</i>	

Automatic Discovery of Subgoals in Reinforcement Learning Using Strongly Connected Components	829
<i>Seyed Jalal Kazemitabar and Hamid Beigy</i>	

IX Special Session: Dynamics of Neural Networks

Bifurcation and Windows in a Simple Piecewise Linear Chaotic Spiking Neuron	837
<i>Tomonari Hasegawa and Toshimichi Saito</i>	
Bifurcation between Superstable Periodic Orbits and Chaos in a Simple Spiking Circuit	844
<i>Yuji Kawai and Toshimichi Saito</i>	
Application of Higher Order Neural Network Dynamics to Distributed Radio Resource Usage Optimization of Cognitive Wireless Networks	851
<i>Mikio Hasegawa, Taichi Takeda, Taro Kuroda, Ha Nguyen Tran, Goh Miyamoto, Yoshitoshi Murata, Hiroshi Harada, and Shuzo Kato</i>	
Synchronized Rhythmic Signals Effectively Influence Ongoing Cortical Activity for Decision-Making: A Study of the Biological Plausible Neural Network Model.....	859
<i>Hiroaki Wagatsuma and Yoko Yamaguchi</i>	
Synchronization Transition in a Pair of Coupled Non-identical Oscillators	867
<i>Yasuomi D. Sato, Yuji Tanaka, and Masatoshi Shiino</i>	
Parameter Analysis for Removing the Local Minima of Combinatorial Optimization Problems by Using the Inverse Function Delayed Neural Network	875
<i>Yoshihiro Hayakawa and Koji Nakajima</i>	
Fractional-Order Hopfield Neural Networks.....	883
<i>Arefeh Boroomand and Mohammad B. Menhaj</i>	

X Special Session: Applications of Intelligent Methods in Ecological Informatics

Classification and Prediction of Lower Troposphere Layers Influence on RF Propagation Using Artificial Neural Networks	893
<i>Martin Mudroch, Pavel Pecháč, Martin Grábner, and Václav Kvičera</i>	
Predicting the Distribution of Fungal Crop Diseases from Abiotic and Biotic Factors Using Multi-Layer Perceptrons.....	901
<i>Michael J. Watts and Sue P. Worner</i>	

Using Time Lagged Input Data to Improve Prediction of Stinging Jellyfish Occurrence at New Zealand Beaches by Multi-Layer Perceptrons	909
<i>David R. Pontin, Sue P. Worner, and Michael J. Watts</i>	
Modelling Climate Change Effects on Wine Quality Based on Expert Opinions Expressed in Free-Text Format: The WEBSOM Approach	917
<i>Subana Shanmuganathan and Philip Sallis</i>	

XI Special Session: Pattern Recognition from Real-World Information by SVM and Other Sophisticated Techniques

A Support Vector Machine with Forgetting Factor and Its Statistical Properties	929
<i>Hiroyuki Funaya, Yoshihiko Nomura, and Kazushi Ikeda</i>	
Improved Parameter Tuning Algorithms for Fuzzy Classifiers	937
<i>Kazuya Morikawa and Shigeo Abe</i>	
Accelerated Classifier Training Using the PSL Cascading Structure	945
<i>Teo Susnjak and Andre L.C. Barczak</i>	
Imitation Learning from Unsegmented Human Motion Using Switching Autoregressive Model and Singular Vector Decomposition	953
<i>Tadahiro Taniguchi and Naoto Iwahashi</i>	
Vision Based Mobile Robot for Indoor Environmental Security	962
<i>Sean W. Gordon, Shaoning Pang, Ryota Nishioka, Nikola Kasabov, and Takeshi Yamakawa</i>	
Multiobjective Multiclass Soft-Margin Support Vector Machine Maximizing Pair-Wise Interclass Margins	970
<i>Keiji Tatsumi, Ryo Kawachi, Kenji Hayashida, and Tetsuzo Tanino</i>	
Functional Networks Based on Pairwise Spike Synchrony Can Capture Topologies of Synaptic Connectivity in a Local Cortical Network Model	978
<i>Katsunori Kitano and Kazuhiro Yamada</i>	
Prediction of the O-glycosylation by Support Vector Machines and Semi-supervised Learning	986
<i>Hirotaka Sakamoto, Yukiko Nakajima, Kazutoshi Sakakibara, Masahiro Ito, and Ikuko Nishikawa</i>	

Practical Approach to Outlier Detection Using Support Vector Regression	995
<i>Junya Nishiguchi, Chosei Kaseda, Hirotaka Nakayama, Masao Arakawa, and Yeboon Yun</i>	
A Variant of Adaptive Mean Shift-Based Clustering	1002
<i>Fajie Li and Reinhard Klette</i>	

XII Special Session: Neural Information Processing in Cooperative Multi-robot Systems

Using Spiking Neural Networks for the Generation of Coordinated Action Sequences in Robots	1013
<i>Pilar Caamaño, Jose Antonio Becerra, Francisco Bellas, and Richard J. Duro</i>	
Neuro-Evolutive System for Ego-Motion Estimation with a 3D Camera	1021
<i>Ivan Villaverde, Zelmar Echegoyen, and Manuel Graña</i>	
Neuro Granular Networks with Self-learning Stochastic Connections: Fusion of Neuro Granular Networks and Learning Automata Theory	1029
<i>Darío Maravall and Javier de Lope</i>	
An Incremental Learning Algorithm for Optimizing High-Dimensional ANN-Based Classification Systems	1037
<i>Abraham Prieto, Francisco Bellas, Richard J. Duro, and Fernando Lopez-Peña</i>	
Towards the Adaptive Control of a Multirobot System for an Elastic Hose	1045
<i>Zelmar Echegoyen, Alicia d'Anjou, Ivan Villaverde, and Manuel Graña</i>	
Economical Implementation of Control Loops for Multi-robot Systems	1053
<i>Jose Manuel Lopez-Guede, Manuel Graña, Ekaitz Zulueta, and Oscar Barambones</i>	

XIII WORKSHOP: Hybrid and Adaptive Systems for Real-Time Robotics Vision and Control

An Improved Modular Neural Network Model for Adaptive Trajectory Tracking Control of Robot Manipulators	1063
<i>Dimitrios Alexios Karras</i>	

Variable Colour Depth Look-Up Table Based on Fuzzy Colour Processing	1071
<i>Heesang Shin and Napoleon H. Reyes</i>	
Towards a Generalised Hybrid Path-Planning and Motion Control System with Auto-calibration for Animated Characters in 3D Environments	1079
<i>Antony P. Gerdelen and Napoleon H. Reyes</i>	
Cultivated Microorganisms Control a Real Robot: A Model of Dynamical Coupling between Internal Growth and Robot Movement ...	1087
<i>Hiroaki Wagatsuma</i>	
Stream Processing of Geometric and Central Moments Using High Precision Summed Area Tables	1095
<i>Chris Messom and Andre Barczak</i>	
Bayesian Fusion of Auditory and Visual Spatial Cues during Fixation and Saccade in Humanoid Robot.....	1103
<i>Wei Kin Wong, Tze Ming Neoh, Chu Kiong Loo, and Chuan Poh Ong</i>	
Solving the Online SLAM Problem with an Omnidirectional Vision System	1110
<i>Vitor Campanholo Guizilini and Jun Okamoto Jr.</i>	

XIV WORKSHOP: Neurocomputing and Evolving Intelligence – NCEI 2008

A Notable Swarm Approach to Evolve Neural Network for Classification in Data Mining	1121
<i>Satchidananda Dehuri, Bijan Bihari Mishra, and Sung-Bae Cho</i>	
FPGA Implementation of an Evolving Spiking Neural Network	1129
<i>Alan Zuppich and Snjezana Soltic</i>	
HyFIS-Yager-gDIC: A Self-organizing Hybrid Neural Fuzzy Inference System Realizing Yager Inference	1137
<i>Sau Wai Tung, Chai Quek, and Cuntai Guan</i>	
Parallel Ant Colony Optimizer Based on Adaptive Resonance Theory Maps	1146
<i>Hiroshi Koshimizu and Toshimichi Saito</i>	
Covariate Shift and Incremental Learning	1154
<i>Koichiro Yamauchi</i>	

A Novel Incremental Linear Discriminant Analysis for Multitask Pattern Recognition Problems	1163
<i>Masayuki Hisada, Seiichi Ozawa, Kau Zhang, Shaoning Pang, and Nikola Kasabov</i>	
Soft Sensor Based on Adaptive Local Learning.....	1172
<i>Petr Kadlec and Bogdan Gabrys</i>	
Directly Optimizing Topology-Preserving Maps with Evolutionary Algorithms	1180
<i>José Everardo B. Maia, André L.V. Coelho, and Guilherme A. Barreto</i>	
RBF NN Based Adaptive PI Control of Brushless DC Motor	1188
<i>Jie Xiu, Yan Xiu, and Shiyu Wang</i>	
Incremental Principal Component Analysis Based on Adaptive Accumulation Ratio	1196
<i>Seiichi Ozawa, Kazuya Matsumoto, Shaoning Pang, and Nikola Kasabov</i>	
Ontology Based Personalized Modeling for Chronic Disease Risk Analysis: An Integrated Approach.....	1204
<i>Anju Verma, Nikola Kasabov, Elaine Rush, and Qun Song</i>	
Frost Prediction Characteristics and Classification Using Computational Neural Networks	1211
<i>Philip Sallis, Mary Jarur and Marcelo Trujillo</i>	
Personalized Modeling Based Gene Selection for Microarray Data Analysis	1221
<i>Yingjie Hu, Qun Song, and Nikola Kasabov</i>	
Integrated Feature and Parameter Optimization for an Evolving Spiking Neural Network	1229
<i>Stefan Schliebs, Michaël Defoين-Plateل, and Nikola Kasabov</i>	
Personalised Modelling for Multiple Time-Series Data Prediction: A Preliminary Investigation in Asia Pacific Stock Market Indexes Movement	1237
<i>Harya Widiputra, Russel Pears, and Nikola Kasabov</i>	
Dynamic Neural Fuzzy Inference System	1245
<i>Yuan-Chun Hwang and Qun Song</i>	
Author Index	1251

Table of Contents – Part II

I Neural Network Based Semantic Web, Data Mining and Knowledge Discovery

A Novel Method for Manifold Construction	3
<i>Wei-Chen Cheng and Cheng-Yuan Liou</i>	
A Non-linear Classifier for Symbolic Interval Data Based on a Region Oriented Approach	11
<i>Renata M.C.R. de Souza and Diogo R.S. Salazar</i>	
A Symmetrical Model Applied to Interval-Valued Data Containing Outliers with Heavy-Tail Distribution	19
<i>Marco A.O. Domingues, Renata M.C.R. de Souza, and Francisco José A. Cysneiros</i>	
New Neuron Model for Blind Source Separation.....	27
<i>Md. Shiblee, B. Chandra, and P.K. Kalra</i>	
Time Series Prediction with Multilayer Perceptron (MLP): A New Generalized Error Based Approach	37
<i>Md. Shiblee, P.K. Kalra, and B. Chandra</i>	
Local Feature Selection in Text Clustering	45
<i>Marcelo N. Ribeiro, Manoel J.R. Neto, and Ricardo B.C. Prudêncio</i>	
Sprinkled Latent Semantic Indexing for Text Classification with Background Knowledge	53
<i>Haiqin Yang and Irwin King</i>	
Comparison of Cluster Algorithms for the Analysis of Text Data Using Kolmogorov Complexity	61
<i>Tina Geweniger, Frank-Michael Schleif, Alexander Hasenfuss, Barbara Hammer, and Thomas Villmann</i>	
Neurocognitive Approach to Clustering of PubMed Query Results	70
<i>Paweł Matykiewicz, Włodzisław Duch, Paul M. Zender, Keith A. Crutcher, and John P. Pestian</i>	
Search-In-Synchrony: Personalizing Web Search with Cognitive User Profile Model	80
<i>Chandra Shekhar Dhir and Soo Young Lee</i>	
Neurocognitive Approach to Creativity in the Domain of Word-Invention	88
<i>Maciej Pilichowski and Włodzisław Duch</i>	

Improving Personal Credit Scoring with HLVQ-C	97
<i>Armando Vieira, João Duarte, Bernardete Ribeiro, and Joao Carvalho Neves</i>	
Architecture of Behavior-Based Function Approximator for Adaptive Control	104
<i>Hassab Elgawi Osman</i>	
On Efficient Content Based Information Retrieval Using SVM and Higher Order Correlation Analysis	112
<i>Dimitrios Alexios Karras</i>	

II Neural Networks Learning Paradigm

A String Measure with Symbols Generation: String Self-Organizing Maps	123
<i>Luis Fernando de Mingo López, Nuria Gómez Blas, and Miguel Angel Díaz</i>	
Neural Network Smoothing of Geonavigation Data on the Basis of Multilevel Regularization Algorithm	131
<i>Vladimir Vasilyev and Ildar Nugaev</i>	
Knowledge-Based Rule Extraction from Self-Organizing Maps	139
<i>Chihli Hung</i>	
A Bayesian Local Linear Wavelet Neural Network	147
<i>Kunikazu Kobayashi, Masanao Obayashi, and Takashi Kuremoto</i>	
Analysis on Equilibrium Point of Expectation Propagation Using Information Geometry	155
<i>Hideyuki Matsui and Toshiyuki Tanaka</i>	
Partially Enhanced Competitive Learning	163
<i>Ryotaro Kamimura</i>	
Feature Detection by Structural Enhanced Information	171
<i>Ryotaro Kamimura</i>	
Gradient Learning in Networks of Smoothly Spiking Neurons	179
<i>Jiří Šíma</i>	
Orthogonalization and Thresholding Method for a Nonparametric Regression Problem	187
<i>Katsuyuki Hagiwara</i>	
Analysis of Ising Spin Neural Network with Time-Dependent Mexican-Hat-Type Interaction	195
<i>Kazuyuki Hara, Seiji Miyoshi, Tatsuya Uezu, and Masato Okada</i>	

Divided Chaotic Associative Memory for Successive Learning	203
<i>Takahiro Hada and Yuko Osana</i>	
Reinforcement Learning Using Kohonen Feature Map Associative Memory with Refractoriness Based on Area Representation	212
<i>Atsushi Shimizu and Yuko Osana</i>	
Automatic Model Selection via Corrected Error Backpropagation	220
<i>Masashi Sekino and Katsumi Nitta</i>	
Self-Referential Event Lists for Self-Organizing Modular Reinforcement Learning	228
<i>Johane Takeuchi, Osamu Shouno, and Hiroshi Tsujino</i>	
Generalisation Performance vs. Architecture Variations in Constructive Cascade Networks	236
<i>Suisin Khoo and Tom Gedeon</i>	
Synchronized Oriented Mutations Algorithm for Training Neural Controllers	244
<i>Vincent Berenz and Kenji Suzuki</i>	
Bioinspired Parameter Tuning of MLP Networks for Gene Expression Analysis: Quality of Fitness Estimates vs. Number of Solutions Analysed	252
<i>André L.D. Rossi, Carlos Soares, and André C.P.L.F. Carvalho</i>	
Sample Filtering Relief Algorithm: Robust Algorithm for Feature Selection	260
<i>Thammakorn Saethang, Santitham Prom-on, Asawin Meechai, and Jonathan Hoyin Chan</i>	
Enhanced Visualization by Combing SOM and Mixture Models	268
<i>Ryotaro Kamimura</i>	
Genetic Versus Nearest-Neighbor Imputation of Missing Attribute Values for RBF Networks	276
<i>Pedro G. de Oliveira and André L.V. Coelho</i>	
Combination of Dynamic Reservoir and Feedforward Neural Network for Time Series Forecasting	284
<i>Štefan Babinec and Jiří Pospíchal</i>	
Learning Nonadjacent Dependencies with a Recurrent Neural Network	292
<i>Igor Farkaš</i>	

A Back-Propagation Training Method for Multilayer Pulsed Neural Networks Using Principle of Duality	300
<i>Kaname Iwasa, Mauricio Kugler, Susumu Kuroyanagi, and Akira Iwata</i>	
Revisiting the Problem of Weight Initialization for Multi-Layer Perceptrons Trained with Back Propagation	308
<i>Stavros Adam, Dimitrios Alexios Karras, and Michael N. Vrahatis</i>	
Analysis on Generalization Error of Faulty RBF Networks with Weight Decay Regularizer	316
<i>Chi Sing Leung, Pui Fai Sum, and Hongjiang Wang</i>	
On Node-Fault-Injection Training of an RBF Network	324
<i>John Sum, Chi-sing Leung, and Kevin Ho</i>	

III Kernel Methods and SVM

Symbolic Knowledge Extraction from Support Vector Machines: A Geometric Approach	335
<i>Lu Ren and Artur d' Avila Garcez</i>	
Asbestos Detection from Microscope Images Using Support Vector Random Field of Local Color Features	344
<i>Yoshitaka Moriguchi, Kazuhiro Hotta, and Haruhisa Takahashi</i>	
Acoustic Echo Cancellation Using Gaussian Processes	353
<i>Jyun-ichiro Tomita and Yuzo Hirai</i>	
Automatic Particle Detection and Counting by One-Class SVM from Microscope Image	361
<i>Hinata Kuba, Kazuhiro Hotta, and Haruhisa Takahashi</i>	
Protein Folding Classification by Committee SVM Array	369
<i>Mika Takata and Yasuo Matsuyama</i>	
Implementation of the MLP Kernel	378
<i>Cheng-Yuan Liou and Wei-Chen Cheng</i>	
Fuzzy Rules Extraction from Support Vector Machines for Multi-class Classification with Feature Selection	386
<i>Adriana da Costa F. Chaves, Marley Vellasco, and Ricardo Tanscheit</i>	
An SVM Based Approach to Cross-Language Adaptation for Indian Languages	394
<i>A. Vijaya Rama Raju and C. Chandra Sekhar</i>	

Automatic Classification System for the Diagnosis of Alzheimer Disease Using Component-Based SVM Aggregations	402
<i>I. Álvarez, M. López, J.M. Górriz, J. Ramírez, D. Salas-Gonzalez, C.G. Puntonet, and F. Segovia</i>	
Early Detection of the Alzheimer Disease Combining Feature Selection and Kernel Machines	410
<i>J. Ramírez, J.M. Górriz, M. López, D. Salas-Gonzalez, I. Álvarez, F. Segovia, and C.G. Puntonet</i>	
Computer Aided Diagnosis of Alzheimer Disease Using Support Vector Machines and Classification Trees	418
<i>D. Salas-Gonzalez, J.M. Górriz, J. Ramírez, M. López, I. Álvarez, F. Segovia, and C.G. Puntonet</i>	
Modeling and Prediction of Nonlinear EEG Signal Using Local SVM Method	426
<i>Lisha Sun, Lanxin Lin, and Chunhao Lin</i>	

IV Neural Networks as a Soft Computing Technology

Suitability of Using Self-Organizing Neural Networks in Configuring P-System Communications Architectures.....	437
<i>Abraham Gutiérrez, Soledad Delgado, and Luis Fernández</i>	
Short Term Load Forecasting (STLF) Using Artificial Neural Network Based Multiple Lags of Time Series	445
<i>Mohd Hafez Hilmi Harun, Muhammad Murtadha Othman, and Ismail Musirin</i>	
Neural Network Regression for LHF Process Optimization	453
<i>Miroslaw Kordos</i>	
Trading Strategy in Foreign Exchange Market Using Reinforcement Learning Hierarchical Neuro-Fuzzy Systems	461
<i>Marcelo F. Corrêa, Marley Vellasco, Karla Figueiredo, and Pedro Vellasco</i>	
Improving Multi Step-Ahead Model Prediction through Backward Elimination Method in Multiple Neural Networks Combination	469
<i>Zainal Ahmad and Rabiatul Adawiah Mat Noor</i>	
A Novel Adaptive Resource-Aware PNN Algorithm Based on Michigan-Nested Pittsburgh PSO	477
<i>Kuncup Iswandy and Andreas König</i>	

Imputation of Missing Data Using PCA, Neuro-Fuzzy and Genetic Algorithms	485
<i>Nthabiseng Hlalele, Fulufhelo Nelwamondo, and Tshilidzi Marwala</i>	
Feature Selection Method with Multi-Population Agent Genetic Algorithm.....	493
<i>Yongming Li and Xiaoping Zeng</i>	
Particle Swarm Optimization and Differential Evolution in Fuzzy Clustering	501
<i>Fengqin Yang, Changhai Zhang, and Tieli Sun</i>	
Intelligent Control of Heating, Ventilating and Air Conditioning Systems	509
<i>Patrick Low Tiong Kie and Lau Bee Theng</i>	
Investigating Ensemble Weight and the Certainty Distributions for Indicating Structural Diversity.....	517
<i>Lesedi Melton Masisi, Fulufhelo Nelwamondo, and Tshilidzi Marwala</i>	

V Neural Networks and Pattern Recognition

Dynamic Programming Stereo on Real-World Sequences	527
<i>Zhifeng Liu and Reinhard Klette</i>	
Happy-Sad Expression Recognition Using Emotion Geometry Feature and Support Vector Machine	535
<i>Linlu Wang, Xiaodong Gu, Yuanyuan Wang, and Liming Zhang</i>	
A New Principal Axis Based Line Symmetry Measurement and Its Application to Clustering	543
<i>Sanghamitra Bandyopadhyay and Sriparna Saha</i>	
Class-Dependent Feature Selection for Face Recognition	551
<i>Zhou Nina and Lipo Wang</i>	
Partial Clustering for Tissue Segmentation in MRI	559
<i>Nicolau Gonçalves, Janne Nikkilä, and Ricardo Vigário</i>	
Time Series Analysis for Long Term Prediction of Human Movement Trajectories	567
<i>Sven Hellbach, Julian P. Eggert, Edgar Körner, and Horst-Michael Gross</i>	
Error Analysis of a Sub-millimeter Real-Time Target Recognition System with a Moving Camera	575
<i>V.M.M. Vieira, G.J. Kane, R. Marmulla, J. Raszkowsky, and G. Eggers</i>	

Automatic Plaque Boundary Extraction in Intravascular Ultrasound Image by Fuzzy Inference with Adaptively Allocated Membership Functions	583
<i>Eiji Uchino, Noriaki Suetake, Takanori Koga, Shohei Ichiyama, Genta Hashimoto, Takafumi Hiro, and Masunori Matsuzaki</i>	
Gabor Neural Network Based Facial Expression Recognition for Assistive Speech Expression	591
<i>Lau Bee Theng</i>	
Investigations into Particle Swarm Optimization for Multi-class Shape Recognition	599
<i>Ee Lee Ng, Mei Kuan Lim, Tomás Maul, and Weng Kin Lai</i>	
Patterns of Interactions in Complex Social Networks Based on Coloured Motifs Analysis	607
<i>Katarzyna Musial, Krzysztof Juszczyszyn, Bogdan Gabrys, and Przemysław Kazienko</i>	
Initialization Dependence of Clustering Algorithms	615
<i>Wim De Mulder, Stefan Schliebs, René Boel, and Martin Kuiper</i>	
Boundary Detection from Spectral Information	623
<i>Jun Ma</i>	
Improvement of Practical Recurrent Learning Method and Application to a Pattern Classification Task	631
<i>Mohamad Faizal bin Samsudin and Katsunari Shibata</i>	
An Automatic Intelligent Language Classifier	639
<i>Brijesh Verma, Hong Lee, and John Zakos</i>	
Gender Classification by Combining Facial and Hair Information	647
<i>Xiao-Chen Lian and Bao-Liang Lu</i>	
A Hybrid Fuzzy Approach for Human Eye Gaze Pattern Recognition ...	655
<i>Dingyun Zhu, B. Sumudu U. Mendis, Tom Gedeon, Akshay Asthana, and Roland Goecke</i>	
Interactive Trouble Condition Sign Discovery for Hydroelectric Power Plants	663
<i>Takashi Onoda, Norihiko Ito, and Hironobu Yamasaki</i>	
An Asbestos Counting Method from Microscope Images of Building Materials Using Summation Kernel of Color and Shape	671
<i>Atsuo Nomoto, Kazuhiro Hotta, and Haruhisa Takahashi</i>	
Evaluation of Prediction Capability of Non-recursion Type 2nd-order Volterra Neuron Network for Electrocardiogram	679
<i>Shunsuke Kobayakawa and Hirokazu Yokoi</i>	

A New ART-LMS Neural Network for the Image Restoration	687
<i>Tzu-Chao Lin, Mu-kun Liu, and Chien-Ting Yeh</i>	
Moving Vehicle Tracking Based on SIFT Active Particle Choosing	695
<i>Tao Gao, Zheng-guang Liu, Wen-chun Gao, and Jun Zhang</i>	
Classification of Fundus Images for Diagnosing Glaucoma by Self-Organizing Map and Learning Vector Quantization	703
<i>Nobuo Matsuda, Jorma Laaksonen, Fumiaki Tajima, and Hideaki Sato</i>	
Facial Expression Recognition Techniques Using Constructive Feedforward Neural Networks and K-Means Algorithm	711
<i>Liyi Ma</i>	
A Neural Network Based Classification of Human Blood Cells in a Multiphysic Framework	720
<i>Matteo Cacciola, Maurizio Fiasché, Giuseppe Megali, Francesco C. Morabito, and Mario Versaci</i>	
Caller Interaction Classification: A Comparison of Real and Binary Coded GA-MLP Techniques	728
<i>Pretesh B. Patel and Tshilidzi Marwala</i>	
A Robust Technique for Background Subtraction in Traffic Video	736
<i>Tao Gao, Zheng-guang Liu, Wen-chun Gao, and Jun Zhang</i>	
Gabor Filters as Feature Images for Covariance Matrix on Texture Classification Problem	745
<i>Jing Yi Tou, Yong Haur Tay, and Phooi Yee Lau</i>	
Investigating Demographic Influences for HIV Classification Using Bayesian Autoassociative Neural Networks	752
<i>Jaishheel Mistry, Fulufhelo V. Nelwamondo, and Tshilidzi Marwala</i>	
Hardware-Based Solutions Utilizing Random Forests for Object Recognition	760
<i>Hassab Elgawi Osman</i>	
A Neural Oscillation Model for Contour Separation in Color Images ...	768
<i>Yu Ma, Xiaodong Gu, and Yuanyuan Wang</i>	
A Color Image Segmentation Using Inhibitory Connected Pulse Coupled Neural Network	776
<i>Hiroaki Kurokawa, Shuzo Kaneko, and Masato Yonekawa</i>	
Generating Saliency Map Related to Motion Based on Self-organized Feature Extracting	784
<i>Satoru Morita</i>	

Intelligent Face Image Retrieval Using Eigenpexels and Learning Similarity Metrics	792
<i>Paul Conilione and Dianhui Wang</i>	
The Role of the Infant Vision System in 3D Object Recognition.....	800
<i>Roberto A. Vázquez, Humberto Sossa, and Beatriz A. Garro</i>	
Virtual Fence for a Surveillance System	808
<i>Yen San Yong, Hock Woon Hon, Yasir Salih Osman, Ching Hau Chan, Siu Jing Then, and Sheau Wei Chau</i>	
Application of mnSOM on Linking External Exposure to Internal Load	816
<i>Stefan W. Roeder, Matthias Richter, and Olf Herbarth</i>	

VI Neuromorphic Hardware and Embedded Neural Networks

Automated and Holistic Design of Intelligent and Distributed Integrated Sensor Systems with Self-x Properties for Applications in Vision, Robotics, Smart Environments, and Culinary Assistance Systems	827
<i>Andreas König</i>	
Hardware Design of Japanese Hand Sign Recognition System	835
<i>Hiroomi Hikawa and Hirotada Fujimura</i>	
Blind Source Separation System Using Stochastic Arithmetic on FPGA	843
<i>Michihiko Hori and Michihito Ueda</i>	
Noise-Tolerant Analog Circuits for Sensory Segmentation Based on Symmetric STDP Learning	851
<i>Gessyca Maria Tovar, Tetsuya Asai, and Yoshihito Amemiya</i>	
A Novel Approach for Hardware Based Sound Classification	859
<i>Mauricio Kugler, Victor Alberto Parcianello Benso, Susumu Kuroyanagi, and Akira Iwata</i>	
The Generalized Product Neuron Model in Complex Domain	867
<i>B.K. Tripathi, B. Chandra, and P.K. Kalra</i>	
Pulse-Type Hardware Neural Network with Two Time Windows in STDP	877
<i>Katsutoshi Saeki, Ryo Shimizu, and Yoshifumi Sekine</i>	
Time Evaluation for WTA Hopfield Type Circuits Affected by Cross-Coupling Capacitances	885
<i>Ruxandra L. Costea and Cornelius A. Marinov</i>	

Circuit FPGA for Active Rules Selection in a Transition P System Region	893
<i>Víctor Martínez, Abraham Gutiérrez, and Luis Fernando de Mingo</i>	

VII Machine Learning and Information Algebra

Model Selection Method for AdaBoost Using Formal Information Criteria	903
<i>Daisuke Kaji and Sumio Watanabe</i>	
The Diversity of Regression Ensembles Combining Bagging and Random Subspace Method	911
<i>Alexandra Scherbart and Tim W. Nattkemper</i>	
On Weight-Noise-Injection Training	919
<i>Kevin Ho, Chi-sing Leung, and John Sum</i>	
Intelligent Control of Heating, Ventilating and Air Conditioning Systems	927
<i>Patrick Low Tiong Kie and Lau Bee Theng</i>	
Bregman Divergences and Multi-dimensional Scaling	935
<i>Pei Ling Lai and Colin Fyfe</i>	
Collective Activations to Generate Self-Organizing Maps	943
<i>Ryotaro Kamimura</i>	
A Closed-Form Estimator of Fully Visible Boltzmann Machines	951
<i>Jun-ichiro Hirayama and Shin Ishii</i>	
Incremental Learning in the Non-negative Matrix Factorization	960
<i>Sven Rebhan, Waqas Sharif, and Julian Eggert</i>	
Contextual Behaviors and Internal Representations Acquired by Reinforcement Learning with a Recurrent Neural Network in a Continuous State and Action Space Task	970
<i>Hiroki Utsunomiya and Katsunari Shibata</i>	
Improving the Quality of EEG Data in Patients with Alzheimers Disease Using ICA	979
<i>François-Benoit Vialatte, Jordi Solé-Casals, Monique Maurice, Charles Latchoumane, Nigel Hudson, Sunil Wimalaratna, Jaeseung Jeong, and Andrzej Cichocki</i>	
Global Minimization of the Projective Nonnegative Matrix Factorization	987
<i>Zhijian Yuan</i>	

- Learning Sparse Representations Using a Parametric Cauchy Density 994
Ling-Zhi Liao

- A One-Layer Recurrent Neural Network for Non-smooth Convex Optimization Subject to Linear Equality Constraints 1003
Qingshan Liu and Jun Wang

VIII Brain-Computer Interface

- A Study on Application of Reliability Based Automatic Repeat Request to Brain Computer Interfaces 1013

Hiromu Takahashi, Tomohiro Yoshikawa, and Takeshi Furuhashi

- Analysis on Saccade-Related Independent Components by Various ICA Algorithms for Developing BCI 1021

Arao Funase, Motoaki Mouri, Yagi Tohru, Andrzej Cichocki, and Ichi Takumi

- Policy Gradient Learning of Cooperative Interaction with a Robot Using User's Biological Signals 1029

Tomoya Tamei and Tomohiro Shibata

- Real-Time Embedded EEG-Based Brain-Computer Interface 1038

Li-Wei Ko, I-Ling Tsai, Fu-Shu Yang, Jen-Feng Chung, Shao-Wei Lu, Tzyy-Ping Jung, and Chin-Teng Lin

IX Neural Network Implementations

- SpiNNaker: The Design Automation Problem 1049

Andrew Brown, David Lester, Luis Plana, Steve Furber, and Peter Wilson

- The Deferred Event Model for Hardware-Oriented Spiking Neural Networks 1057

Alexander Rast, Xin Jin, Mukaram Khan, and Steve Furber

- Particle Swarm Optimization with SIMD-Oriented Fast Mersenne Twister on the Cell Broadband Engine 1065

Jun Igarashi, Satoshi Sonoh, and Takanori Koga

- DNA Computing Hardware Design and Application to Multiclass Cancer Data 1072

Sun-Wook Choi and Chong Ho Lee

- Author Index** 1081

Part I

**INNS Symposium “New Directions in
Neural Networks”**

Integrative Probabilistic Evolving Spiking Neural Networks Utilising Quantum Inspired Evolutionary Algorithm: A Computational Framework

Nikola Kasabov

Knowledge Engineering and Discovery Research Institute, KEDRI

Auckland University of Technology, Auckland, New Zealand

nkasabov@aut.ac.nz

<http://www.kedri.info>

Abstract. Integrative evolving connectionist systems (iECOS) integrate principles from different levels of information processing in the brain, including cognitive-, neuronal-, genetic- and quantum, in their dynamic interaction over time. The paper introduces a new framework of iECOS called integrative probabilistic evolving spiking neural networks (ipSNN) that incorporate probability learning parameters. ipSNN utilize a quantum inspired evolutionary optimization algorithm to optimize the probability parameters as these algorithms belong to the class of estimation of distribution algorithms (EDA). Both spikes and input features in ipESNN are represented as quantum bits being in a superposition of two states (1 and 0) defined by a probability density function. This representation allows for the state of an entire ipESNN at any time to be represented probabilistically in a quantum bit register and probabilistically optimised until convergence using quantum gate operators and a fitness function. The proposed ipESNN is a promising framework for both engineering applications and brain data modeling as it offers faster and more efficient feature selection and model optimization in a large dimensional space in addition to revealing new knowledge that is not possible to obtain using other models. Further development of ipESNN are the neuro-genetic models – ipESNG, that are introduced too, along with open research questions.

1 Introduction: Integrative Evolving Connectionist Systems (iECOS)

Many successful artificial neural network (ANN) models have been developed and applied to date [3,9,10,19,21,26,30,32], the most recent ones being Spiking Neural Networks (SNN) [14,15,23-25,33-37]. SNN have a great potential for brain data analysis [1,4,5,7,45] and data modelling [8,38,40,42,44,46-48]. However, despite some past work [2,14,35,36,41], current SNN models cannot model *probabilistically* data that are large, complex, noisy and dynamically changing in a way that reflects the stochastic nature of many real-world problems and brain processes [4,16,28,38].

The brain is a dynamic information processing system that evolves its structure and functionality in time through information processing at different levels: cognitive-, ensemble of neurons-, single neuron-, molecular (genetic)-, quantum [26-29]. The

information processes at each level are very complex and difficult to understand as they evolve in time, but much more difficult to understand is the interaction between them and how this interaction affects learning and cognition in the brain. These information processes are manifested at different time scales, e.g. cognitive processes happen in seconds, neuronal – in milliseconds, molecular- in minutes, and quantum - in nano-seconds. They also happen in different dimensional spaces, but they “work” together in the brain and contribute together to its intelligence.

Recently new information about neuronal- [1,25], genetic- [5,31,45] and quantum [6,22,43] levels of information processes in the brain has been obtained. For example, whether a neuron spikes or does not spike at any given time could depend not only on input signals but also on other factors such as gene and protein expression levels or physical properties [22,31,45]. The paradigm of *Integrative Evolving Connectionist Systems (iECOS)* [27-29, 39], previously proposed by the author, considers the *integrated optimisation* of all these factors represented as parameters and features (input variables) of an ANN model. This approach will be used here to develop a principally new framework - integrative probabilistic evolving SNN (ipESNN).

2 Evolving Spiking Neural Network Models

2.1 SNN – General Principles

SNN represent information as trains of spikes, rather than as single scalars, thus allowing the use of such features as frequency, phase, incremental accumulation of input signals, time of activation, etc. [3,5,14,23,47]. Neuronal dynamics of a spiking neuron are based on the increase in the inner potential of a neuron (post synaptic potential, PSP), after every input spike arrival. When a PSP reaches a certain threshold, the neuron emits a spike at its output (Fig. 1).

A wide range of models to simulate spiking neuronal activity have been proposed (for a review, see [25]). The Hodgkin- Huxley model is based on experimental study of the influence of conductance of three ion channels on the spike activity of the axon. The spike activity is modelled by an electric circuit, where the *chloride channel* is modelled with a parallel resistor-capacitor circuit, and the *sodium* and *potassium* channels are represented by voltage-dependent resistors.

In another model - the spike response model (SRM), a neuron i receives input spikes from pre-synaptic neurons $j \in \Gamma_i$, where Γ_i is a pool of all neurons pre-synaptic to neuron i . The state of the neuron i is described by the state variable $u_i(t)$ that can be interpreted as a total postsynaptic potential (PSP) at the membrane of soma – fig.1. When $u_i(t)$ reaches a firing threshold $\vartheta_i(t)$, neuron i fires, i.e. emits a spike. The value of the state variable $u_i(t)$ is the sum of all postsynaptic potentials, i.e.

$$u_i(t) = \sum_{j \in \Gamma_i} \sum_{t_j \in F_j} J_{ij} \varepsilon_{ij} (t - t_j - \Delta_{ij}^{ax}) \quad (1)$$

where: the weight of synaptic connection from neuron j to neuron i is denoted by J_{ij} , which takes positive (negative) values for excitatory (inhibitory) connections, respectively; depending on the sign of J_{ij} , a pre-synaptic spike, generated at time t_j increases (or decreases) $u_i(t)$ by an amount of $\varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax})$, where Δ_{ij}^{ax} is an axonal

delay between neurons i and j which increases with Euclidean distance between neurons. The positive kernel $\varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax}) = \varepsilon_{ij}(s)$ expresses an individual postsynaptic potential (PSP) evoked by a pre-synaptic neuron j on neuron i and can be expressed by a double exponential formula (2):

$$\varepsilon_{ij}^{\text{synapse}}(s) = A^{\text{synapse}} \left(\exp\left(-\frac{s}{\tau_{\text{decay}}^{\text{synapse}}}\right) - \exp\left(-\frac{s}{\tau_{\text{rise}}^{\text{synapse}}}\right) \right) \quad (2)$$

where: $\tau_{\text{decay/rise}}^{\text{synapse}}$ are time constants of the rise and fall of an individual PSP; A is the PSP's amplitude; the parameter *synapse* represents the type of the activity of the synapse from the neuron j to neuron i , that can be measured and modeled separately for a *fast_excitation*, *fast_inhibition*, *slow_excitation*, and *slow_inhibition*.

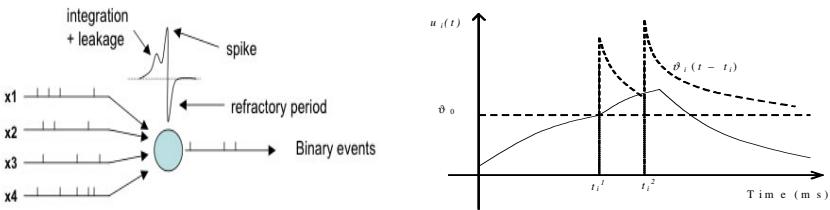


Fig. 1. A schematic representation of a spiking neuron model (from [5])

External inputs from the input layer of a SNN are added at each time step, thus incorporating the background noise and/or the background oscillations. Each external input has its own weight $J_{ik}^{\text{ext_input}}$ and amount of signal $\varepsilon_k(t)$, such that:

$$u_i^{\text{ext_input}}(t) = J_{ik}^{\text{ext_input}} \varepsilon_k(t) \quad (3)$$

2.2 Evolving Spiking Neural Networks (ESNN)

ESNN evolve/develop their structure and functionality in an incremental way from incoming data based on the following principles [26]:

- (i) New spiking neurons are created to accommodate new data, e.g. new patterns belonging to a class or new output classes, such as faces in a face recognition system;
- (ii) Spiking neurons are merged if they represent the same concept (class) and have similar connection weights (defined by a threshold of similarity).

In [40] an ESNN architecture is proposed where the change in a synaptic weight is achieved through a simple spike time dependent plasticity (STDP) learning rule:

$$\Delta w_{j,i} = \text{mod}^{\text{order}(j)} \quad (4)$$

where: $w_{j,i}$ is the weight between neuron j and neuron i , $\text{mod} \in (0,1)$ is the modulation factor, $\text{order}(j)$ is the order of arrival of a spike produced by neuron j to neuron i .

For each training sample, it is the *winner-takes-all* approach used, where only the neuron that has the highest PSP value has its weights updated. The postsynaptic

threshold (PSP_{Th}) of a neuron is calculated as a proportion $c \in [0, 1]$ of the maximum postsynaptic potential, $\max(PSP)$, generated with the propagation of the training sample into the updated weights, such that:

$$PSP_{Th} = c \max(PSP) \quad (5)$$

Creating and merging neurons based on localised incoming information and on system's performance are main operations of the ESNN architecture that make it continuously *evolvable*. Successful applications of ESNN for taste recognition, face recognition and multimodal audio-visual information processing, have been previously reported [40,46,47].

2.3 Computational Neurogenetic Models as iECOS

A further extension of the SRM, that takes into account the ion channel activity (and thus brings the benefits of both Hodgkin-Huxley model and the SRM), that is also based on neurobiology, is called computational neuro-genetic model (CNGM) as proposed in [5,26]. Here different synaptic activities that are influencing the spiking activity of a neuron are represented as functions of different proteins (neuro-transmitters, neuro-receptors and ion channels) that affect the PSP value and the PSP threshold. Some proteins and genes known to be affecting the spiking activity of a neuron such as fast_excitation, fast_inhibition, slow_excitation, and slow_inhibition (see formula (2)) are summarized in Table 1. Besides the genes coding for the proteins mentioned above and directly affecting the spiking dynamics of a neuron, a CNGM may include other genes relevant to a problem in hand, e.g. modeling a brain function or a brain disease, for example: c-jun, mGLuR3, Jerky, BDNF, FGF-2, IGF-I, GALR1, NOS, S100beta [5,45]). CNGM are iECOS as they integrate principles from neuronal and molecular level of information processing in the brain.

Table 1. Neuronal parameters and related proteins: PSP - postsynaptic potential, AMPAR - (amino- methylisoxazole- propionic acid) AMPA receptor, NMDAR - (n-methyl-d-aspartate acid) NMDA receptor, GABRA - (gamma-aminobutyric acid) GABA receptor, GABRB - GABAB receptor, SCN - sodium voltage-gated channel, KCN = kalium (potassium) voltage-gated channel, CLC = chloride channel (from [5])

Neuronal parameter	Protein
Fast excitation PSP	AMPAR
Slow excitation PSP	NMDAR
Fast inhibition PSP	GABRA
Slow inhibition PSP	GABRB
Firing threshold	SCN, KCN, CLC

However, it is also known that the spiking activity of the brain is stochastic [1,5,6,7]. And this is what is missing in the above SNN-, ESNN- and CNGM models that leave them not very suitable so far as large scale modeling techniques to model complex tasks. The problem is how to represent and process probabilities associated with spiking activity and how to build large ESNN probabilistic models.

3 Integrative Probabilistic Evolving SNN (ipESNN)

3.1 Biological Motivations

Some biological facts support the idea of ipESNN models [1,5,6,7]:

- For a neuron to spike or not to spike at a time t , is a “matter” of probability.
- Transmission of an electrical signal in a chemical synapse upon arrival of action potential into the terminal is probabilistic and depends on the probability of neurotransmitters to be released and ion channels to be open.
- Emission of a spike on the axon is also probabilistic.

The challenge is to develop a probabilistic neuronal model and to build ipESNN and ipESNG models for brain study and engineering applications. As the proposed below ipESNN model use quantum computation to deal with probabilities, we fist introduce some principles of quantum computing.

3.2 The Quantum Principle of *Superposition*

The smallest information unit in today's digital computers is one *bit*, existing as state ‘1’ or ‘0’ at any given time. The corresponding analogue in a quantum inspired representation is the quantum bit (*qbit*) [12,18,20]. Similar to classical bits a *qbit* may be in ‘1’ or ‘0’ states, but also in a *superposition* of both states. A qbit state $|\Psi\rangle$ can be described as:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (6)$$

where α and β are complex numbers that are used to define the probability of which of the corresponding states is likely to appear when a *qbit* is read (measured, collapsed). $|\alpha|^2$ and $|\beta|^2$ give the probability of a qbit being found in state ‘0’ or ‘1’ respectively. Normalization of the states to unity guarantees:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (7)$$

at any time. The *qbit* is not a single value entity, but is a function of parameters which values are complex numbers. In order to modify the probability amplitudes, *quantum gate operators* can be applied to the states of a *qbit* or a *qbit* vector. A quantum gate is represented by a square matrix, operating on the amplitudes $|\alpha|$ and $|\beta|$ in a Hilbert space, with the only condition that the operation is reversible. Such gates are: NOT-gate, rotation gate, Hadamard gate, and others [18, 20].

Another quantum principle is *entanglement* - two or more particles, regardless of their location, can be viewed as “correlated”, undistinguishable, “synchronized”, coherent. If one particle is “measured” and “collapsed”, it causes for all other entangled particles to “collapse” too.

The main motivations for the development of the ipSNN that utilize quantum computation are: (1) The biological facts about stochastic behavior of spiking neurons and SNN; (2) The properties of a quantum representation of probabilities; (3) The recent manifestation that quantum inspired evolutionary algorithms (QiEA) are probability estimation of distribution algorithms (EDA) [11].

3.3 QiEA Belong to the Class of EDA Algorithms

It was proven in [11] that QiEA, such as the vQEA, belong to the class of EDA. Using a quantum gate operator over consecutive iterations would lead to a change of the global state of a system described by a qbit chromosome(s) as shown in fig. 2. At the beginning (fig.2a) all states are equally probable and in the end – (fig2.c) the system converges to a local minimum according to a chosen fitness function.

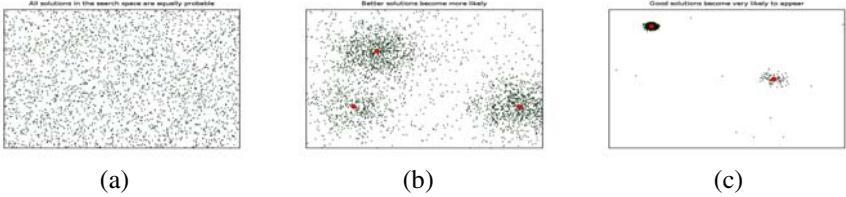


Fig. 2. An example of state convergence to local minima for a system described by a qbit register (chromosome) over 2 applications of a rotation quantum gate operator. The darker points represent states that have a higher probability of occurrence (the figure is produced by Stefan Schliebs).

3.4 The ipESNN Framework

The proposed here ipESNN framework is based on the following principles:

(a) *A quantum probabilistic representation of a spike:* A spike, at any time t , is both present (1) and not present (0), which is represented as a *qbit* defined by a probability density amplitude. When the spike is evaluated, it is either present or not present. To modify the probability amplitudes, a quantum gate operator is used, for example *the rotation gate*:

$$\begin{bmatrix} \alpha_i^j(t+1) \\ \beta_i^j(t+1) \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_i^j(t) \\ \beta_i^j(t) \end{bmatrix} \quad (8)$$

More precisely, a spike arriving at a moment t at each synapse Sij connecting a neuron Ni to a pre-synaptic neuron Nj , is represented as a qbit $Qij(t)$ with a probability to be in state “1” $\beta_{ij}(t)$ (probability for state “0” is $\alpha_{ij}(t)$). From the SNN architecture perspective this is equivalent to the existence (non-existence) of a connection Cij between neurons Nj and Ni .

(b) *A quantum probabilistic model of a spiking neuron for ipSNN:* A neuron Ni is represented as a qbit vector, representing all m synaptic connections to this neuron:

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array} \right] \quad (9)$$

At a time t each synaptic qbit represents the probability for a spike to arrive at the neuron. The post-synaptic inputs to the neuron are collapsed into spikes (or no spikes) and the cumulative input $u_i(t)$ to the neuron Ni is calculated as per. eq.(1).

Based on the above principles two architectures of a feed-forward ipESNN and a recurrent ipESNN are developed as illustrated in fig.3(a) and 3(b) respectively. All

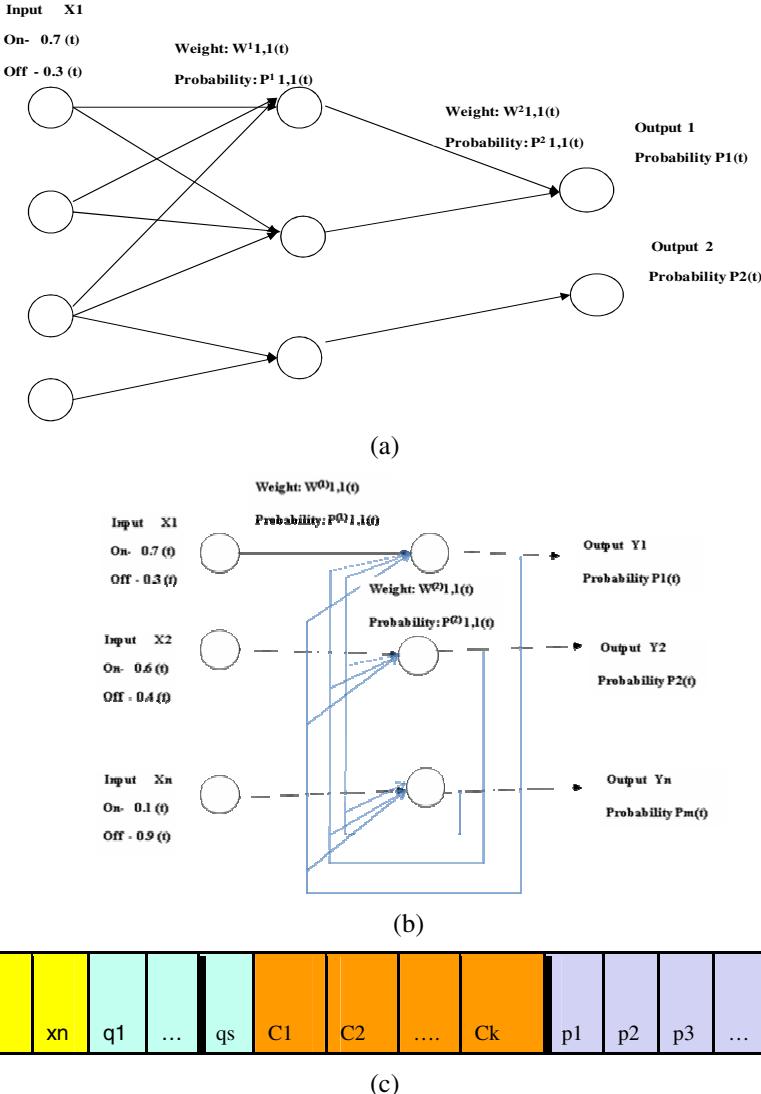


Fig. 3. (a) A schematic diagram of a feed-forward ipESNN; (b) and a recurrent ipESNN. (c) All input features (x_1, x_2, \dots, x_n), the ESNN parameters (q_1, q_2, \dots, q_s), the connections between neurons (C_1, C_2, \dots, C_k) and the probability of the neurons to spike (p_1, p_2, \dots, p_m) at time (t) are represented as an integrated qbit register that is operated as a whole [27,29].

input features (x_1, x_2, \dots, x_n), the ESNN parameters (q_1, q_2, \dots, q_s), the connections between the inputs and the neurons, including recurrent connections (C_1, C_2, \dots, C_k) and the probability of the neurons to spike (p_1, p_2, \dots, p_m) at time (t) are represented in an integrated *qbit* register that is operated upon as a whole [27,29] (fig.3c).

This framework goes beyond the traditional “wrapper” mode for feature selection and modelling [16]. The ipESNN single qbit vector (a chromosome) is optimised

through vQEA, therefore an ipESNN is a probabilistic EDA model. It was demonstrated that the vQEA is efficient for integrated feature and SNN parameter optimisation in a large dimensional space and also useful for extracting unique information from the modelled data [11,39]. All probability amplitudes together define a probability density ψ of the state of the ipESNN in a Hilbert space. This density will change if a quantum gate operator is applied according to an objective criterion (fitness function). This representation can be used for both tracing the learning process in an ipESNN system or the reaction of the system to an input vector.

(c) *ipSNN learning rules:* As the ipESNN model is an ESNN, in addition to the ESNN learning rules (formulas 4,5) there are rules to change the probability density amplitudes of spiking activity of a neuron. The probability $\beta_{ij}(t)$ of a spike to arrive from neuron N_j to neuron N_i (the connection between the two be present) will change according to STDP rule, which is implemented using the quantum rotation gate. In a more detailed model, $\beta_{ij}(t)$ will depend on the strength and the frequency of the spikes, on the distance D_{ij} between neurons N_j and N_i , and on many other physical and chemical parameters that are ignored in this model but can be added if necessary.

(d) *The principle of feature superposition representation* [27,29]: A vector of n qubits represents the probability of using each input variable x_1, x_2, \dots, x_n in the model at a time t . When the model computes, all features are “collapsed”, where “0” represents that a variable is not used, and “1” – the variable is used.

The above principles can also be used to develop more sophisticated ipESNN models, as the one presented below – the ipESNG model.

4 Integrative Probabilistic Evolving Spiking Neuro-Genetic Models (ipESNG)

A schematic diagram of an ipESNG model is given in fig.4a. The framework combines ipESNN and a gene regulatory network (GRN) similar to the CNGM [5]

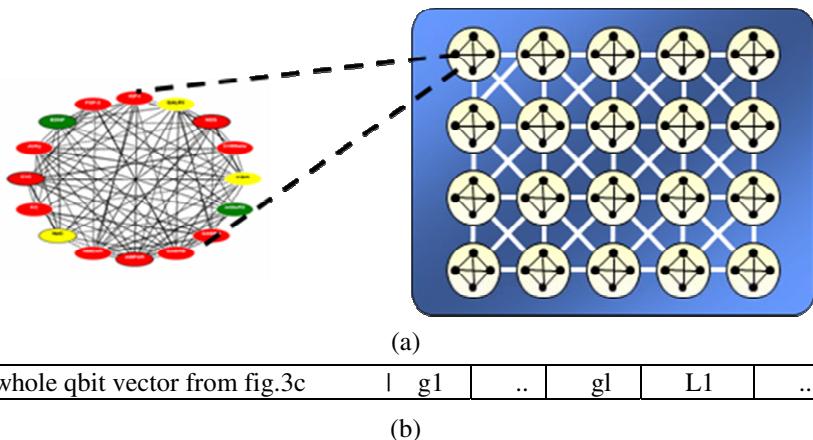


Fig. 4. (a)A schematic diagram of an ipSNG model (similar to the CNGM from [5]; (b) In addition to the parameters shown in fig.3c, an ipESNG represents in a qbit register also gene expression levels (g_1, g_2, \dots, g_l) and the connections between the genes (L_1, \dots, L_r) from a GRN

from 2.3. The qbit vector for optimization through the QEA is given in fig.4b. In addition to the ipESNN parameters from fig.3c, the ipESNG model has gene expression parameters g_1, g_2, \dots, g_l , each of them also represented as a qbit with two states (state “1” – gene is expressed; state “0” – gene is not expressed”0). Each link L_i ($i=1,2,\dots,r$) between two genes in the GRN is represented as a quantum bit with 3 states (“1” positive connection; “0” – no connection; “-1” – negative connection).

5 Open Questions and Future Research

A significant originality of the proposed ipSNN models is that for the first time elements from the mathematical apparatus of quantum computation are used to develop a probabilistic ESNN model. Both input features, SNN parameters and output neuronal activities are represented as an integrated qbit vector defined by a probability wave function and used to calculate *probabilistically* the state of the whole system at any given time. The learning process of an ipSNN involves not only changing the connection weights by using learning rules, but also changing the probability density function of the qbit vector by the application of a quantum-inspired evolutionary algorithm [11]. The proposed ipSNN framework will be used for the development of a feed-forward, and a recurrent models as shown in fig.3a,b,c. The feed-forward ipSNN will be tested on a large-scale multimodal audiovisual information processing task [47]. Our hypothesis is that the ipSNN model will be faster and more accurate than other techniques, revealing useful information such as new features. The recurrent ipSNN model will be used to create and test an associative memory that is anticipated to be capable of storing and retrieving much larger number of patterns when compared to traditional associative memories [13,21]. Other QEA will be tested for probability estimation, such as quantum inspired particle swarm optimization. These models will also be used for a single neuron modeling on data from a *2-photon Laser Laboratory* [31] with the hypothesis that they are applicable to modeling processes in a neuron. We expect that new information about learning processes in the brain can be discovered.

Acknowledgement. Stefan Schliebs and Peter Hwang for technical support, Prof. H. Kojima and Prof. A. Villa for fruitful discussions.

References

1. Abbott, L.F., Sacha, B.: Synaptic plasticity: taming the beast. *Nature Neuroscience* 3, 1178–1183 (2000)
2. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. *Cognitive Science* 9, 147–169 (1985)
3. Arbib, M. (ed.): *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge (2003)
4. Belatreche, A., Maguire, L.P., McGinnity, M.: Advances in Design and Application of Spiking Neural Networks. *Soft Comput.* 11(3), 239–248 (2006)
5. Benuskova, L., Kasabov, N.: *Comput. Neurogenetic Modelling*. Springer, NY (2007)

6. Bershadskii, A., et al.: Brain neurons as quantum computers: *in vivo* support of background physics. *Reports of the Bar-Ilan University, Israel*, vol. 1-12 (2003)
7. Brette, R., et al.: Simulation of networks of spiking neurons: A review of tools and strategies. *Journal of Computational Neuroscience* 23(3), 349–398 (2007)
8. Castellani, M.: ANNE - A New Algorithm for Evolution of ANN Classifier Systems. In: *IEEE Congress on Evolutionary Computation, CEC 2006*, pp. 3294–3301 (2006)
9. Dayan, P., Hinton, G.E.: Varieties of Helmholtz machines. *Neural Networks* 9, 1385–1403 (1996)
10. Dayan, P., Hinton, G.E., Neal, R., Zemel, R.S.: The Helmholtz machine. *Neural Computation* 7, 1022–1037 (1995)
11. Defoin-Platel, M., Schliebs, S., Kasabov, N.: Quantum-inspired Evolutionary Algorithm: A multi-model EDA. *IEEE Trans. Evolutionary Computation* (in print, 2009)
12. Deutsch, D.: Quantum computational networks. *Proceedings of the Royal Society of London A*(425), 73–90 (1989)
13. Ezhov, A., Ventura, D.: Quantum neural networks, in *Future Directions for Intelligent Systems and Information Sciences*. In: Kasabov, N. (ed.) *Future directions for intelligent systems*. Springer, Heidelberg (2000)
14. Gerstner, W., Kistler, W.M.: *Spiking Neuron Models*. Cambridge Univ. Press, Cambridge (2002)
15. Gerstner, W.: What's different with spiking neurons? In: Mastebroek, H., Vos, H. (eds.) *Plausible Neural Networks for Biological Modelling*, pp. 23–48. Kl. Ac. Publ., Dordrecht (2001)
16. Guyon, I., et al. (eds.): *Feature Extraction, Foundations and Applications*. Springer, Heidelberg (2006)
17. Han, K.-H., Kim, J.-H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Trans. on Evolutionary Computation*, 580–593 (2005)
18. Hey, T.: Quantum computing: an introduction. *Comp. & Control Eng. J.* 10(6) (1999)
19. Hinton, G.E., Dayan, P., Frey, B.J., Neal, R.: The wake-sleep algorithm for unsupervised neural networks. *Science* 268, 1158–1161 (1995)
20. Hirvensalo, M.: *Quantum computing*. Springer, Heidelberg (2004)
21. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* 79, 2554–2558 (1982)
22. Huguenard, J.R.: Reliability of axonal propagation: The spike doesn't stop here. *PNAS* 97(17), 9349–9350 (2000)
23. Izhikevich, E., Desai, N.: Relating STDP to BCM. *Neural Comp.* 15, 1511–1523 (2003)
24. Izhikevich, E.: Simple model of spiking neurons. *IEEE Tr. NN* 14(6), 1569–1572 (2003)
25. Izhikevich, E.: Which model to use for cortical spiking neurons? *IEEE Tr. NN* 15(5), 1063–1070 (2004)
26. Kasabov, N.: *Evolving Connectionist Systems: The Knowl. Eng. Appr.* Springer, Heidelberg (2007)
27. Kasabov, N.: *Integrative Connectionist Learning Systems Inspired by Nature: Current Models, Future Trends and Challenges*. Natural Computing. Springer, Heidelberg (2008)
28. Kasabov, N.: Brain-, Gene-, and Quantum Inspired Computational Intelligence: Challenges and Opportunities. In: Duch, W., Manzduk, J. (eds.) *Challenges in Computational Intelligence*, pp. 193–219. Springer, Heidelberg (2007)
29. Kasabov, N.: Evolving Intelligence in Humans and Machines: Integrative Evolving Connectionist Systems Approach. *IEEE Computational Intelligence Magazine* 3(3), 23–37 (2008)

30. Kasabov, N.: Found. of neural networks, fuzzy systems and knowl. eng. MIT Press, Cambridge (1996)
31. Katsumata, S., Sakai, K., Toujoh, S., Miyamoto, A., Nakai, J., Tsukada, M., Kojima, H.: Analysis of synaptic transmission and its plasticity by glutamate receptor channel kinetics models and 2-photon laser photolysis. In: Proc. of ICONIP 2008. LNCS. Springer, Heidelberg (2009)
32. Kohonen, T.: Self-Organizing Maps, 2nd edn. Springer, Heidelberg (1997)
33. Kistler, G., Gerstner, W.: Spiking Neuron Models - Single Neurons, Populations, Plasticity. Cambridge Univ. Press, Cambridge (2002)
34. Maass, W., Bishop, C. (eds.): Pulsed Neural Networks. MIT Press, Cambridge (1999)
35. Pavlidis, N.G., Tasoulis, O.K., Plagianakos, V.P., Nikiforidis, G., Vrahatis, M.N.: Spiking neural network training using evolutionary algorithms. In: Proceedings IEEE International Joint Conference on Neural Networks, vol. 4, pp. 2190–2194 (2005)
36. Pfister, J.P., Barber, D., Gerstner, W.: Optimal Hebbian Learning: a Probabilistic Point of View. In: Kaynak, O., Alpaydin, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 92–98. Springer, Heidelberg (2003)
37. Bohte, S.M., La Poutré, H.A., Kok, J.N.: Error-Backpropagation in Temporally Encoded Networks of Spiking Neurons. Neurocomputing 48(1-4), 17–37 (2002)
38. Bohte, S.M., Kok, J.N.: Applications of spiking neural networks. Information Processing Letters 95(6), 519–520 (2005)
39. Schliebs, S., Defoين-Platel, M., Kasabov, N.: Integrated Feature and Parameter Optimization for an Evolving Spiking Neural Network. In: Proc. of ICONIP 2008, Auckland, NZ. LNCS, vol. 5506/5507. Springer, Heidelberg (2009)
40. Soltic, W.S., Kasabov, N.: Evolving spiking neural networks for taste recognition. In: Proc. WCCI 2008, Hong Kong. IEEE Press, Los Alamitos (2008)
41. Specht, D.F.: Enhancements to probabilistic neural networks. In: Proc. Int. Joint Conference on Neural Networks, June 1992, vol. 1, pp. 761–768 (1992)
42. Tuffy, F., McDaid, L., Wong Kwan, V., Alderman, J., McGinnity, T.M., Kelly, P., Santos, J.: Spiking Neuron Cell Based on Charge Coupled Synapses. In: Proc. IJCNN, Vancouver (2006)
43. Ventura, D., Martinez, T.: Quantum associative memory. Information Sciences 124(1-4), 273–296 (2000)
44. Verstraeten, D., Schrauwen, B., Stroobandt, D., Van Campenhout, J.: Isolated word recog. with the Liquid State Machine: a case study. Inf. Proc. Letters 95(6), 521–528 (2005)
45. Villa, A.E.P., et al.: Cross-channel coupling of neuronal activity in parvalbumin-deficient mice susceptible to epileptic seizures. Epilepsia 46(suppl. 6), 359 (2005)
46. Wysoski, S., Benuskova, L., Kasabov, N.: On-line learning with structural adaptation in a network of spiking neurons for visual pattern recognition. In: Kollias, S.D., Stafloulopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 61–70. Springer, Heidelberg (2006)
47. Wysoski, S., Benuskova, L., Kasabov, N.: Brain-like Evolving Spiking Neural Networks for Multimodal Information. In: Proc. ICONIP 2007, Kitakyushu. LNCS. Springer, Heidelberg (2007)
48. Yadav, A., Mishra, D., Yadav, R.N., Ray, S., Kalra, P.K.: Time-series prediction with single integrate-and-fire neuron. Applied Soft Computing 7(3), 739–745 (2007)

A Spiking Network of Hippocampal Model Including Neurogenesis

Yusuke Tabata and Masaharu Adachi

Graduate School of Engineering, Tokyo Denki University,
2-2 Kanda-Nishiki-cho, Chiyoda-ku, Tokyo 101-8457, Japan
07gmd91@ms.dendai.ac.jp, adachi@eee.dendai.ac.jp

Abstract. In this paper, we construct a spiking network model based on the firing-rate coding hippocampal model proposed by Becker. Basal training patterns are presented to the model network and spiking self organizing map learning is applied to the network in order to store the training patterns. We then apply a morphogenesis model in the dentate gyrus region to generate new neurons and investigate the influence of such neurogenesis on the storage and recall of novel memory. As a result, the storage capacity is essentially unchanged by the morphogenetic algorithm even when the number of training patterns is changed.

1 Introduction

The hippocampus was first considered to play a role in short-term memory following a report of a patient in whom partial lesion of the temporal lobe resulted in short-term memory loss. Yonelinas subsequently evaluated the memory of hypoxia-ischemia patients using a cognitive test [1] and proposed a relationship between short-term memory and hippocampal activity. Becker reproduced similar results by computer simulation [2], thereby supporting the hypothesis put forward by Yonelinas. Becker also examined the influence of neurogenesis in the dentate gyrus (DG) region of the hippocampus on the acquisition of novel memory. However, Becker's model consists of simple rate coding model neurons that are not biologically plausible. In this work, we apply a morphogenesis model proposed by Butza *et al.* [3] to the DG region in an attempt to make connections between generated and existing neurons that are biologically plausible in the DG. We introduce a spiking neuron and morphogenesis model to the hippocampal model in order to examine the effects of neurogenesis on novel memories by numerical experiments.

2 Hippocampal Model

2.1 Structure of the Model Network

In this study, we construct a hippocampal model based on Becker's aforementioned network model consisting of rate coding neurons. We introduce a spiking

neuron model to the network model in order that there are connections with appropriate spike transmission delays between regions in Becker's hippocampal model. We then examine numerically the influences of neurogenesis in the DG on novel memory acquisition.

2.2 Constituent Neuron Model

We use the model proposed by Izhikevich [4] to describe constituent neuron dynamics. Though this model is a simple second-order differential equation, it can reproduce much of the behavior of real neurons depending on the values of parameters.

2.3 Self-organization of Spiking Neurons

In order to store basal memories in the hippocampal model, we apply self organizing map (SOM) learning which uses only spike intervals [5]. Here, the updating amount of the weight w_{ij} from neuron j to neuron i of competitive neurons in the SOM is expressed as follows.

$$\Delta w_{ij} = \eta \frac{T_{out} - t_i}{T_{out}} (s_j - w_{ij}) \quad (1)$$

where η denotes the learning constant, T_{out} denotes the standard firing time, s_j is the input spike strength from j th neuron (it is a fixed value $s_j = 1$ ($t = t_j$), $s_j = 0$ (otherwise) in this paper), and t_i is the firing time of i th neuron. Furthermore, competitive neurons have excitatory and inhibitory lateral connections in the same regions, and the lateral connection weight \tilde{w}_{ik} of neuron i connected from neuron k is expressed in the following equation.

$$\tilde{w}_{ik} = \begin{cases} 0 & (|i - k| = 0), \\ 5 & (|i - k| = 1), \\ 0 & (|i - k| = 2), \\ -|i - k| & (|i - k| \geq 3). \end{cases} \quad (2)$$

The lateral connection weights based on the distance between neurons work similarly to the neighborhood function of conventional SOM.

3 Morphogenesis Model

In this study, the morphogenesis model proposed by Butza *et al.* [3] is applied to the DG region. This model adds apoptosis and neurogenesis to the algorithm of Dammasch *et al.* [6], enabling us to simulate apoptosis and neurogenesis in the DG region. Thus, we expect that the structure and the activity of the DG will be efficiently modified by the morphogenesis model.

3.1 Overview of Morphogenesis

The morphogenetic algorithm used in this paper consists of a short-term procedure and a long-term one. In the former, neurogenesis occurs and neuronal activities in the DG network are observed. In the latter, we use the observed neuronal activities to update the existing connection weights in the DG network. In addition, the newly generated neurons and existing neurons may form new connections in the latter procedure. In the morphogenetic algorithm, neurons in the network are classified as high-state neurons, low-state neurons, stable state neurons, or neurons to be deleted, according to their activity. The synaptic weights of the input to a high-state neuron are decreased at a rate determined by the summation of the neuron's synaptic weights and its activity [3]. On the contrary, in the case of a low-state neuron, the synaptic weights are increased at a rate determined in the same manner as for the high-state neuron. In this paper, we introduce spiking neurons to the morphogenetic algorithm, thus modifying the original method given in [3] in order to determine neuronal activity.

3.2 Acquisition of the Neuronal Activity

The activity a_i of i th neuron is determined by the average membrane potential of the Izhikevich model in the short-term procedure. The average membrane potential p_i of i th neuron is expressed as follows.

$$p_i = \frac{v_i + \zeta}{\Delta t} \quad (3)$$

where v_i denotes the membrane potential of i th neuron. ζ denotes bias of the membrane potential, whose value is common to all neurons in the network. The state value of the neuron is given by a sigmoid function of p_i as follows.

$$a_i = \frac{1}{1 + e^{\frac{p_i + \alpha}{\beta}}} \quad (4)$$

where α and β denote the parameters for adjusting center and steepness of the sigmoid function, respectively.

3.3 Classification of the Morphogenetic State of Neurons

As mentioned earlier, we use the morphogenesis model proposed by Butza *et al.* [3]. The morphogenetic state of neurons is classified as high state, low state, stable state, or to be deleted according to the activity of the neuron as follows.

- (a) Neurons whose activity is $a_i < 0.25$ or $a_i > 0.75$ are classified as to be deleted and they are deleted in the next morphogenetic step.
- (b) Neurons of $0.25 \leq a_i < 0.45$ are classified as low-state neurons.
- (c) Neurons of $0.45 \leq a_i \leq 0.55$ are classified as stable state neurons.
- (d) Neurons of $0.55 < a_i \leq 0.75$ are classified as high-state neurons.

These classifications are intended to ensure neurons whose activities are far from the middle of their dynamic range greatly influence the network. For high-state neurons, the generation of synaptic connections is inhibited, and the existing synaptic weights are decreased or the connections themselves are extinguished. For low-state neurons, the generation of synaptic connections is activated, and the weights of existing synaptic connections are increased. Therefore, in this morphogenetic algorithm, the activity of an individual neuron plays an important role [3].

4 Method of Numerical Experiments

In our hippocampal model, we used 100 Izhikevich-type neurons in each region. In addition, the connection weights between regions are initialized with pseudo-random numbers between 1.5 and 2.5. The learning rates used in eq.(II), in DG, CA3, CA1 receiving the spikes from ECout are set to $\eta^{DG,ECout} = \eta^{CA3,ECout} = \eta^{CA1,ECout} = 0.21$. In addition, the learning rates of ECin from CA1 and CA1 from CA3 are set to $\eta^{ECin,CA1} = 0.15$ and $\eta^{CA1,CA3} = 0.05$, respectively.

4.1 Construction of Learning Patterns

Each training pattern is formed in excitable neurons equivalent to 10% of the number of the neurons firing in ECin. The spike strength is set to $s = 1$, and the intervals of spikes are set with pseudo-random numbers between 42 and 44.

4.2 Training Sequence of Stored Patterns

In the following numerical experiments, training consists of two phases. In the first phase, basal training patterns are presented to the model network without any connections from DG neurons to CA3 ones. In the second phase, additional patterns are presented to the model network after connections from DG neurons to CA3 neurons are generated. The patterns presented in the first and second phases are referred as PatternSet 1 and 2, respectively. In both the first and second phases each pattern is presented 50 times during 700[ms] for each phase.

4.3 Evaluation of Recall

In order to investigate the influence of neurogenesis on the recall of the basal training pattern, we performed the following numerical experiments. After training phase 1, the firing patterns of the ECout neurons to each input pattern of PatternSet1 were recorded. Next, firing patterns of the ECout neurons after the 30th iteration of phase 2 training were recorded. Phase 2 training was stopped after the 50th iteration of each training input pattern. Then, the input patterns of PatternSet1 and PatternSet2 were presented to the ECin neurons. For each input pattern, the recall of the stored pattern was considered successful when the firing pattern of the ECout neurons to the input coincided more than 90% with the corresponding recorded pattern.

4.4 Application of Morphogenetic Algorithm to the Hippocampal Model

We apply the morphogenetic algorithm to the DG region in the hippocampal model which successfully stores PatternSet1. The parameters for the algorithm are set as follows: $\Delta t = 200[\text{ms}]$, and the duration of the long-term procedure is set to $400[\text{ms}]$. The number of initial neurons is 30, and the maximum number of neurons in the DG network is 100. The number of initial connections from a newly generated neuron is 20, and the connection weights are set by a random number in the range of [50, 60]. For neuron activity acquisition, parameters are set to $\alpha = -90$, $\beta = -10$, $\zeta = 170$. In the training of patterns to be stored, apoptosis of neurons in the DG network is inhibited for the duration of pattern presentation, that is, the inhibition duration is set to $3.5[\text{s}]$. Moreover, one neuron is generated for every pattern presentation duration (i.e., $700[\text{ms}]$) in the DG network.

4.5 Evaluation of Connections from Generated Neurons to CA3

In this study, the neurons generated in the DG region by the morphogenetic algorithm are connected to CA3. The CA3 neurons to be connected from the generated neurons in DG are chosen at random.

5 Results

We have introduced the spiking neuron model to Becker's hippocampal network model and the morphogenetic algorithm to the DG network in order to focus on neuronal activity in DG network with the morphogenetic algorithm. The results of numerical experiments on memory storage and recall in the network are as follows.

The number of firings and the number of synaptic connections of the neurons in the DG network fluctuated when training PatternSet1 was presented, as

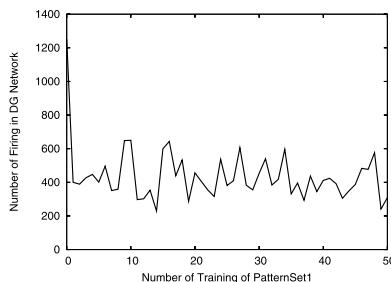


Fig. 1. Number of firings in the DG network during training of PatternSet1

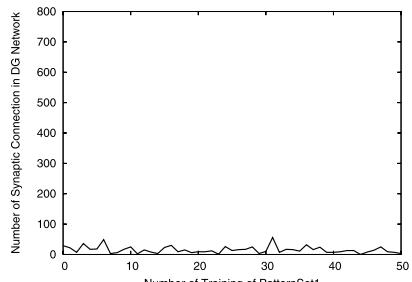


Fig. 2. Number of synaptic connections in the DG network during training of PatternSet1

shown in Figs. 1 and 2, respectively, even though the patterns of PatternSet1 were presented periodically. After the training of PatternSet1, the connections from neurons to CA3 neurons were generated as described in Sec. 4.5. After the training of PatternSet2a, which consisted of two patterns, it can be seen from Figs. 3 and 4 that the number of the firings and the number of synaptic connections of existing neurons, respectively, were monotonically increased by the neurogenetic algorithm. Such behaviors are different from those in the case of training PatternSet1. Figures 5 and 6 show the activity a_i of the DG neurons at the end of the presentation of PatternSet1 and PatternSet2a, respectively, where the degree of grayness of each cell represents the value of a_i . Neuronal activity shows a similar tendency to that seen in Figs. 1, 4. In other words, during the training of PatternSet1, the activities of neurons fluctuate as shown in Fig. 5, whereas in the training of PatternSet2a, the number of active neurons is monotonically increased as shown in Fig. 6. The network trained with PatternSet2a recalls on average $1.28(\pm 0.56)$ patterns out of the 2 patterns for 20 training trials.

We also examined the case when the number of PatternSet2b is four. The first phase of training of PatternSet1 is performed in the same manner as in the previous experiment before PatternSet2b consisting of four patterns is presented to the hippocampal model. From Figs. 7 and 8, we find that the number of firings and the number of the synaptic connections do not show monotonic increases but fluctuate in a certain range. From Fig. 9, we see from the extent of grayness indicating activity that some of the DG neurons show survival or extinction during the training of PatternSet2b.

The network trained with PatternSet2b recalls on average $2.52(\pm 1.29)$ patterns out of the 4 patterns for 20 training trials. This number of recalls is almost the same as for PatternSet2a since $1.28/2 \approx 2.52/4$.

Furthermore, the number of recalled patterns is shown in Table 1 in the cases that the number of training patterns is varied from 8 to 12.

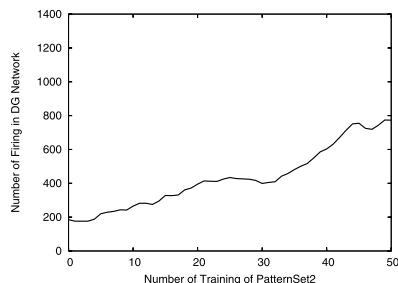


Fig. 3. Number of firings in the DG network during training of PatternSet2a consisting of 2 patterns

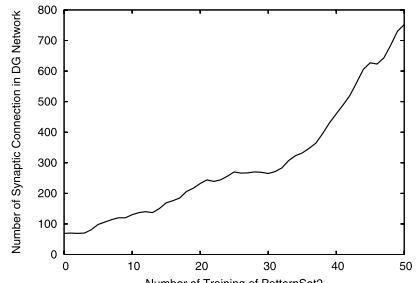


Fig. 4. Number of synaptic connections in the DG network during training of PatternSet2a consisting of 2 patterns

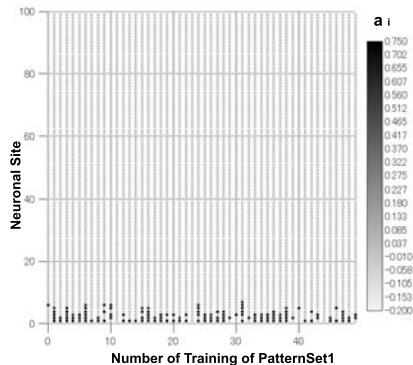


Fig. 5. DG neuronal activity during training of PatternSet1

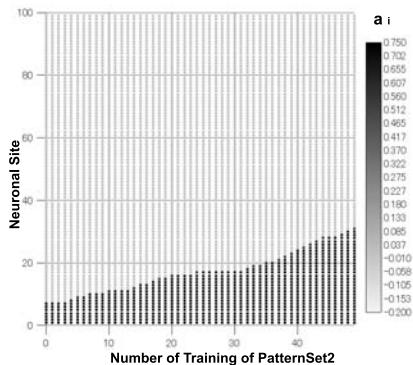


Fig. 6. DG neuronal activity during training of PatternSet2a consisting of 2 patterns

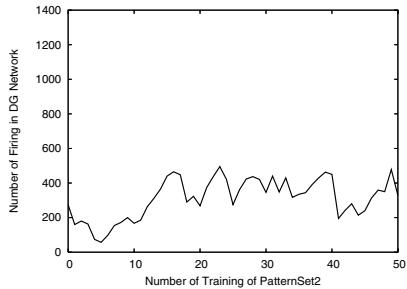


Fig. 7. Number of firings in the DG network during training of PatternSet2b consisting of 4 patterns

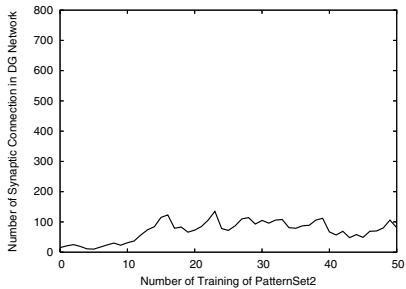


Fig. 8. Number of synaptic connections in the DG network during training of PatternSet2b consisting of 4 patterns

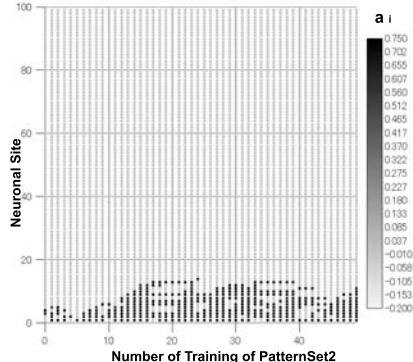


Fig. 9. DG neuronal activity during training of PatternSet2b consisting of 4 patterns

Table 1. Number of recalled patterns with 8 to 12 training patterns in the second training phase

	PatternSet2	PatternSet1
# of PatternSet2	average number of recalls	average number of recalls
8	5.29(± 1.28)	7.33(± 2.59)
9	5.60(± 1.94)	6.29(± 2.26)
10	6.24(± 1.80)	6.69(± 2.54)
11	6.57(± 1.88)	5.88(± 2.24)
12	7.17(± 2.38)	5.00(± 2.16)

6 Conclusions

In this study, we introduced a spiking neuron model and applied a morphogenetic algorithm to Becker's hippocampal model in the expectation that the structure and the activity of the DG network would become biologically plausible. As a result, we found that the activities and the state of morphogenesis are changed as the number of training patterns is changed. The result can be understood as follows. When the number of additional training patterns is small, the DG network can maintain the current state even though the morphogenetic algorithm is active. On the other hand, when the number of additional training patterns is large, it becomes difficult to maintain a steady state because the input patterns may have larger deviations than in the case of a small number of training patterns. In such case, the DG network adjusts its activity by changing the number and the weights of synaptic connections to cope with the input change, and the rates of recalled patterns in PatternSet2a and PatternSet2b are almost the same. Moreover, the storage capacity of the network is essentially unchanged regardless of the number of training patterns in the second training phase. In other words, the storage capacity is largely maintained by the morphogenetic algorithm.

In future, we plan to apply a precise morphogenesis model to the hippocampal model.

Acknowledgment. This work was supported by KAKENHI (No.18560386).

References

1. Yonelinas, A., Kroll, N., Quamme, J., Lazzara, M., Sauve, M., Widaman, K., Knight, R.: Effects of extensive temporal lobe damage or mild hypoxia on recollection and familiarity. *Nature Neuroscience* 5(11), 1236–1241 (2002)
2. Becker, S.: A Computational Principle for Hippocampal Learning and Neurogenesis. *Hippocampus* 15, 722–738 (2005)
3. Butza, M., Lehmannb, K., Dammaschc, I.E., Teuchert-Noodta, G.: A theoretical network model to analyse neurogenesis and synaptogenesis in the dentate gyrus. *Neural Networks* 19, 1490–1505 (2006)
4. Izhikevich, E.M.: Simple Model of Spiking Neurons. *IEEE Transactions on Neural Networks* 14(6), 1569–1572 (2003)
5. Ruf, B., Schmitt, M.: Self-Organization of Spiking Neurons Using Action Potential Timing. *IEEE Transactions on Neural Networks* 9(3), 575–578 (1998)
6. Dammasch, E., Wagner, G.P., Wolff, J.R.: Self-Stabilization of Neuronal Networks. *Biological Cybernetics* 54, 211–222 (1986)

NeuroEvolution Based on Reusable and Hierarchical Modular Representation

Takumi Kamioka^{1,2}, Eiji Uchibe¹, and Kenji Doya^{1,2,3}

¹ Okinawa Institute of Science and Technology, Okinawa, Japan
kamioka@oist.jp

² Nara Institute of Science and Technology, Nara, Japan

³ ATR Computational Neuroscience Laboratories, Kyoto, Japan

Abstract. The framework of neuroevolution (NE) provides a way of finding appropriate structures as well as connection weights of artificial neural networks. However, the conventional NE approach of directly coding each connection weight by a gene is severely limited in its scalability and evolvability. In this study, we propose a novel indirect coding approach in which a phenotypical network develops from the genes encoding multiple subnetwork modules. Each gene encodes a subnetwork consisting of the input, hidden, and output nodes and connections between them. A connection can be a real weight or a pointer to another subnetwork. The structure of the network evolves by inserting new connection weights or subnetworks, merging two subnetworks as a higher-level subnetwork, or changing the existing connections. We investigated the evolutionary process of the network structure using the task of double pole balancing. We confirmed that the proposed method by the modular developmental rule can produce a wide variety of network architectures and that evolution can trim them down to the most appropriate ones required by the task.

1 Introduction

Artificial neural networks are helpful both for designing adaptive machines and for understanding brain functions. While a wealth of learning algorithms are available for optimizing connection weights, finding an appropriate network architecture usually requires heuristics search. Neuroevolution (NE) is a framework for optimizing the architecture as well as the connection weights of artificial neural networks [1, 2]. The conventional NE approach is to encode each component of the connection weight matrix by a gene and to evolve the network structure by inserting a new row or a column to the matrix [3]. However this *direct* genetic encoding scheme naturally has a limitation in scalability, because the dimension of the search space becomes high as the network evolves large [4]. An alternative approach is the *indirect* genetic coding which assumes a certain “developmental” process to convert a given genotype into a grown phenotypic network [5, 6, 7, 8, 9, 10]. Such indirect coding enables modular organization of the network and allows single genes to take global effects on the network function, which contribute to better *evolvability* [10, 11].

In this paper, we propose a novel developmental process for indirect genetic coding that enables reuse of network modules in a hierarchical manner. While previous indirect coding schemes, such as the cellular encoding (CE) [12], modularNEAT [13], and the embedded cartesian genetic programming (ECGP) [14] could encode modular networks, none of them could efficiently represent networks with hierarchical structures. In our proposed method, each gene encodes a subnetwork consisting of the input, hidden, and output nodes and connections between them. A connection can be either a real weight or a pointer to another subnetwork. The structure of the network evolves by inserting new connection weights or subnetworks, merging two subnetworks as a higher-level subnetwork, or changing the existing connections. By allowing a subnetwork to have connections realized as another network, our genetic encoding scheme enables a compact representation of networks with multiple levels of hierarchy.

In the following sections, we describe our genetic encoding and network development schemes and the methods for evolutionary optimization of the network structures and connection weights. We then investigate the evolutionary process of the network structure using the task of double pole balancing.

2 Representation of Reusable and Hierarchical Module

2.1 Genetic Representation

The genotype consists of two dynamic arrays to represent subnetworks and weights. Each subnetwork is composed of 2 types of genes (See Figure 1). The **nodes gene** represents the number of input, output, and hidden neurons in the subnetwork. The **connection genes** encode connections between the nodes in the subnetwork. The connection is represented by a pointer to a real weight or another subnetwork. If the pointer refers to a subnetwork, the input and output nodes indexed in the connection gene are linked to the nodes in the referred subnetwork. The genotype represents network by hierarchically using subnetwork. The first subnetwork (subnetwork 0) plays a special role of interfacing with the sensory input and the motor output for a given problem.

2.2 Network Development

The entire network develops by decoding the genome starting from the subnetwork 0. Each subnetwork develops as follows.

1. Prepare the number of input, output, and hidden neurons in the subnetwork.
2. Assign a unique number to all nodes in the order of input, output and hidden nodes.
3. Connect nodes according to connection genes.
 - (a) If the pointer point to a weight, make the weighted connection between indicated nodes.
 - (b) If the pointer points to a subnetwork, develop the pointed subnetwork and unify the input and output nodes of the developed subnetwork with the nodes specified in the connection gene.

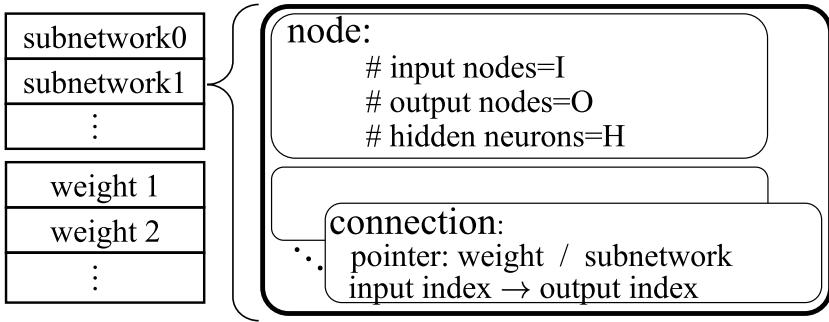
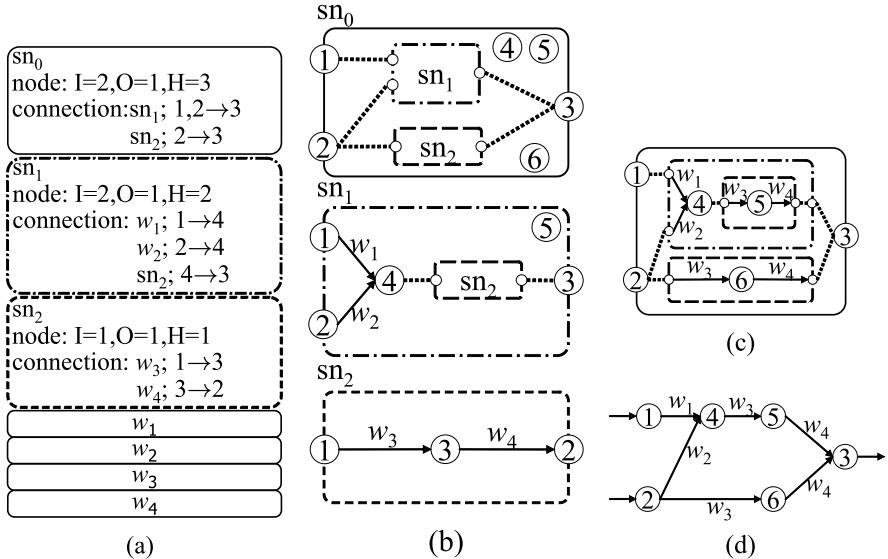
**Fig. 1.** Genotype**Fig. 2.** An example of genotype-phenotype mapping. (a) genotype, (b) developed subnetworks, (c) unified entire network, and (d) functional phenotype.

Figure 2 shows an example of genotype-phenotype mapping. Figure 2(b) shows each subnetwork represented by the genotype shown in Figure 2(a). Dotted lines represent unity and the nodes connected by dotted lines are functionally one node. Figure 2(c) shows the modularly composed entire network and Figure 2(d) shows the corresponding functional network.

3 Optimization

A critical issue in designing the evolutionary process is the “competing convention problem” which happens crossover operations between genotypes encoding very different phenotype [15, 16].

In order to avoid this problem we do not use any crossover operators and use only mutation operators. We take two optimizing phases called a structural exploration and weight improvement as in EANT¹⁷ and EANT2¹⁸.

3.1 Structural Exploration

In the phase of structural exploration, we prepare six genetic operations which are categorized into two classes. The following operations is to explore new structures:

- *add new weight*: Add a new weight and make a new connection. (Figure 3(b)).
- *add new subnetwork*: Add a new subnetwork and make a new connection. (Figure 3(c)).
- *integrate subnetworks*: Make a new subnetwork integrating two subnetworks. (Figure 3(d)). This operation do not directly affect phenotype. This makes a chance to connect between two subnetworks.

On the other hand, the following three operations is to explore the reuse of existing modular structures:

- *add connection*: Make new connection by a existing subnetwork (Figure 3(e)).
- *replace connection*: Replace a connection with a existing subnetwork, (Figure 3(f)).
- *delete connection*: Delete a connection gene (Figure 3(g)).

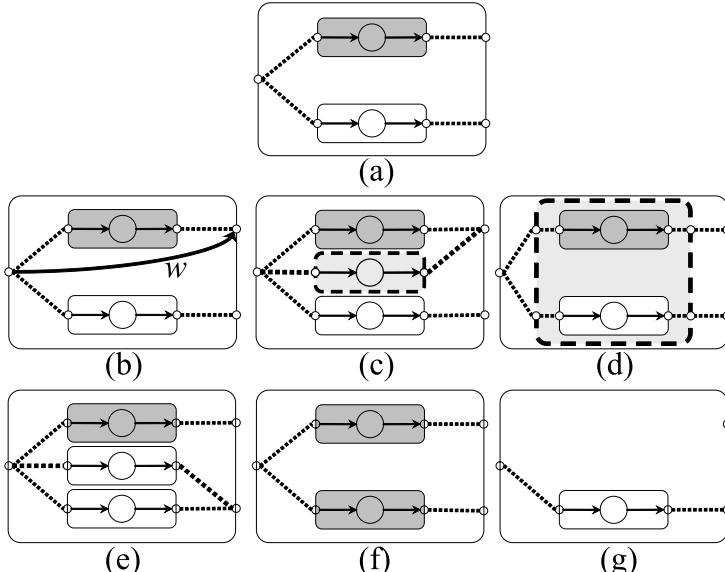


Fig. 3. Genetic operations for structural exploration: (a) original subnetwork, (b) add new weight, (c) add new subnetwork, (d) integrate subnetworks, (e) add connection, (f) replace connection, and (g) delete connection

Operations to explore reuse of existing modular structures are only executed to subnetwork 0 in order to avoid looped reference.

3.2 Weight Optimization (CMA-ES)

In the weight improvement phase, it is possible to apply any parameter optimization methods. Here, we select a method called covariance matrix adaptation (CMA-ES) [19]. CMA-ES is a variant of evolution strategy (ES) that avoids random adaptation of the strategy parameters and it is known that CMA-ES can find good solutions much faster even though the parameter space is a high dimensional space.

We apply $(\mu/\mu, \lambda)$ CMA-ES with population size $\lambda = 4 + 3 \ln n$, the number of offspring $\mu = \lambda/2$ (recommended by Hansen and Ostermeier [19]) and minimum variance $\sigma_{\min} = 0.05$ (used in Igel [20]). 4 termination criteria except problem solved are used as follows

- The total number of evaluations exceeds $900 * (n + 3) * (n + 3)$.
- The total number of generations exceeds 1000.
- The fitness function value differences are smaller than 0.001.
- The fitness function value differences of best values in all generations are smaller than 0.001.

4 Experiment

The pole balancing or inverted pendulum problem is a convenient benchmark for proof of concepts of artificial learning systems. Especially, the extended problem called Double Pole Balancing without Velocity (DPNV), which has 2 poles and only given the position of the cart and angles of each poles as inputs has often been used in the field of NE [21, 3, 20, 7, 17, 22]. The solution of DPNV needs at least one recurrent connection because it is a non-Markovian task.

4.1 Double Pole Balancing without Velocity

The DPNV consists of two poles hinged to a wheeled cart on finite stretch of track that must be balanced by applying a force to the cart at regular intervals. The inputs to the ANN are (x, θ_1, θ_2) which represent the position of cart, the angle of the first pole and the angle of the second pole, respectively. The applied forced is proportional to the output of the ANN. The DPNV is regarded as a non-Markovian task since velocities of the cart and two poles are not available as inputs to the network. In order to balance two poles simultaneously, it is known that at least one recurrent connection is required in the network.

We adopt a fitness function and termination conditions introduced by [21]. The fitness function is a weighted sum of two separate fitness measurements $0.1f_1 + 0.9f_2$ taken over 1000 time steps.

$$f_1 = t/1000,$$

$$f_2 = \begin{cases} 0 & \text{if } t < 100 \\ \frac{0.75}{\sum_{i=t-100}^t (|x^i| + |\dot{x}^i| + |\theta_1^i| + |\dot{\theta}_1^i|)} & \text{otherwise} \end{cases}$$

where t is the number of time steps the poles was balanced. At the beginning of the trial, state variables $\{x, \theta_1, \theta_2\}$ are initialized within the range $x \in [-2.4, 2.4]$, $\theta_1, \theta_2 \in [-36^\circ, 36^\circ]$ from a fixed initial position (set to zero except $\theta_1 = 4.5^\circ$).

One trial ends when the best individual in the population passes the following two tests. The first test is to balance two poles for 10^5 time steps starting from 4.5° initialization. The second test is to balance two poles for 1000 time steps starting from at least 200 out of 625 different initial starting states. Initial start positions are $(x, \dot{x}, \theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2) \in \{4.32k_1 - 2.16, 2.70k_2 - 1.35, 0.123k_3 - 0.062, 0.3k_4 - 0.15, 0, 0\}$ where each of k_1, k_2, k_3 , and k_4 takes one of five values $\{0.05, 0.25, 0.5, 0.75, 0.95\}$.

The pole-balancing environment was implemented using a realistic physical model with friction, and fourth-order Runge-Kutta integration with a step size of 0.01s. Mostly examined setting are used for equations of motion and parameters (see [22]).

5 Simulation Results

We used a small population size for structural exploration ($N = 10$). Individuals in the initial population had no hidden nodes and all input nodes were directly connected to the output node. Connection weights of each individual were optimized by CMA-ES. After optimizing the connection weights by CMA-ES, a roulette selection strategy was applied to select individuals. The genetic operations described in Section 3.1 were applied into selected individuals according to the mutation rates p_n and p_r . The mutation rate p_n for searching the new structure was set as $p_n = 0.8$. Since the mutation rate p_r for searching the reuse of existing structure plays a crucial role, we tested three values $p_r \in \{0.0, 0.2, 0.4\}$.

Experimental results in Table 1 shows the number of evaluations for several NE methods. These results were averaged over three simulation runs. It should be noted that CMA-ES optimized the connection weights for the fixed topology while CE, NEAT, AGE and our proposed method optimized both weights and topologies.

Table 1. The result of the DPNV. We tested the three proposed methods with different mutation rates: $p_r \in \{0.0, 0.2, 0.4\}$. CE [21], NEAT [3] and CMA-ES [20], were also evaluated for comparision.

Method	Evaluations
proposed($p_r = 0.4$)	55,910
proposed($p_r = 0.2$)	474,711
proposed($p_r = 0.0$)	683,931
CE	840,000
NEAT	24,543
AGE	25,065
CMA-ES	6,061

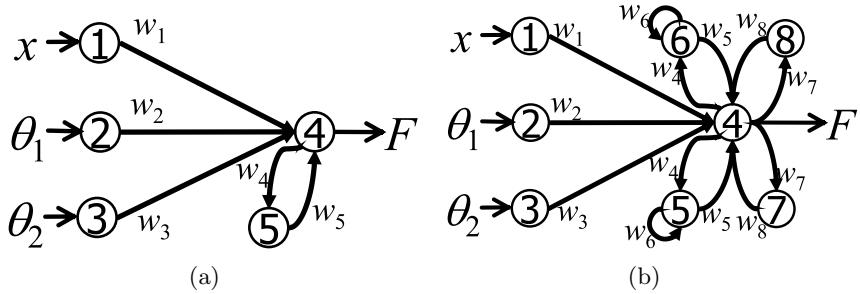


Fig. 4. Solutions of DPNV: (a) The simplest structure obtained by $p_r = 0.0$, and (b) The modular structure obtained by $p_r = 0.4$

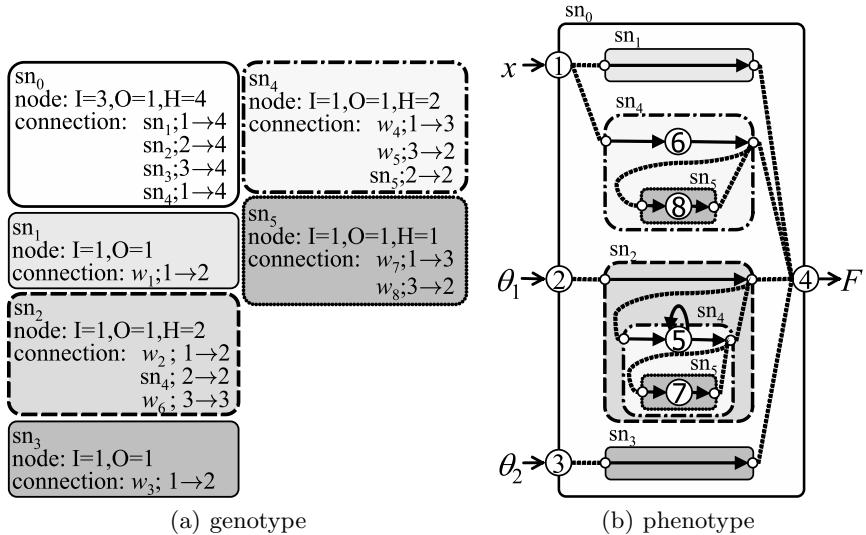


Fig. 5. Hierarchically reused modular structure

The solution obtained by the proposed method ($p_r = 0.0$) was the simplest structure which had only one hidden neuron (Figure 4(a)). In addition, we found that several modular structures (Figure 4(b)) evolved in a larger mutation rate for searching reuse of modules ($p_r = 0.4$). Two modules which are neural connection through nodes 5 and 7 are reused for recurrent connection of output neurons in nodes 6 and 8. Our proposed methods did not perform the best because the DPNV task could be solved without using a modular structure.

Our proposed method realized exploration of modular structures by reusing hierarchical structure. The modular structure shown in the Figure 5(b) was developed by a reuse of multi-level structure in the hierarchically represented module. The genotype encoding the modular structure is shown in the Figure 5(a).

All boxes represent subnetworks including nodes gene and connection genes. The subnetwork2 refers to subnetwork4 as a recurrent connection of the output node. Likewise, the subnetwork4 refers to subnetwork5. The module which is hierarchically represented by subnetwork2, 4 and 5 is developed in the connection between θ_1 and F shown by broken line box in the Figure 5(b). The subnetwork4 is also referred from x and F as the connection in subnetwork 0 from the node 1 to the node 4. Hierarchical modular representation is explored and reused in such a manner. The modular solution shown in the Figure 4(b) was obtained from the structure of Figure 5 by a single mutation of replacing the connection in subnetwork 0 from node 1 to node 4 referring the subnetwork4 with the subnetwork2.

6 Discussion and Future Work

In this paper, we proposed a novel indirect genetic coding and network development scheme that enables efficient exploration of hierarchical networks structures by allowing a network module to include a copy of another network module. We verified that networks with useful structural features, including symmetry, repetition, and hierarchy, can be developed and that the ones appropriate for a given task can be selected by evolution. The simple benchmark task we used here, DPNV, does not require modularly organized networks for control. Therefore, it is not surprising that the performance of the proposed method measured in the number of evaluations was not superior to other indirect gene coding methods.

One limitation in the present implementation is that subnetworks are organized in a tree shape. By allowing the same subnetwork to be reused by multiple subnetworks, or even by allowing recursive reference of subnetworks, more complex network structures can be described by shorter genes. However, this can potentially result in infinite growth in the development process and thus an appropriate regulatory mechanism may need to be introduced. Another important direction is combination of evolution of network structures with learning of connection weights. The weight optimization by CMA-ES can be regarded as one way of learning, but we can apply other learning frameworks, most notably the reinforcement learning [23]. In such combination, evolution of the parameters of learning [24] and possibly the learning algorithm itself are important topics of further research.

Acknowledgement

The implementation of the benchmark problem is directly based on source code by Kenneth O. Stanley. The implementation of the CMA-ES relies directly on source code by Nikolaus Hansen.

References

- [1] Miikkulainen, R.: Evolving neural networks. In: Procs. of Genetic and Evolutionary Computation Conference (GECCO 2007), pp. 3415–3434. ACM, New York (2007)

- [2] Floreano, D., Dürr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. *Evolutionary Intelligence* 1(1), 47–62 (2008)
- [3] Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10(2), 99–127 (2002)
- [4] Reisinger, J., Miikkulainen, R.: Acquiring evolvability through adaptive representations. In: Procs. of Genetic and Evolutionary Computation Conference (GECCO 2007), pp. 1045–1052. ACM, New York (2007)
- [5] Nolfi, S., Parisi, D.: “genotypes” for neural networks. In: *The handbook of brain theory and neural networks* table of contents, pp. 431–434 (1998)
- [6] Seys, C.W., Beer, R.D.: Effect of encoding on the evolvability of an embodied neural network. In: GECCO 2006 Workshop Procs., Workshop on Complexity through Development and Self-Organizing Representations (CODESOAR) (2006)
- [7] Dürr, P., Mattiussi, C., Floreano, D.: Neuroevolution with analog genetic encoding. In: Procs. of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX). LNCS, vol. 9, pp. 671–680. Springer, Heidelberg (2006)
- [8] Mattiussi, C., Floreano, D.: Analog genetic encoding for the evolution of circuits and networks. *IEEE Transactions on Evolutionary Computation* 11(5), 596–607 (2007)
- [9] Gauci, J., Stanley, K.O.: A case study on the critical role of geometric regularity in machine learning. In: Procs. of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008). AAAI Press, Menlo Park (2008) (to appear)
- [10] Kirschner, M., Gerhart, J.: Evolvability. *Proceedings of the National Academy of Sciences (PNAS)* 95(15), 8420–8427 (1998)
- [11] Raff, R.A., Sly, B.J.: Modularity and dissociation in the evolution of gene expression territories in development. *Evolution & Development* 2(2), 102–113 (2000)
- [12] Gruau, F.: Automatic definition of modular neural networks. *Adaptive Behaviour* 3(2), 151–183 (1995)
- [13] Reisinger, J., Stanley, K.O., Miikkulainen, R.: Evolving reusable neural modules. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 69–81. Springer, Heidelberg (2004)
- [14] Walker, J.A., Miller, J.F.: Automatic acquisition, evolution and re-use of modules in cartesian genetic programming. *IEEE Transactions on Evolutionary Computation* 12(4) (August 2008)
- [15] Schaffer, J.D., Whitley, D., Eshelman, L.J.: Combinations of genetic algorithms and neural networks: a survey of the state of the art. In: International workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN 1992), pp. 1–37 (1992)
- [16] Radcliffe, N.J.: Forma analysis and random respectful recombination. In: Procs. of the Fourth International Conference on Genetic Algorithms, pp. 222–229. Morgan Kaufmann Publishers, San Francisco (1991)
- [17] Kassahun, Y.: Towards a Unified Approach to Learning and Adaptation. PhD thesis, Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universität zu Kiel (2006)
- [18] Siebel, N.T., Sommer, G.: Evolutionary reinforcement learning of artificial neural networks. *International Journal of Hybrid Intelligent Systems* 4(3), 171–183 (2007)
- [19] Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 9(2), 159–195 (2001)
- [20] Igel, C.: Neuroevolution for reinforcement learning using evolution strategies. In: Procs. of Congress on Evolutionary Computation (CEC 2003), vol. 4, pp. 2588–2595. IEEE Press, Los Alamitos (2003)

- [21] Gruau, F., Whitley, D., Pyeatt, L.: A comparison between cellular encoding and direct encoding for genetic neural networks. In: Genetic Programming 1996: Procs. of the First Annual Conference, pp. 81–89. MIT Press, Cambridge (1996)
- [22] Gomez, F., Schmidhuber, J., Miikkulainen, R.: Accelerated neural evolution through cooperatively coevolved synapses. *The Journal of Machine Learning Research* 9, 937–965 (2008)
- [23] Elfwing, S., Uchibe, E., Doya, K., Christensen, H.I.: Evolutionary development of hierarchical learning structures. *IEEE Transactions on Evolutionary Computations* 11(2), 249–264 (2007)
- [24] Elfwing, S., Uchibe, E., Doya, K., Christensen, H.I.: Darwinian embodied evolution of the learning ability for survival. *Adaptive Behavior* (to appear)

A Common-Neural-Pattern Based Reasoning for Mobile Robot Cognitive Mapping

Aram Kawewong*, Yutaro Honda, Manabu Tsuboyama, and Osamu Hasegawa

Department of Computational Intelligence and Systems Science

Imaging Science and Engineering Laboratory

Tokyo Institute of Technology

4259-R2 Nagatsuta, Midori-ku, Yokohama, 228-8503

{kawewong.a.aa,honda.a.ab,tuboyama.m.aa,hasegawa.o.aa}@m.titech.ac.jp

Abstract. Neural Pattern-Based Reasoning for real-world robot navigation problems is proposed. Based on the concept of Pattern-Based Reasoning, the method enables a mobile robot to solve goal-oriented navigation problems in a remarkably short time with low memory consumption. Given a simple learning environment, the observed input vectors are processed by a Self-Organizing Incremental Neural Network (SOINN) to generate Spatial Common Patterns (CPs), which are useful in other unfamiliar environments. Performing goal-oriented navigation in unfamiliar environments, with prior information neither of the map nor the goal, the robot recognizes the partial area by referring to the nearest CPs and forming a pattern of CPs called A-Pattern. The sequential A-Patterns are used to derive the map of the environment. This map is optimized based on reasoning, as the new transitions between areas could be generated automatically. The method is evaluated by solving one real-world maze and three Webots simulated mazes. The results show that the proposed method enables the robot to find the markedly shorter path in only one episode, whereas use of the Reinforcement Learning requires more episodes. The map contains more information than the current hybrid map building or topological map. The map does not rely on coordinate, resulting in non-sensitivity to the error in self-pose estimation.

1 Introduction

The navigation problem is an important and attractive topic in robotics. Arriving at the unknown destinations in unseen environments in the shortest time is the most common objective of such studies. Reinforcement Learning is a popular choice for solving this problem. Unfortunately, reinforcement learning fundamentally requires numerous episodes for maximum reward convergence [8]. Reinforcement Learning might be suited to problems which are complex and difficult to solve by reasoning (so that it needs to be reinforced). Nonetheless, real-world robot navigation problems might not require such difficult reasoning. Later, the Simultaneous Localizing and Mapping (SLAM) becomes popular in

* Corresponding author.

the topic of robot navigation [11]. In any events, if the map could be acquired, then the shortest path would also be easily obtained. However, to acquire the map in the unknown environment, the robot must acquire a map of its environment while simultaneously localizing itself relative to this map [11]. The output of SLAM is generally the metric map. Because of its high memory consumption, the metric map is then converted into the topological map [4]. Though the Topological Map (TM) is suitable for high-level presentation, it lacks information about the places, so that the Hybrid-Map is proposed [2], [6]. The TM abstracts away the specific nature of sensory input, while the metric map precisely captures the structure of simple local region. Nonetheless, most of the map-building method requires the guided-exploration [11], and the transitions between places strongly rely on the orientation and position of the robot. Additionally, most map-building focus on dealing with the uncertainty in robot's pose estimation, so that the reasoning is not applied. If the robot could reason about the path, it would be able to know some new transitions between places without real tries, i.e. given $A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$ as the fact observed by the robot where A, B, C and D are places, it should also reasonably know the transition between D and B. As such, we are motivated to solve this problem by reasoning.

Regarding reasoning, although myriad approaches have been proposed, most remain insufficient for real mobile robots because they are symbol-based and therefore suitable to a specific domain. Reasoning based on patterns is proposed to eliminate that limitation. Sudo et al. [10] proposed the architecture for reasoning with pattern-based if-then rules. Different from other traditional reasoning methods, this architecture can process patterns as real-valued vectors and store pattern-based if-then rules of propositional logic. As portrayed in Fig. 1 (left), the architecture has an input layer and a competitive layer. The input layer obtains a fact (Conditional Rule \rightarrow Sequential Rule). New nodes are generated adaptively when the input layer obtains a fact. Each fact is represented with several nodes connected by edges. Those edges are different-type edges from those used to generate clusters. Nevertheless, although the authors claimed real-world problem application, their evaluation was based only on simulated visual data provided by humans. Considering real-world robot navigation problems, this architecture poses a disadvantage. Using an if-then rule as the input for the input layer lacks flexibility: it would be difficult for the robot to generate a new fact. For example, $A \rightarrow B$ may be a fact in only one environment. If the robot changed the environment in which $A \rightarrow B$ will never occur, it might get confusing as the previous fact is still remembered. Thus, the robot need to forget those facts obtained in the first environment before entering the new one, but this would prevent the robot from knowledge transferring; namely, the robot should reconnection A to other patterns instead of deleting it. Also, considering that the facts $A \rightarrow B$, $A \rightarrow C$, and $A \rightarrow D$ are given, it would be better to remember only one A, and link it to B, C and D, than remembering all new three facts.

Our proposed architecture, as portrayed in Fig. 1 (right), is based on the concept of common-patterns (CP). The CPs are self-organized from a simple preliminary environment. These CPs are used to represent the area in the third

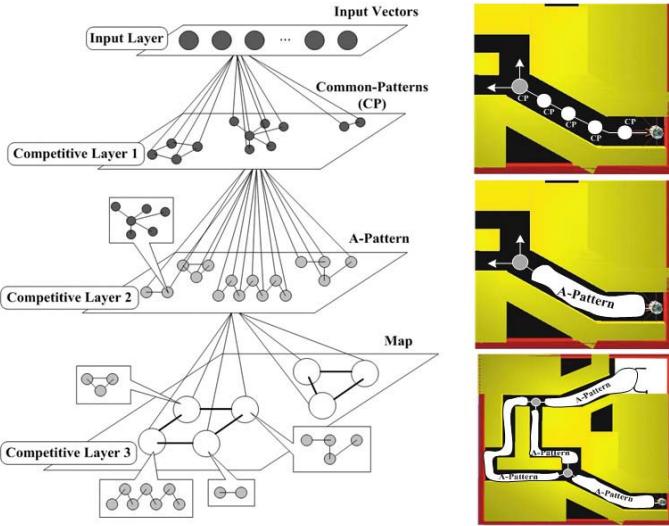


Fig. 1. (Left) The illustration of our proposing Common-Pattern-Based Reasoning Architecture. (Right) The illustration how the proposed architecture be applied to the navigation in maze.

layer, which we term A-Pattern. The map is the connection of the A-Patterns. The main contribution of this paper is to apply the *pattern-based reasoning* (PBR) to build the map of the unfamiliar environment, seek the unknown goal, convert each path fragment into the sequential common-patterns, and find the shortest path to reach the goal (once it has been found) in one go (one episode). Unlike the other topological map, the proposed method does not ignore most sensory data, but instead refer most of them to the existing common-patterns, resulting in low rate of memory consumption and containing more information. Also, the method does not require the guided exploration.

2 Proposed Architecture

As portrayed in Fig. 1, our architecture is consisted of four layers of two main stages. The first stage, called Learning Stage, is composed of two first layers. The robot is given a simple environment for generating the common-patterns (CPs) of spatial information (i.e., corner, transaction). The common-pattern, or CP, is a quantized vector for being referred in general environment. According to our concept, the robot learns a set of spatial CPs from a simple environment and transfers them to use in other different environment. The second stage, called Living Stage, is composed of the last two layers. The robot realize the unfamiliar environment by referring each observed input vectors to the most appropriated CP. This process is repeated until the junction is detected, and the area is represented by the pattern of CP (third layer in Fig.1 (left)). At

the junction, the robot makes decision, continues to next area, and repeats the process. Once the goal is discovered, the last layer would be formed by the connected A-Patterns, called Map. As the map may contains many duplicated A-Patterns, the map is optimized. The controller of the robot is done according to the Subsumption Architecture [3].

2.1 Generating Common-Pattern (Learning Stage)

We assume that some characteristics exist in every environment. The problem is for the robot to know these common characteristics without supervision. The answer is obtainable using SOINN: an unsupervised clustering method proposed by Shen and Hasegawa [9]. The SOINN's behavior relies on two main parameters: λ and age_{dead} . The SOINN inserts a new node and removes nodes of low probability density when the number of input signals is an integer multiple of λ . The parameter age_{dead} indicates the length of edges between nodes.

At the learning environment, the robot incrementally observes feature vectors x corresponding to the constant time step t . These vectors are processed (clustered) by SOINN with $\lambda=age_{dead}=20$. After a period of time, the clusters are derived (second layer of Fig.1 (Left)). Because this common characteristic would be used as the reference when solving the navigation problems, we call these characteristics spatial *common patterns* or CPs. Note that, generally, Common Pattern is not limited to only spatial information. It could also be applied with visual or sound information.

2.2 Area Representation and Map Generation (Living Stage I)

The confusion can occur easily if only a single CP is used for area representation. For that reason, the place is better represented by a pattern of CPs, which we term *A-Pattern*, denoted by \mathcal{A} , as portrayed in the third layer of Fig.1 (Left). Gradually and incrementally, the robot realize each individual area and connect them to generate the map \mathcal{M} , as shown in the last layer of Fig.1 (left). Considering in the context of reasoning, the CPs are chained based only on if-then rules. Every time the robot passes the individual area, it makes a reference by forming a A-Pattern; \mathcal{A} . This sequence would be chained incrementally in the form of $\mathcal{A}_1 \rightarrow \mathcal{A}_2; \mathcal{A}_2 \rightarrow \mathcal{A}_3; \dots$ until the destination is located. Chaining these \mathcal{A} forms the map \mathcal{M} .

2.3 Reasoning and Map Optimization (Living Stage II)

As the problem is the navigation, the map generation process ends once the goal is discovered. As the robot move arbitrarily, the map may contains many duplicated A-Patterns, so that the optimization is needed. Given $\mathcal{M} = \mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{n_p}$ as the map, minus sign ($-$) indicates the entrance direction of \mathcal{A}_i , where $i \in I^+, i \lesssim n_p$, while plus sign ($+$) indicates the exit direction. Note that, for any areas, the signs are assigned only at the first time of traversal. Given $junc()$ as the junction; $junc(\mathcal{A}_i^-)$ is the junction before \mathcal{A}_i , and $junc(\mathcal{A}_i^+)$ is the junction

after \mathcal{A}_i . Each \mathcal{A}_i is connected by the junction. The transition, which brings the robot from \mathcal{A}_i to \mathcal{A}_{i+1} , has been realized by the robot. Namely, from \mathcal{A}_i to \mathcal{A}_{i+1} , there exists the transition Δ_1 , where $i \in I^+, i < (n_p)$. The transition Δ can be either real value or vector. However, in this study, we simply define the transition as the integer indicating the index of choice made at the junction. The algorithm has two steps. First, 1) based on the derived M in which its element \mathcal{A}_i is connected by the initial transition Δ_i , the robot find the inversed transition, Δ_i^{-1} , by using the function Λ_1 . Given N_w as the number of the spaces at the junction: $junc(\mathcal{A}_i^+)$, where $junc(\mathcal{A}_i^+) = junc(\mathcal{A}_{i+1}^-)$, and $\Lambda_1(\Delta_i) = N_w - \Delta_i + 1 = \Delta_i^{-1}$. Second, 2) now that the robot knows all bidirectional transition between \mathcal{A} , it starts to search for the duplicated \mathcal{A} in \mathcal{M} . Since the real-world usually contains multiple loops, it is very likely that the robot pass the same junction more than once. Therefore it is essential to union those similar junctions in order to re-connect \mathcal{A} and eliminate all duplicated \mathcal{A} . To recognize the similar junctions, it needs to consider from the connected \mathcal{A} . Specifically, given \mathcal{A}_1 is approximately identical to \mathcal{A}_2 , then $junc(\mathcal{A}_1^+) = junc(\mathcal{A}_2^+)$ and $junc(\mathcal{A}_1^-) = junc(\mathcal{A}_2^-)$. The similarity between \mathcal{A}_1 and \mathcal{A}_2 are determined by function f_S given the following.

$$l(\mathcal{A}_1) : \text{length of } \mathcal{A} = a_1, a_2, \dots, a_{l(\mathcal{A}_1)}, a \in \mathcal{C} \text{ where } \mathcal{C} \text{ is a set of CPs}$$

$$l(\mathcal{A}_2) : \text{length of } \mathcal{A} = a'_1, a'_2, \dots, a'_{l(\mathcal{A}_2)}, a' \in \mathcal{C}$$

$$d_{max} : \text{maximum distance among all CPs}$$

$$\epsilon_b = \frac{\sum_i^{\min(l(\mathcal{A}_1), l(\mathcal{A}_2))} E}{\min(l(\mathcal{A}_1), l(\mathcal{A}_2))} \quad (1)$$

$$E = \sqrt{\sum_{j=1}^{\min(l(\mathcal{A}_1), l(\mathcal{A}_2))} (a_j - a'_j)^2}. \quad (2)$$

$$f_S = 1 - \frac{\epsilon_b}{d_{max}}. \quad (3)$$

If $f_S > \mu$, then $\mathcal{A}_1 \approx \mathcal{A}_2$. Parameter μ is determined by the user. In this work, our Webots simulated robot contains 10% slip noise at its wheels, we simply set $\mu=0.9$. Note that the inversed of \mathcal{A}_2 , namely \mathcal{A}_2^{-1} , must also be compared with \mathcal{A}_1 according to the assumption stated at the beginning of this subsection. Fig. 2(a) illustrates this. Two areas, \mathcal{A}_i and \mathcal{A}_j , are detected as similar. If $\mathcal{A}_i \approx \mathcal{A}_j$, then the connecting is performed in the left branch. If $\mathcal{A}_i \approx \mathcal{A}_j^{-1}$, then the connecting is performed in the right branch. Fig.2 (a) illustrates how to make the connection by assuming that the path shown in Fig.2 (b) goes into the case of $\mathcal{A}_i \approx \mathcal{A}_j$. Consequently, \mathcal{A}_{j-1} and \mathcal{A}_{j+1} would be connected to $junc(\mathcal{A}_i^-)$ and $junc(\mathcal{A}_i^+)$ respectively. Regarding to $junc(\mathcal{A}_i^-)$, Based on the derivation method described above, four transitions initially existed; namely, $\Delta_{\mathcal{A}_{j-1} \rightarrow \mathcal{A}_i}$, $\Delta_{\mathcal{A}_{j-1} \rightarrow \mathcal{A}_i}^{-1}$, $\Delta_{\mathcal{A}_{i-1} \rightarrow \mathcal{A}_i}$, $\Delta_{\mathcal{A}_{i-1} \rightarrow \mathcal{A}_i}^{-1}$. By reasoning, two new transitions are generated: $\Delta_{\mathcal{A}_{j-1} \rightarrow \mathcal{A}_{i-1}}$ and $\Delta_{\mathcal{A}_{j-1} \rightarrow \mathcal{A}_{i-1}}^{-1}$. As for $junc(\mathcal{A}_i^+)$, the connection is done mostly in the same way of $junc(\mathcal{A}_i^-)$ as shown in Fig. 2(b). The robot repeats checking the similar A-Pattern and fills the new areas to the junction. Then, the robot considers the next pattern $\mathcal{A}_{(i+1)}$ and fills the junction

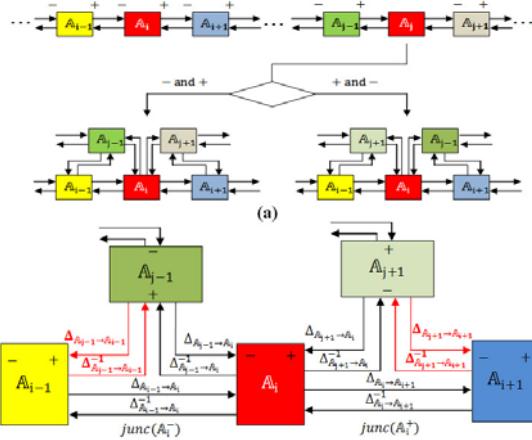


Fig. 2. (a) The A-Pattern \mathcal{A}_j is considered to be similar to \mathcal{A}_i either in non-inversed or inverted. The left branch is for non-invert similarity and the right branch is for invert similarity. (b) The illustration transitions derivation.

$junc(\mathcal{A}_{i+1}^+)$ and $junc(\mathcal{A}_{i+1}^-)$. This process is repeated until no similar patterns left. At the time the map is obtained, it may be complete or incomplete. The complete map is the map with all areas disclosed, and vice versa. If the map is complete, then the shortest path could be found. Otherwise, the path might not be the shortest one if the robot reached the goal by using the longer path. To solve this problem, we let the robot perform the additional survey during its traversal back to the start point. In the real world, the navigating cannot be divided by the episodes; namely, after the goal has been found, the robot still needs to traverse back to the start point by itself. During the way back, the robot explores the map more to see if there would be another shorter path. Particularly, assume that the robot finds the goal at time step t and that map \mathcal{M}_t is derived, given n as the number of junctions $\mathcal{J} = J_1, J_2, \dots, J_n$ corresponding to $\mathcal{M}_t = \mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{n+1}$, and \mathcal{A}_i as the goal area where $i \in I^+, i \lesssim n$, the robot tries to enter another different area \mathcal{A}'_i at junction J_i . As long as the length of the new path is less than the length of the old path, the robot continues moving along the new path. It is worth to note that this PBR optimization requires the area to be distinctive and asymmetrical. However, these constraints may be reasonable if the visual data is available.

3 Experimental Results

Evaluation is done using both Webots Simulator [7] and real e-puck robot. Only distance sensors are activated. No Vision, GPS, or compass is available. There are two main stages for both simulated and real experiment: Learning and Living. In learning, (i) the robot is given a learning environment (Fig.3 (a) and Fig.4 (c)) to self-organize and generate a set of common-patterns. The result of generated CPs of the real-experiment is shown in Fig.4 (a). In the living, (ii) the robot uses

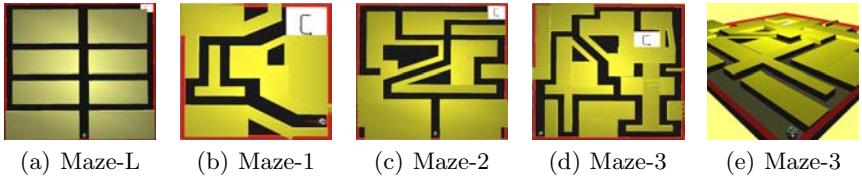


Fig. 3. Mazes simulated by Webots. (a) A maze for CPs generation. (b-d) An unfamiliar maze used for testing. (e) Side view of typical simulated maze-3.

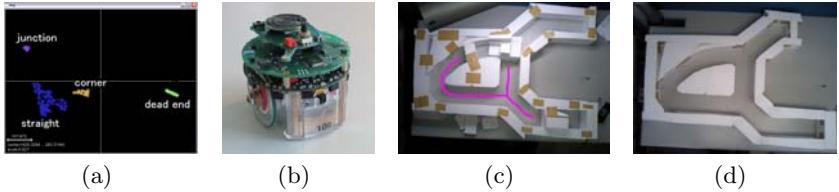


Fig. 4. (a) Depiction of CP generation using real e-puck. (b) Photograph of an actual e-puck robot. (c) The environment for learning CPs. Pink lines shows the available area. (d) The testing environment.

the generated CP to represent each observation point in every area, and finally generate the map of the unfamiliar environment. For the simulated experiments, the robot generates the CPs from Maze-L (Fig.3 (a)) and uses them to navigate in Maze-1, Maze-2 and Maze-3 (Fig.3 (b-d)). For the real experiment, the robot generates the CPs from the maze shown in Fig.4 (c), and uses them to navigate in maze of Fig.4 (d). The simulation results are shown in Fig.5. The robot is given one episode to explore the map while seeking the goal, and then take the shortest path to reach the goal again in the second episode. In Fig.5, panel (a) shows the required steps of the path in the first and second episode, while panel (b) shows the result in the aspect of time (s). For the result of real experiment, the robot uses 121 s to search for the goal and 44 s to reach the goal again with the shortest path. (63.63% time reduction). The simulation results prove that our method can incrementally build the map and seek the goal in one episode, while the real robot result prove that the proposed method could be successfully implemented on the real robot.

We discuss the advantages of the proposed architecture in two aspects: Pattern-Based Reasoning and Map-Building Approach. For the first aspect (i), Fig.1 clearly shows the difference between our architecture and that of [10]. Our proposed architecture is based on the Neural-Common Patterns which could be transferred to use in other environments. The rule (Conditional-Sequential) is not specific to any particular environment. The robot can use the last layer to remember the map of many environments at the same time as the areas are formed by connecting the CPs. For the second aspect (ii), our map-building differs from those of current methods in many ways. First, the common-patterns do not contain any position or orientation, so that the loop closing is done based on the

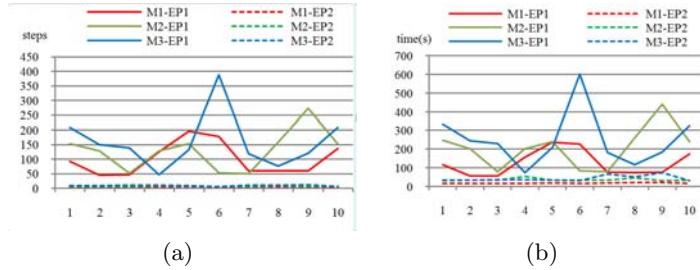


Fig. 5. Results of the mazes with simulated Webots. The x-Axis for all graphs shows the trial index. (a) The number of steps required for reaching the goal in the first (EP1) and the second (EP2) episodes corresponding to each maze. (b) The result in the aspect of time (s). Mi-EPj denotes the episode jth of Maze-i.

comparison between patterns. This allows our method to free from suffering from the error in self-pose estimation. Second, comparing to the current hybrid-map or topological map, our map take most of the sensory data into consideration, resulting in containing more information (most hybrid map lay emphasis on node while ignoring the edge [2], [6]). Comparing to the reinforcement learning [1], [5], our method requires only one episode to find the shortest path because the decision making is done reasonably.

References

1. Strosslin, T., et al.: Robust Self-Localization and Navigation Based on Hippocampal Place Cells. *Neural Networks* 18(9), 1125–1140 (2005)
2. Blanco, J., et al.: Toward a Unified Bayesian Approach to Hybrid Metric-Topological SLAM. *IEEE Transactions on Robotics* 24(2), 259–270 (2008)
3. Brooks, R.: A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation* RA-2(1), 14–23 (1986)
4. Choset, H., Nagatani, K.: Topological Simultaneous Localization without Explicit Localization. *IEEE Transactions on Robotics and Automation* 17(2), 125–137 (2001)
5. Grudic, G., et al.: Using Policy Gradient Reinforcement Learning on Autonomous Robot Controllers. In: *IEEE Int. Conf. on Intelligent Robots and Systems* (2003)
6. Kuipers, B., et al.: Local Metrical and Global Topological Maps in the Bybrid Spatial Semantic Hierarchy. In: *IEEE Int. Conf. on Robotics and Automation* (2004)
7. Michel, O.: WebotsTM: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems* 1(1), 39–42 (2004)
8. Sutton, R.S., Barto, A.G.: *Reinforcement Learning*. MIT Press, Cambridge (1998)
9. Shen, F., Hasegawa, O.: An Incremental Network for Online Unsupervised Classification and Topology Learning. *Neural Networks* 19(1), 90–106 (2005)
10. Sudo, A., et al.: Neural Pattern-Based Reasoning System for Online Learning. In: *14th Int. Conf. on Neural Information Processing* (2007)
11. Thrun, S., Burgard, W., Fox, D.: *Probabilistic Robotics*. MIT Press, Cambridge (2005)

Identifying Emotions Using Topographic Conditioning Maps

Athanasiou Pavlou and Matthew Casey

Department of Computing, University of Surrey,
Guildford, Surrey, GU2 7XH, UK
{a.pavlou,m.casey}@surrey.ac.uk
<http://www.cs.surrey.ac.uk>

Abstract. The amygdala is the neural structure that acts as an evaluator of potentially threatening stimuli. We present a biologically plausible model of the visual fear conditioning pathways leading to the amygdala, using a topographic conditioning map (TCM). To evaluate the model, we first use abstract stimuli to understand its ability to form topographic representations, and subsequently to condition on arbitrary stimuli. We then present results on facial emotion recognition using the sub-cortical pathway of the model. Compared to other emotion classification approaches, our model performs well, but does not have the need to pre-specify features. This generic ability to organise visual stimuli is enhanced through conditioning, which also improves classification performance. Our approach demonstrates that a biologically motivated model can be applied to real-world tasks, while allowing us to explore biological hypotheses.

1 Introduction

Emotions are an essential survival tool. The amygdala is the critical neural structure that acts as an evaluator of potentially threatening stimuli, priming our bodies for action [1]. As such, the amygdala is an attractive area to study because simulating such emotions may be an effective way for an artificial system to interact with humans, adjusting its responses to different events. One way in which such emotions can be modelled is through studying the elicitation mechanisms of the brain at the anatomical and behavioural levels. Fear was the first emotion that allowed such studies to be conducted, because of its evolutionary significance for survival. In addition, fear is easy to elicit or even artificially create (classical conditioning) [2]. This occurs when repeatedly pairing a neutral stimulus, the conditioned stimulus (CS), with an unconditioned stimulus (US), typically a loud burst of noise or electric shock [3]. The neural pathways of this process have been extensively studied for the auditory modality of rats [3]. However, several aspects such as their role for processing a wider spectrum of emotions [4] as well as anatomical interconnectivity of participating structures [5] are still under investigation for the visual modality. In particular, a dual neural pathway has been identified through which visual stimuli flow through thalamic areas and feed directly (from the lateral posterior nucleus) and indirectly (through the lateral geniculate nucleus and visual cortices) to the amygdala [1].

This study presents a model of visual fear conditioning that explores the dual cortical and sub-cortical visual pathways leading to the amygdala (Fig. II). This model captures the basic properties of the participating structures, and is the first that can condition on complex visual stimuli by extending the work of Armony et al [6] to the visual domain. We first evaluate the capability of our extended algorithm on processing abstract stimuli, and then we test the efficiency of the model's sub-cortical pathway to categorize the emotions expressed in face images. By these enhancements we aim to examine the degree to which a biologically constrained architecture can tackle real-world problems and gain insight into how such emotional detection takes place at early stages of sensory processing in the brain. The architecture of our model enables us not only to use it as an emotion detector but also, by applying conditioning, we can embed significance evaluation (learning higher responses when a particular stimuli is presented). In contrast with other techniques (see [7] for a review on facial emotion detection) our architecture does not rely on any manual feature extraction or particularly imposed geometric relations, suggesting that it can be used on a generic class of visual inputs. In addition, we aim to provide a more detailed framework that can assist and verify results of current neurobiological experiments of this field.

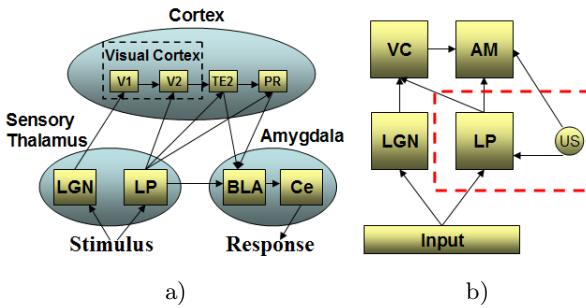


Fig. 1. Visual fear conditioning pathways II a) lateral geniculate nucleus (LGN), lateral posterior nucleus (LP), basolateral amygdala complex (BLA), central amygdaloid nucleus (Ce), primary (V1) and secondary (V2) visual cortices, temporal cortical areas (TE2), and perirhinal cortex (PR), b) model schematic with LGN, LP, early visual cortices (VC) and the amygdala (AM). The area within the dashed line denotes the part of the model used for the evaluation using real emotional expressions.

2 Method

We represent the anatomical structures participating in visual fear conditioning using a series of feedforward neural modules that are trained competitively. Armony et al [6] modelled auditory fear conditioning using one-dimensional modules. To successfully model visual pathways, the topographic relationship between regions within a single stimulus is crucial. Our work extends Armony et al's to have two-dimensional topographic maps. We introduce the notion of lateral inhibition using a neighbourhood as formulated by Kohonen [8]. Conditioning on these maps can then occur via an additional input (equivalent to

the US), which is active for the CS, affecting the plasticity of all neurons in the map. This combination of Hebbian learning in a topographic map is similar to the Laterally Interconnected Synergetically Self-Organizing Map (LISSOM) technique developed from studies on V1, which has successfully been used to model selectivity in the visual cortex [9]. Whereas such techniques offer more plausible models of receptive fields, our model builds on the key strength of Armony et al's approach in that it allows us to condition on arbitrary inputs, while still maintaining the ability to successfully model different layered brain structures at a sufficient level of detail (cf. [6]).

Fig. 1b shows a schematic of our model of visual fear conditioning. The cortical pathway is represented as connections from the LGN and LP to the VC, which then feed the AM. The sub-cortical pathway feeds the output of the LP directly to the AM. The visual stimulus is input to the LGN and LP and the US is input to the LP and AM, taking a value of 1 when conditioning, and 0 at all other times. This US always has a fixed weight value, which for us is 0.7. Each module consists of a lattice of neurons that are fully connected to the input, such that a neuron (i, j) has an output y corresponding to an m -dimensional input x :

$$u_{ij} = \sum_{k=1}^m x_k w_{kij}(t), \quad (1)$$

$$y_{ij} = \begin{cases} f(u_{ij}) & \text{if } \|c_{ij} - c_{win}\| < h(t) \\ f(u_{ij} - y_{win}) & \text{otherwise} \end{cases}, \quad (2)$$

$$f(u) = \begin{cases} 1 & u \geq 1 \\ u & 0 < u < 1 \\ 0 & u \leq 0 \end{cases}, \quad (3)$$

where $w_{kij}(t)$ is the weight from input k for neuron (i, j) in the lattice at time step $t \geq 0$, initialized with uniformly distributed small random values. Note in equation 2 that a neuron is considered to be in the winner area if the distance from the neuron (i, j) to the winner in the lattice is less than the current radius value $h(t)$. Here we use c_{ij} and c_{win} to denote the lattice co-ordinates of the two neurons. All neurons outside this area are inhibited by the activation value of the winning neuron $y_{win} = \max_{ij} f(u_{ij})$.

Competitive learning is achieved by updating each weight, except those fixed for the US, and then normalizing all weights to prevent exponential growth:

$$w'_{kij}(t+1) = w_{kij}(t) + \epsilon(t)x_k y_{ij}, \quad (4)$$

$$w_{kij}(t+1) = \frac{w'_{kij}(t+1)}{\sum_{l=1}^m w'_{lij}(t+1)}, \quad (5)$$

where $\epsilon(t)$ is the learning rate at time step t , corresponding to the presentation of a single input. This differs slightly from Armony et al's [6] formulation in that all weights are updated, not just those that have an input that is above average.

3 Experiments and Evaluation

The experiments presented in this section aim to evaluate the model's capabilities in terms of biological relevance and then application to more complex tasks. By the use of abstract stimuli we can obtain a clear insight into the model's ability to form topographic representations, as well as its parameter requirements. By then focusing on recognising emotions in faces, we can examine the model's potential for more challenging tasks, especially those already associated with the amygdala [4].

3.1 Experiments on Abstract Stimuli

The first phase of the evaluation aims to determine if the model can correctly form topographic maps. Here we use overlapping patterns of Gaussian activation in varying locations to represent visual stimuli. We use 266 patterns as the training examples corresponding to a series of spatial locations within an input representing azimuth [-90, 90] and elevation [-65, 65] (similar to a human's visual field). Within this, we allow the positioning of an object at a discrete interval of 10, so that we can encode 19 different positions for azimuth and 14 for elevation. For a stimulus at azimuth p and elevation q , we have an input x as:

$$x_{pq} = \lambda e^{-(\frac{p^2-q^2}{\sigma^2})}, \quad (6)$$

where λ is the maximum amplitude and σ the radius, chosen as $\lambda = 1$ and $\sigma = 10$ for these experiments.

Training takes place by layers so that the input feeds into the LGN and LP first. After training is finished these modules' outputs are used to train the VC. In turn, when the VC has finished training, then its combined outputs with the LP train the AM in a similar fashion (Fig. IIb). The LGN, LP and VC are represented by a 10 by 10, and the AM by a 5 by 5 neuron lattice. The smaller size of the AM reflects biology [1]. The chosen sizes of the maps provide detailed representation of the inputs while remaining computationally efficient. All 266 inputs were presented during each epoch in uniformly random order. A decreasing per epoch Gaussian neighbourhood radius function and an exponential learning rate function were used:

$$h(t) = r_{min} + (r_{max} - r_{min})e^{-(\frac{(t/t_e)^2}{2r_s^2})}, \quad (7)$$

$$\epsilon(t) = l_{min} + (l_{max} - l_{min})e^{-(\frac{(t/t_e)^2}{2l_s^2})}, \quad (8)$$

where r_{min} and r_{max} are the minimum and maximum radius for the neighbourhood and r_s is the bandwidth; similarly, l_{min} , l_{max} and l_s for the learning rate. The values only vary per epoch, thus t_e defines the number of time steps per epoch (266). Through trial and error the chosen parameter values produced stabilized map organization over the training period. All modules used the same set of parameters differing on the minimum neighbourhood radius. Here, r_{max}

was equal to the lattice width (10 or 5), r_{min} was 2 for the LGN, 3 for the LP, 1 for the VC and 4 for the AM corresponding to the differing visual coarseness in each module, while $r_s = 300$, $l_{max} = 0.1$, $l_{min} = 0.001$ and $l_s = 13$. A stable topographic organization was achieved after 700 epochs of training (Fig. 2).

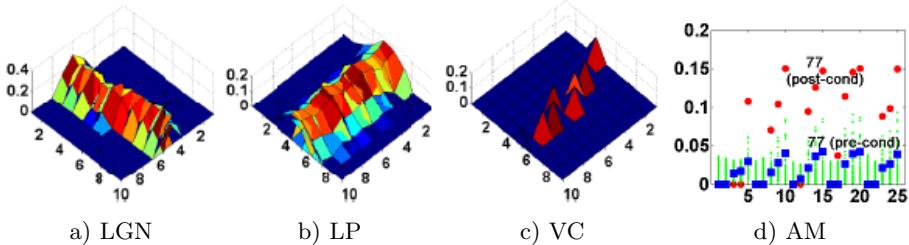


Fig. 2. Topographic organization using stimuli with azimuth 0 and elevation in the range [-65 65] for the a) LGN ($r_{min} = 2$), b) LP ($r_{min} = 3$), and c) VC ($r_{min} = 1$). Observe that the LGN and LP have different topographic orientations for the same locations, but these have successfully been aligned in the VC. Neuron activations in the AM pre- and post-conditioning (squares, circles) are shown in d) in response to CS 77, together with responses from the rest of the inputs (dots).

For the second phase we evaluate the effect of conditioning on the maps, selecting a single stimulus as the CS. Training differs from the pre-conditioning phase in that now all the layers are trained concurrently. We note that the map radii have already reached their minimum values, however the learning rate continues to drop after each epoch (continuing from the value it had on the last epoch of pre-conditioning). This time training occurred for 530 epochs until the map activation patterns were again stable.

The results from the model’s pre-conditioned phase indicate successful map organization with different specificity of input representation on each map depending on the minimum neighbourhood radii used (more specific representation of inputs on the VC compared to the rest). A large radius for the AM was selected to ensure that no topographic organization will occur since the amygdala is not known to have such capability. As we see from Fig. 2 the post-conditioning activations of stimulus 77 on the AM have significantly increased compared to their pre-conditioning values, as have the activations corresponding to the locations surrounding 77. Overall, these results show both that the maps correctly organise the stimuli by similarity in the input (location), while allowing us to condition a particular output to gain a higher activation, as per Armony et al’s results on one-dimensional modules [6].

3.2 Experiments on Emotional Expressions in Face Images

Having established the model’s capability of handling abstract two dimensional input we proceed to examine its efficiency on the real-world task of emotion recognition using real face images taken from the MMI Facial Expression

Database collected by Pantic & Valstar [10]. The experiments are restricted to the sub-cortical pathway (LP module) since this is adequate for observing map organisation as well as conditioning effects, and allows us to explore the efficacy of this ‘coarse’ visual processing pathway computationally.

The same method of training (pre- and post-conditioning phases) was followed as before. This trial differs from the previous experiment in that the map size is now scaled up to a 32 by 32 lattice to correspond with an increased input size, while all other parameters remain the same. For training we use 598 frames of a single person (training subject) taken from the MMI database. The frames are taken from video files containing transitions from a neutral state to a gradually increasing emotional expression (anger, disgust, fear, happiness, sadness and surprise) that reaches a peak (peak frame) and from that returning to the original (neutral) expression. Fig. 3 shows the map activations for the training data for the peak frame of each emotion. The only pre-processing of the images was cropping to have approximate upper and lower boundaries the eyebrows and chin, resizing this to 32 by 32 pixels and finally grey-scaling them. For testing we used frames from videos of two other subjects from the MMI database. From the first subject (S002) we acquired frames depicting gradations of anger, happiness and sadness whilst from the second subject (S031) frames depicting gradations of disgust, fear and surprise. In total we used 121 testing frames taking 21 consecutive frames for each emotion that include the peak emotion frame.

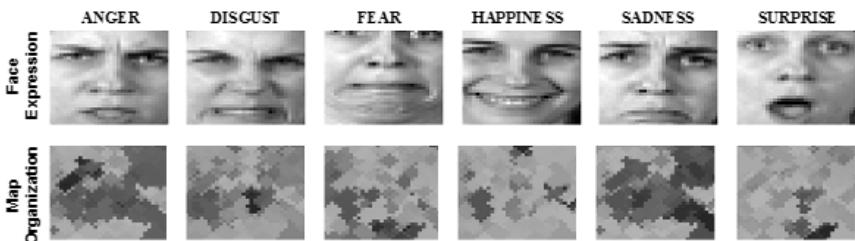


Fig. 3. Example trianing inputs for each emotion (peak video frame) and corresponding map activations (pre-conditioning). Higher activations are denoted by darker shading. Note that the activated areas are distinct and differ for each emotion.

To label the map on the training data, for each peak frame for the six emotions we recorded each neuron’s activation. Since each emotion produced a cluster of highly active neurons (Fig. 3), we labelled progressively larger clusters to determine the ability of the map to have the same pattern of activity during testing: the top 0% (just the most active neuron), 10%, 20% and 30%. To obtain a class for a test image, we determined the number of neurons that overlapped in these areas to those that had been labelled for each emotion, picking the label with the majority of overlap. The top 20% of active neurons gave the best results, as shown in Table 1. Classification accuracy is 76% or more for fear, happiness, sadness and surprise, whereas anger and disgust are misclassified (as sadness, and as anger, disgust and sadness, respectively), with an overall accuracy of 62%.

Table 1. Confusion matrix of the number of successfully classified frames for the top 20% of active neurons, shown for pre- and post-conditioning (on the peak frame of the angry face). The training peak frames were compared to 21 testing frames per emotion. Unclassified patterns are shown as NK.

	Pre-conditioning							Post-conditioning								
	An	Di	Fe	Ha	Sa	Su	NK	Accuracy	An	Di	Fe	Ha	Sa	Su	NK	Accuracy
Anger (S002)	0	0	0	0	21	0	0	0%	21	0	0	0	0	0	0	100%
Disgust (S031)	3	3	0	0	12	0	3	14%	3	4	0	0	8	0	6	19%
Fear (S031)	0	0	16	0	0	0	5	76%	0	0	21	0	0	0	0	100%
Happiness (S002)	0	0	0	19	2	0	0	90%	0	0	0	21	0	0	0	100%
Sadness (S002)	0	0	0	21	0	0	0	100%	0	0	0	0	21	0	0	100%
Surprise (S031)	0	0	2	0	0	19	0	90%	0	0	12	0	0	9	0	43%

This performance can be attributed to the similarity of the training peak frames for anger, disgust and sadness (Fig. 3). While these results do not approach the performance of feature-based techniques (overall 89% [1]), they at least show that the model is capable of distinguishing between facial expressions. Our biologically motivated question is whether conditioning on an arbitrary stimulus can enhance this recognition rate?

To understand this, we chose to condition the model on the peak frame of the anger examples (CS 55). After conditioning, the map activations both of training and testing subjects for anger were increased, while activation levels for the rest of the emotions remained similar. This matches the results seen on the abstract stimuli and shows that the model is capable of behaving in the same way on more complex input. Further, conditioning affected classification performance (Table II), with anger and disgust improved (100% and 19%), while the performance for surprise decreased (43%). This follows from the localised increase in activations in the map, where anger, disgust and surprise are similarly located. Conditioning provokes an increased level of activation from the neurons most active for the CS, and because of the neighbourhood, a similar increase in surrounding neurons. As a result, the CS activations are separated from those that they were previously close. Overall the model achieved an accuracy of 77%. This is comparable with feature-based techniques such as [1] which for the emotions of happiness (joy), surprise, disgust, anger, sadness and fear achieved 100%, 63%, 100%, 89%, 95% and 89% respectively, overall 89%, albeit using a larger test sample with 62 subjects (1500 frames). Here, we show that a non-feature based technique can gain comparable results just through a process of conditioning on topographic maps, motivated from biology and without any hard-coded feature extraction.

4 Conclusion

In this paper, we presented a topographic conditioning map (TCM) model of the cortical and sub-cortical visual pathways leading to the amygdala. We evaluated

the model on abstract stimuli to determine whether it could learn topographic relationships, and how conditioning on an arbitrary input affected the responses. Our motivation is whether taking inspiration from biology can help both neuroscience through developing an understanding of the different pathways computationally, while providing new techniques to tackle complex problems, particularly those associated with vision which offer significant challenges. While in this paper we have not directly addressed the former except to provide a model with which we can explore neuroscientific hypotheses in the future, for the latter we have demonstrated that the TCM is capable of successfully recognising emotional facial expressions. Our model differs from other work in this area [7][11] in that none of the features of each emotional class were chosen manually. Rather, the model developed distinct activation clusters for each of the six emotions it was trained on during pre-conditioning. Conditioning the model then enhanced the map activation levels to improve classification performance for a selected emotion. Future work involves testing on a larger number of stimuli, and to extend the model to tackle the important challenge of object invariance (cf. [12]).

References

1. Morris, J.S., Öhman, A., Dolan, R.J.: A subcortical pathway to the right amygdala mediating “unseen” fear. *Proceedings of the National Academy of Sciences* 96, 1680–1685 (1999)
2. Pavlov, I.P.: *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. Oxford University Press, London (1927)
3. LeDoux, J.E.: Emotion, memory and the brain. *Scientific American Special Edition* 12(1), 62–71 (2002)
4. Murrey, E.A.: The amygdala, reward and emotion. *Trends in Cognitive Sciences* 11(11), 489–497 (2007)
5. Pessoa, L.: To what extent are emotional visual stimuli processed without attention and awareness. *Current Opinion in Neurobiology* 15, 188–196 (2005)
6. Armony, J.L., Servan-Schreiber, D., Romanski, L.M., Cohen, J.D., LeDoux, J.E.: Stimulus generalization of fear responses: Effects of auditory cortex lesions in a computational model and in rats. *Cerebral Cortex* 7(2), 157–165 (1997)
7. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J.: Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine* 18(1), 32–80 (2001)
8. Kohonen, T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 59–69 (1982)
9. Miikkulainen, R., Bednar, J.A., Choe, Y., Sirosh, J.: *Computational Maps in the Visual Cortex*. Springer Science+Business Media, New York (2005)
10. Pantic, M., Valstar, M., Rademaker, R., Maat, L.: Web-based database for facial expression analysis. In: *IEEE International Conference on Multimedia and Expo. (ICME 2005)* (July 2005)
11. Hupont, I., Cerezo, E., Baldassarri, S.: Facial emotional classifier for natural interaction. *Electronic Letters on Computer Vision and Image Analysis* 7(4), 1–12 (2008)
12. Stringer, S.M., Perry, G., Rolls, E.T., Proske, J.H.: Learning invariant object recognition in the visual system with continuous transformations. *Biological Cybernetics* 94, 128–142 (2006)

A Gene Regulatory Model for the Development of Primitive Nervous Systems

Yaochu Jin¹, Lisa Schramm², and Bernhard Sendhoff¹

¹ Honda Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach, Germany

² Technische Universität Darmstadt, Karolinenplatz 5, 64289 Darmstadt, Germany

Abstract. This paper presents a model for the development of primitive nervous systems in a hydra-like animat controlled by a gene regulatory network. The gene regulatory network consists of structural genes that simulate the main cellular events during neural development at an abstract level, namely, cell division, cell migration, and axon growth, and regulatory genes that control the expression of the structural genes. The developmental model is evolved with an evolutionary algorithm to achieve the correct developmental order. After the genetically controlled neural development is completed, the connectivity and weights of the neural networks are further adapted to allow the animat for performing simple behaviors such as the food catching behavior of a hydra. Our preliminary results suggest that the proposed developmental model is promising for computational simulation of the evolution of neural development for understanding neural organization in biological organisms.

1 Introduction

Understanding the evolution of biological neural organization using computational models has attracted increasing attention. Two issues are considered essential for this body of research to be biologically plausible. First, the evolution of neural organization should be coupled with that of the body plan [14,7]. Second, the influence of neural development on the evolution of the nervous system should also be taken into account [9].

A few computational models for neural development have been reported. Can-gelosi et al. [2] suggested a developmental model for cell division and migration that uses a rule-rewriting grammar. A model for neurogenesis that includes metabolic reactions and diffusion was proposed by Kitano [8], though no functionality of the developed neural network has been considered in the paper. A recurrent artificial neural network was used for modeling the development of a spiking neural network for the control of a Khepera robot [5].

This work suggests a computational model for neural development based on a gene regulatory network (GRN) for morphological development [13]. The gene regulatory network is composed of a number of structural genes whose expression is controlled by regulatory genes. The gene regulatory model is evolved to achieve the correct developmental order, i.e., first cell division, then cell migration, and finally axon growth [10]. The neurons are distributed over the surface

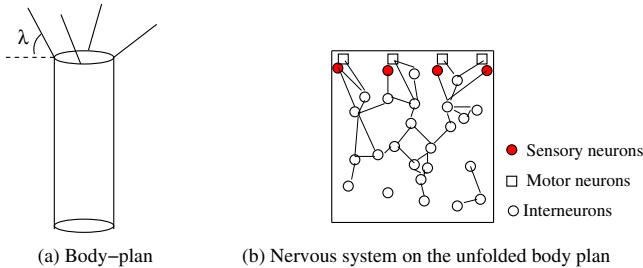


Fig. 1. The hydra-like animat. (a) The body plan with four tentacles, and (b) the nervous system shown on the 2-D area representing the unfolded body surface.

of a cylinder that simulates the body of hydra, which are phylogenetically the first to have a nervous system. Neurons are then connected according to a probability inversely proportional to the distance between them. In order to perform a food-catching behavior, the connectivity and weights of developed neural system are further adapted. Part of this work has been reported in [1].

Section 2 describes the body plan of the hydra-like animat, the gene regulatory model for neural development, as well as the neural network dynamics based on the integrate and fire spiking neural model. The experimental setup and simulation results for evolving the developmental order are presented in Section 3. The developed neural system is further adapted to catch as many pieces of food as possible while minimizing energy consumption, which is presented in Section 4. The paper concludes with discussions in Section 5.

2 The Model

2.1 A Hydra-Like Animat

The physical body of the hydra-like animat simulates very loosely that of a hydra, which consists of a cylinder with four tentacles equally distributed on its top. Each tentacle is driven by a muscle cell controlled by a neural network composed of motor neurons, interneurons and sensory neurons, all of which are modeled with integrate and fire (IAF) neurons, refer to Fig. II. To simplify the experimental setup, four sensory neurons and four motor neurons are put on the top of the body, while the number and position of the interneurons are determined by the developmental process. During the behavior adaptation, four pieces of food drop sequentially around the body and the closest sensory neuron will generate a number of spikes as the input to the neural network.

2.2 Genome for Neural Development

The genome in our model consists of structural units (SUs) representing genes that can result in certain cellular behaviors, such as cell division, cell migration,

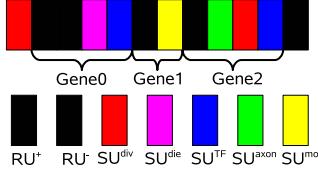


Fig. 2. The example of the genome for neural development

axon growth, and cell death, or can produce transcription factors (TFs) for intra- and inter-cellular interactions. The SUs are preceded by a number of regulatory units (RUs), refer to Fig. 2. Note that multiple RUs may regulate the expression of a single or multiple SUs and that RUs can be activating (RU^+) or repressive (RU^-). Three different cell types are simulated in the model: stem cells, interstitial cells (after division) and neurons (after axon growth), depending on the activation status of the structural units of the cell.

Each RU and TF has a certain affinity value that determines whether a TF can regulate a RU. If the difference between the affinity values of a TF and a RU is smaller than a predefined threshold ϵ (in this work ϵ is set to 0.2), the TF can be bound to the RU to regulate. The affinity overlap ($\gamma_{i,j}$) between the i -th TF and j -th RU is defined by:

$$\gamma_{i,j} = \max(\epsilon - |\text{aff}_i^{\text{TF}} - \text{aff}_j^{\text{RU}}|, 0). \quad (1)$$

If $\gamma_{i,j}$ is greater than zero, then the concentration c_i of the i -th TF is checked to see if it is above a threshold ϑ_j defined in the j -th RU:

$$b_{i,j} = \begin{cases} \max(c_i - \vartheta_j, 0) & \text{if } \gamma_{i,j} > 0 \\ 0 & \text{else} \end{cases} \quad (2)$$

Thus, the activation level contributed by the j -th RU (denoted by $a_j, j = 1, \dots, N$) can be calculated as follows:

$$a_j = \sum_{i=1}^M b_{i,j}, \quad (3)$$

where M is the number of TFs that regulate the j -th RU. Assume the k -th structural gene is regulated by N RUs, the expression level of the gene can be defined by

$$\alpha_k = 100 \sum_{j=1}^N h_j a_j (2s_j - 1), \quad (4)$$

where $s_j \in (0, 1)$ denotes the strength and sign (positive for activating and negative for repressive) of the j -th RU, which is subject to evolution, and h_j is an additional parameter representing the regulatory strength of j -th TF. If $\alpha_k > 0$, then the k -th gene is activated and its corresponding behaviors are performed.

A SU that produces a TF encodes all parameters related to the TF, such as the affinity value, the decay rate, the diffusion rate, as well as the amount of TF to be produced:

$$A = \beta \frac{2}{1 + e^{-20 \cdot f \cdot \alpha}} - 1, \quad (5)$$

where f is a parameter also encoded in the SU, and β is a parameter for expression rate of the SU.

A TF produced by a SU can be internal (for inter-cellular interactions) and external (intra-cellular interactions). Internal TFs stay inside a cell and do not diffuse. External TFs can diffuse into the whole simulation area, a process that depends on the diffusion parameter of the TF. To determine how much of a produced TF is external, a percentage ($p^{\text{ex}} \in (0, 1)$) is also encoded in the corresponding gene. Thus, $p^{\text{ex}} \cdot A$ is the amount of external TF and $(1 - p^{\text{ex}}) \cdot A$ is that of the internal TF.

External TFs are put on four grid points around the center of the cell, which undergoes first a diffusion and then decay process:

$$\text{Diffusion: } \mathbf{u}_i(t) = \mathbf{u}_i(t-1) + 0.1 \cdot D_i^f \cdot (\mathbf{G} \cdot \mathbf{u}_i(t-1)), \quad (6)$$

$$\text{Decay: } \mathbf{u}_i(t) = \min((1 - 0.1 \cdot D_i^c) \mathbf{u}_i(t), 1), \quad (7)$$

where \mathbf{u}_i is a vector of the concentrations of the i -th TF at all grid points, matrix \mathbf{G} defines which grid points are adjoining, D_i^f is the diffusion parameter, and D_i^c is the decay rate.

There are also a few parameters defined by the SUs that generate cellular behaviors. The SU for cell division specifies the angle of division, indicating whether the daughter cell is placed above or below the mother cell. For cell migration, two parameters are encoded, one for direction (moving up or down) and the other for moving velocity. The SU for axon growth encodes the expected lifetime (t_{ttd}) of a neuron and three parameters (c_1, c_2, c_3) determining the probability threshold for axon growth. The limited lifetime of the neuron, t_{life} , is defined by the following Gaussian distribution:

$$t_{\text{life}} \sim \mathcal{N}(20t_{\text{ttd}}, 4). \quad (8)$$

The threshold for whether the i -th neuron is to be connected to the j -th neuron is calculated as follows:

$$\varphi_{ij} = \frac{c_1}{1 + e^{c_2 \cdot (d_{ij} - 10c_3)}}, \quad (9)$$

where d_{ij} is the distance between the i -th and j -th neurons. The distance is computed in the 2D region (refer to Fig. II(b)), which is the distance between the two neurons along the surface of the cylindrical body. Then, a random number p ($p \sim \mathcal{N}(0, 1)$) is generated, and if $p < \varphi_{ij}$, a connection between the two neurons will be generated. The connectivity between the interneurons and the sensory neurons, as well as between the interneurons and motor neurons is determined in the same way. Note, however, that there is no direct sensory-sensory, motor-motor, and sensory-motor connection.

2.3 Neural Network Dynamics

The dynamics of the neural network is modeled by IAF spiking neurons [3]:

$$\tau_m \frac{dV(t)}{dt} = -(V(t) - V_{\text{resting}}) + R_m I(t), \quad (10)$$

where τ_m is the membrane time constant, R_m is the membrane resistance, $I(t)$ is the external current, $V(t)$ is the membrane potential, V_{resting} is the resting potential. When the membrane potential $V(t)$ is larger than a given threshold V_{th} , a spike is generated. The spiking event is characterized by its firing time t^f . After spiking, the membrane potential is reset to V_{reset} for a refractory period τ_{ref} . In an IAF neural network, the membrane potential of a neuron (V_k) can be calculated by:

$$V_k(t) = \sum_{i=1}^H w_{ik} y_i(t), \quad (11)$$

where w_{ik} is the weight between neurons k and i , N is the total number of its presynaptic neurons, and $y_i(t)$ is the unweighted contribution of the i -th presynaptic neuron:

$$y_i(t) = \varepsilon(t - t_i^f - d_{ik}), \quad (12)$$

where ε is a spike response function modeling the post-synaptic potential, t_i^f is the firing time of neuron i , and d_{ik} is the synaptic delay.

The spiking network is simulated using the Neural Simulation Tool [6].

3 Evolving the Developmental Order

An evolutionary algorithm is applied to evolve the genome to achieve the correct developmental order, i.e., the genes for cell division, cell migration and axon growth should be activated sequentially during the development. In addition, the gene for division should be deactivated before the gene for migration is activated, as the gene for migration should be deactivated before the gene for axon growth is activated. Gene transposition, gene duplication, and mutations are employed as the genetic variations. If gene transposition is performed, two randomly chosen units (both SUs and RUs are possible) are marked, and all units between these two marked units are cut out and pasted at another randomly chosen position. A gene duplication differs with gene transposition in that all units between the markers are copied and pasted at another randomly chosen position. All real-valued parameters (scaled between [0,1]) in the genome are adapted using an evolution strategy (ES) [12]. A repairing operator is introduced so that a single gene contains only one type of SU related to cellular behaviors.

The following fitness function is used to penalize wrong developmental orders:

- If two cellular behaviors are active simultaneously, a penalty of 2 is applied.
- If three cellular behaviors are active simultaneously, a penalty of 5 is applied.
- If axon growth is followed by cell division, 2 is added.
- If axon growth is followed by cell movement, 1 is added.
- If cell migration is followed by cell division, 1 is added.

The target of the evolution is to minimize the fitness function. In the simulations, the parent and offspring population sizes are set to 1000 and 3000, respectively. The probability for gene transposition and duplication is set to 0.05, and the mutation step-size is initialized to $1e^{-4}$, with its lower bound being set to $1e^{-6}$ during evolution. The development begins with a single stem cell put in the center of the simulation area.

The fitness profile is shown in Fig. 3(a). We can see that the correct developmental order is achieved after over 300 generations. The activation levels of the gene for cellular behaviors (refer to Eq. 4) are shown in Fig. 3(b). A value above zero indicates that corresponding gene is activated and repressed if it drops below or equal to zero.

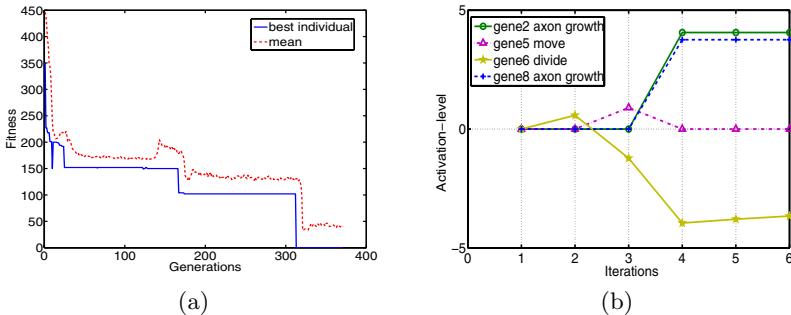


Fig. 3. (a) Fitness profile. (b) Activation levels of the genes.

4 Evolution of the Neural Network for Food-Catching

Twelve stem cells with the genome that performs the cellular behaviors in the correct order are distributed on the simulation area so that a neural network is developed. For the neural network to perform the food-catching behavior, we employ an ES to adapt its connectivity and weights. In this work, we assume that only one tentacle is needed to catch a piece of food. Thus, the target of the behavior adaptation is that the motor neuron closest to the dropping food should fire as strong as possible to maximize the possibility to catch the food, while the activity of other motor neurons should be minimal to reduce energy consumption.

To achieve the above-mentioned target, the following fitness function is defined for the i -th motor neuron (output neuron):

$$F_i = -f_i + \sum_{j \neq i} ns_j, \text{ for } i, j = 1, \dots, 4, \quad (13)$$

where ns_j is the number of spikes generated by other motor neurons in the simulated time period, penalizing unnecessary energy loss, f_i reflects the performance of the i -th neuron measured by the time (t_i^c) for the corresponding tentacle to stand vertically, i.e., when the angle $\lambda = 90$, refer to Fig. 1 or by the maximum λ during in the simulation period, if λ never reaches 90 degree during that time. The angle of the i -th tentacle is calculated as follows:

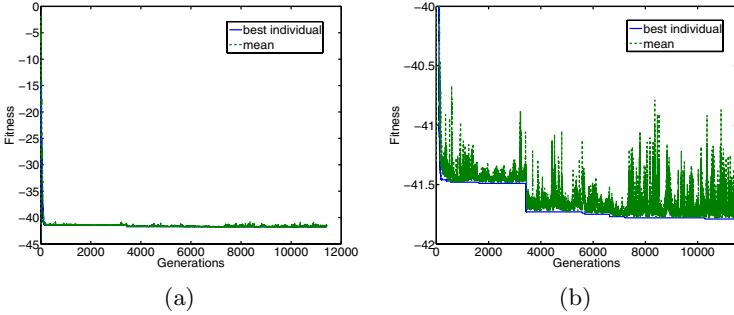


Fig. 4. (a) Fitness profile, and (b) fitness after zoomed in

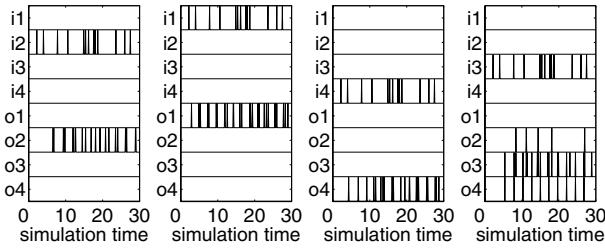


Fig. 5. Spikes at the input and output neurons for four different inputs

$$\lambda_i(t) = \lambda_i(t - 0.1) - g * 0.1 + s_i(t), \quad \lambda_i \in [0, 90]. \quad (14)$$

The second term on the right side of Eq. 14 simulates the passive dropping-down of tentacle due to gravity (g), $t \in [0, 30]$ is the simulation time in millisecond in behavior adaptation, $s_i(t)$ equals 0 or 1 depending on whether there is a spike at time t . The final fitness is summed over four runs with different food dropping conditions. We use a (30, 200)-ES and the initial mutation step-sizes are set to 0.1 with a lower bound of $1e^{-6}$. The maximal weight is set to 3000, which can be excitatory or inhibitory, and the synaptic delays are fixed to 1. All parameters are scaled between zero and one. The fitness curve for behavior optimization is shown in Fig. 4. The spikes of the input and output neurons of the best adapted individual is shown in Fig. 5. From the figure, we can see that in three of the four cases, the neural network performs optimally. In one case (last panel in Fig. 5), outputs 2 (o_2) and 4 (o_4) also fire, though only output 3 (o_3) is assumed to spike. Nevertheless, o_3 does fire the strongest.

5 Discussions and Conclusions

This paper suggests a neural developmental model based on a gene regulatory network. The GRN has evolved successfully to achieve the correct developmental

order, i.e., cell division, cell migration and axon growth in sequence in a simulated evolution that undergoes gene transposition, duplication and mutation. After the gene regulated neural development is complete, the connectivity and weights of the neural network are further adapted using an ES for performing a food-catching behavior in a hydra-like animat with success.

The current developmental model can be improved in various respects. Since no gene deletion is implemented in the model, the complexity of the genome tends to become more complex due to gene duplications. The cell migration behavior resulted from the current model is quite deterministic, as there is little local interactions between the cells. In the long run, a gene regulatory model for both neural and morphological development will also be investigated.

Acknowledgments

Authors are grateful to J. Eppler for his assistance in using NEST.

References

1. Dellaert, D., Beer, R.D.: Toward an evolvable model of development for autonomous agent synthesis. In: Artificial Life IV, pp. 246–257 (1994)
2. Cangelosi, A., Nolfi, S., Parisi, D.: Cell division and migration in a 'genotype' for neural networks. Networks - Computational in Neural Systems 5, 479–515 (1994)
3. Dayan, P., Abbott, L.F.: Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. MIT Press, Cambridge (2001)
4. Downing, K.L.: Supplementing evolutionary developmental systems with abstract models of neurogenesis. In: Genetic and Evolutionary Computation Conference, pp. 990–996 (2007)
5. Federici, D.: Evolving developing spiking neural networks. In: Congress on Evolutionary Computation, vol. 1, pp. 543–550 (2005)
6. Gewaltig, M.-O., Diesmann, M.: NEST (Neural Simulation Tool). Scholarpedia 2(4), 1430 (2007), <http://www.nest-initiative.org/>
7. Jones, B.H., Jin, Y., Sendhoff, B., Yao, X.: Evolving functional symmetry in a three dimensional model of an elongated organism. In: Artificial Life XI, pp. 305–312 (2008)
8. Kitano, H.: A simple model of neurogenesis and cell differentiation based on evolutionary large-scale chaos. Artificial Life 2(1), 79–99 (1995)
9. Psujek, S., Beer, R.D.: Developmental bias in evolution: Evolutionary accessibility of phenotypes in a model of evo-devo systems. Evolution and Development 10(3), 375–390 (2008)
10. Sanes, D.H., Reh, T., Harris, W.A.: Development of the Nervous System. Academic Press, Amsterdam (2006)
11. Schramm, L.: A model for nervous systems development controlled by a gene regulatory network. Diploma thesis, Technische Universität Darmstadt (2007)
12. Schwefel, H.-P.: Evolution and Optimum Search. John Wiley, Chichester (1994)
13. Steiner, T., Schramm, L., Jin, Y., Sendhoff, B.: Emergence of feedback in artificial gene regulatory networks. In: Congress on Evolutionary Computation, pp. 867–874. IEEE, Los Alamitos (2007)

Real-Time Epileptic Seizure Detection on Intra-cranial Rat Data Using Reservoir Computing

Pieter Buteneers, Benjamin Schrauwen, David Verstraeten,
and Dirk Stroobandt

Ghent University, Belgium

Abstract. In this paper it is shown that Reservoir Computing can be successfully applied to perform real-time detection of epileptic seizures in Electroencephalograms (EEGs). Absence and tonic-clonic seizures are detected on intracranial EEG coming from rats. This resulted in an area under the Receiver Operating Characteristics (ROC) curve of about 0.99 on the data that was used. For absences an average detection delay of 0.3s was noted, for tonic-clonic seizures this was 1.5s. Since it was possible to process 15h of data on an average computer in 14.5 minutes all conditions are met for a fast and reliable real-time detection system.

1 Introduction

Epilepsy is a neurological disorder of the brain where the patient is disturbed by mostly recurring seizures. Around 1% of the world's population suffers from this illness [1]. Although a cure for this disorder has not yet been found, medication is in most cases sufficient to block the seizures.

To determine whether the applied medication is working, a doctor needs to determine the remaining epileptic activity, and thus also the seizures, on hours of recorded EEG data. Especially in the case of absence epilepsy, where the number of seizures is a very good indication, an automatic detection system is highly desired. Additionally anti-epileptic drugs are known for their side-effects. To avoid these side-effects one could apply a closed-loop system, where the medication is applied in real-time when a seizure occurs. A third application of a fast and reliable detection system is as a warning system. The environment of the patient can then be alerted when a seizure occurs so they are able to help and protect the patient in this unpleasant and dangerous episode.

In this study we show how Reservoir Computing (RC) can be successfully applied to detect epileptic seizures on EEG in real-time. RC is a training method for recurrent neural networks where only a simple linear readout function is trained for a randomly created network or reservoir.

The seizures of two different types of generalized epilepsy are detected: absence epilepsy and tonic-clonic epilepsy. Spike Wave Discharges (SWDs) are the EEG patterns that occur when a patient with absence epilepsy is having a seizure. These absences last from several seconds to a few minutes and are generally very

well treatable with medication. Tonic-clonic seizures on the other hand are not as easily suppressed by medication. They last from about a minute to even more than 15 minutes and consist of two phases as can be distinguished in Fig. 3. In the tonic phase the muscles are strained followed by convulsions in the clonic phase. Although the data used for this study is intracranial rat data, it can be used as a model for human data.

In section 2 Reservoir Computing is explained. Section 3 describes the reservoir setup used to detect SWDs and evaluates the results by a comparison in detection delay and ROC curves with other methods. Section 4 briefly describes the similar setup used for tonic-clonic seizure detection and gives the test results on the data set used. In the last section a conclusion is drawn with a link to future work.

2 Reservoir Computing

Reservoir Computing (RC) [2] is a very recent training method for recurrent neural networks and is based on the Echo State Network (ESN) approach which was presented by H. Jaeger in 2001 [3]. RC, which has been shown to render very good results at speech recognition [4], greatly simplifies the training of recurrent neural networks. To generate an output, a linear readout function is attached to a randomly created recurrent neural network, which is called a reservoir, and only the weights of this function are trained. This way the known stability issues of RNNs are easily avoided as well as the long training time [5].

The RC concept is shown to work with several types of neurons: spiking neurons, analog neurons... For this study Leaky Integrator Neurons [6] are used. These are basic hyperbolic tangent neurons followed by a simple first order lowpass filter.

The operation of the reservoir can be described as follows: If we use $\mathbf{x}[k]$ to represent the current activation for each of the neurons in the reservoir we can calculate the next reservoir state $\mathbf{x}[k + 1]$ using the following equation:

$$\mathbf{x}[k + 1] = (1 - \gamma) \cdot \mathbf{x}[k] + \gamma \cdot \tanh(W_{res}^{res} \mathbf{x}[k] + W_{inp}^{res} \mathbf{u}[k] + W_{bias}^{res})$$

In this equation γ represents the leak rate and is used to set the cutoff frequency of the lowpass filter in the neurons, $\mathbf{u}[k]$ is the input vector and W_a^b represents the randomly generated weight vector from layer a to b . To generate the output $\hat{\mathbf{y}}[k]$ the following equation is used:

$$\hat{\mathbf{y}}[k] = W_{res}^{out} \mathbf{x}[k] + W_{bias}^{out}$$

Here $\hat{\mathbf{y}}[k]$ is used instead of $\mathbf{y}[k]$ because the latter is often used as the desired output. In these previous equations only W_{res}^{out} and W_{bias}^{out} are trained for which in this study a ridge regression based algorithm was used.

To render good results the reservoir itself is just slightly tweaked [5]. For W_{res}^{res} and W_{inp}^{res} the optimal connection fraction is determined. This factor states the percentage of weights that are not zero. The processing power of reservoirs is

greatest when it operates at the edge of stability [7]. Therefore the spectral radius (the largest absolute eigenvalue) of W_{res}^{res} is scaled together with W_{inp}^{res} . The elements in W_{bias}^{res} are all equal to 1 and are also scaled. These connection factors and scaling parameters together with γ are optimized by a Monte-Carlo simulation to get statistically relevant results.

Before presenting the data to the reservoir, it is preprocessed to gain better results. Therefore the detection system has for each of the seizure types a specific though simple preprocessing step, which will be discussed in the relevant sections.

3 SWD Detection

3.1 The Reservoir Setup

As preprocessing, the best channel is manually selected by an expert from the EEG and resampled from 200Hz to 100Hz. For each of these datasets the mean absolute value is determined. This value is used as a scaling factor which results in a mean absolute value of 1 for each of the datasets.

As shown in Fig. 1 an absence seizure has a main frequency around 8 Hz and several harmonics. The best results were obtained by using the absolute values of the following signals as input for the reservoir: the first derivative and the frequency bands around 8, 16 and 24 Hz.

The reservoir that is used consists of 200 neurons which are randomly connected to the input layer and each other with a connection fraction of 90%. For the spectral radius of the reservoir 1.3 was used and the connections from the

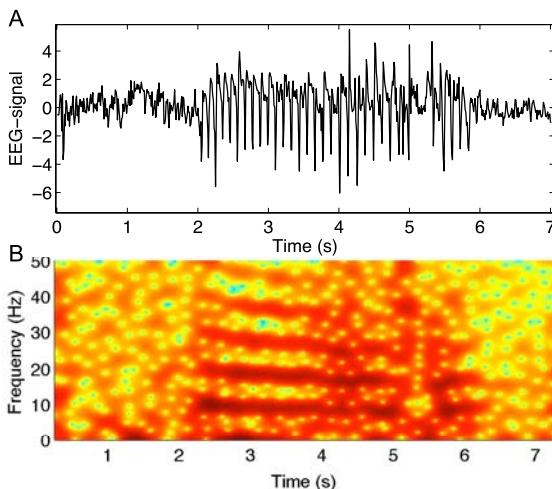


Fig. 1. Absence seizure (A) with its frequency content (B). An absence seizure has a main frequency around 8Hz and several harmonics. These harmonics together with the first derivative are used as input to the reservoir.

input to the reservoir are scaled up to 1.6. These values might seem unstable (see section 2), but due to the bias (of 2.8) and the non-linearity in the neurons this results in stable reservoirs that operate in the highly non-linear area of the hyperbolic tangent functions. The optimal value for the leak rate was found to be $\gamma = 0.06$.

As training the output weights were optimized to generate an output of 1 when the considered sample is part of a seizure and -1 in the opposite case. To classify the output a simple threshold is used. If an output sample is higher than the threshold it is considered part of a seizure, in the other case not part of a seizure.

3.2 Evaluation

A comparison was made with five other detection methods for SWDs. The first one was presented by Van Hese et al. in [8] and considers the energy in an EEG signal to distinguish between normal EEG and a seizure. The second method was presented by Westerhuis et al. in [9] and uses the first derivative of the EEG to detect seizures. The third was presented by Fanselow et al. in [10] and uses the amplitude of the EEG signal. The fourth method by White et al. [11] makes use of a simple spike detector combined with a autocorellation constraint. And the last method that was used for comparison was presented by Osorio et al. [12] and uses the wavelet transform of the EEG signal. The first four methods use intervals in which the EEG signal is tested and state that a seizure has to last at least a certain amount of intervals. To make a fair comparison the length of these intervals was optimized on the same data as was used to train the reservoir.

In order to generate ROC curves the specificity and sensitivity is measured for each threshold value between the maximum and minimum output value. In ROC curves sensitivity is plotted versus specificity and thus the higher the Area Under Curve (AUC) the better the performance.

Every result is based on the output of the complete dataset, in total 15 hours and 17 minutes of EEG-data coming from 13 different Genetic Altered Epilepsy Rats from Strasbourg, GAERS rats [13]. The data contains seizures that last from about 8 seconds to a little more than 2 minutes with a frequency of about 1.5 seizures every minute. The reservoir and the intervals are trained on only 10% of this data which was enough to achieve good results. For testing all data was used.

In Fig. 2 the ROC curves are shown for the five methods together with the presented Reservoir Computing approach. It is clear that previous work is outperformed by the method using Reservoir Computing which results in a maximum AUC of 0.987.

The detection delay is defined as the time needed for the reservoir output to become higher than the threshold value after the start of a seizure. To make this consistent the threshold value for which specificity and sensitivity are equal (the Equal Error Rate or EER point) is used. Since all methods, except the one presented by Osorio et al., used for a comparison are not created with the intention to detect in real-time, they don't perform well on detection speed.

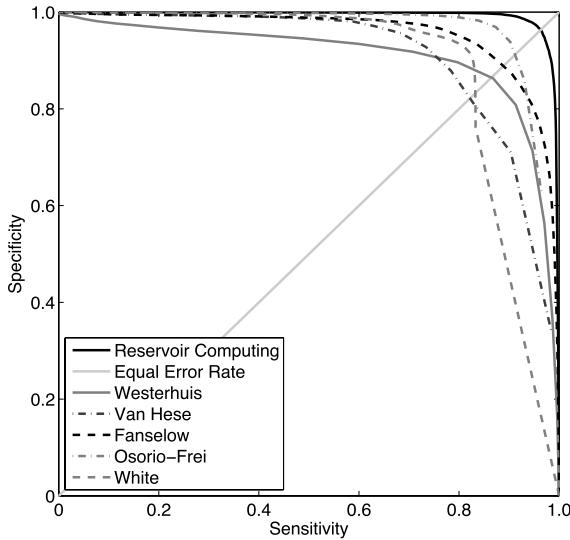


Fig. 2. ROC curves showing the better performance achieved using Reservoir Computing with a AUC of 0.987. The method by Van Hese et al. results in a AUC of 0.906, the method by Westerhuis et al. in a AUC of 0.910, the method by Fanselow et al. in a AUC of 0.956, the method by White et al. in a AUC of 0.885 and the method by Osorio and Frei in a AUC of 0.959.

These methods therefore resulted in a detection delay of more than 3 seconds. The method by Osorio et al. results in an average detection delay of only 0.9 seconds and with a standard deviation of 0.55 seconds. All methods are again outperformed by the presented method which results in an average detection delay of 0.3 seconds and 0.2 standard deviation.

4 Detection of Tonic-Clonic Seizures

4.1 The Reservoir Setup

Although tonic-clonic seizures create a very different pattern in the EEG, as shown in Fig. 3A, these seizures can be detected with almost the same setup as for SWDs. As shown in Fig. 3B, the frequency content doesn't contain the same harmonic structure as SWDs. Still it is clear that the higher frequency bands contain the most information.

For this type of seizures all four available channels were used instead of one. Each of them was again resampled, from 500Hz to 100Hz, and rescaled by dividing by its mean absolute value. To separate the high frequency content the simplest first order high-pass filter, the first derivative, is applied to all four of them.

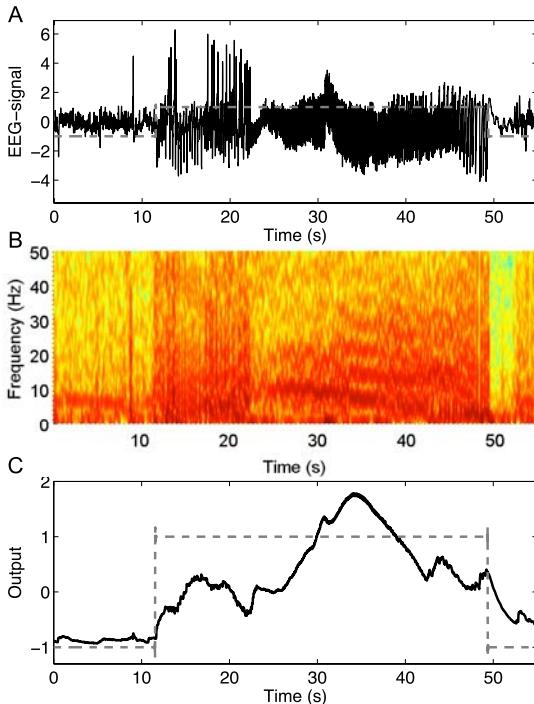


Fig. 3. Tonic-clonic seizure (A) with its frequency content (B) which doesn't show a main frequency for the seizure. The slow decay of the generated output (C) illustrates the slowness of the reservoir.

To optimize the parameters we started from the same reservoir of 200 neurons as used in the previous section. The only setting that differed much from the previous setup was the leak rate. A clearly lower optimum was achieved for $\gamma = 0.003$, which results in a lower cutoff frequency for the lowpass filter. A post-processing method wasn't used in this setup.

Fig. 3C shows the output generated on the seizure plotted in Fig. 3A. The figure illustrates that a low threshold (about -0.5) is needed to get good results and that most false-positive results have their origin in the slow decay of the output after the seizure. This was not the case with SWDs and is caused by the lower cutoff frequency of the lowpass filter in the neurons. A lower cutoff frequency results in a slower change in neuron activations and thus a slower reservoir.

4.2 Evaluation

In total 4 hours and 23 minutes of data of 4 different rats was used. The shortest seizure, shown in Fig. 3A, lasted about 40 seconds, the longest a little less than 12 minutes. This is remarkably longer than SWDs. Since the data studied contained only one tonic-clonic seizure every 40 minutes only the seizures together

with some non-seizure data before and after the seizure were cut out of the data. This resulted in 20% seizure data and 80% non-seizure data. Only 20% of the total data set was needed for training to get good results, the other data was used to test the performance.

An average AUC of 0.993 was achieved with a detection delay of on average 1.5 seconds. In 85% of the cases the seizures were detected within 3 seconds. This is a factor 5 slower than with SWD detection but still quite fast. To be used in a closed-loop environment the detection system needs to be able to detect in less than a second and even be able to predict the seizures in order to suppress them with fast-working medication.

5 Conclusion and Future Work

We have shown that it is possible to detect absence and tonic-clonic seizures on rat data with Reservoir Computing in a very reliable way. For both seizure types it resulted in a surface under the ROC curve of about 0.99. Therefore it can be used as a reliable aid to evaluate EEGs where the amount of seizures needs to be determined.

The proposed method is not only reliable, but it is also fast: 85% of the tonic-clonic seizures are detected within 3 seconds. Absences were detected with even less delay. Here 85% of the seizures were detected within 0.5 seconds.

Both conditions, reliability and low detection delay, enable the use of Reservoir Computing in a real-time warning or even a real-time treatment system for absence epilepsy in rats. However to achieve the same for tonic-clonic seizures more research needs to be done to further reduce the detection delay. Although good results were achieved for rat data, it is also paramount to test this system on human data before considering a real-time warning or treatment system.

In order to thoroughly evaluate the presented method it is necessary to compare the given results with more methods than the ones used here, certainly for tonic-clonic seizures where no comparison was made. Also a larger data set is needed for a better comparison.

References

- [1] Annegers, J.F.: The epidemiology in epilepsy. In: The treatment of epilepsy: principle and practice, pp. 165–172. Williams and Wilkins, Baltimore (1996)
- [2] Schrauwen, B., Verstraeten, D., Van Campenhout, J.: An overview of reservoir computing: theory, applications and implementations. In: Proceedings of the European Symposium on Artificial Neural Networks (ESANN) (2007)
- [3] Jaeger, H.: The “echo state” approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology (2001)
- [4] Verstraeten, D., Schrauwen, B., Stroobandt, D.: Isolated word recognition using a Liquid State Machine. In: Proceedings of the 13th European Symposium on Artificial Neural Networks (ESANN), Evere, May 2005, pp. 435–440. D-side Publications (2005)

- [5] Jaeger, H.: Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the “echo state network” approach. Technical Report GMD Report 159, German National Research Center for Information Technology (2002)
- [6] Jaeger, H., Lukosevicius, M., Popovici, D.: Optimization and applications of echo state networks with leaky integrator neurons. *Neural Networks* 20, 335–352 (2007)
- [7] Legenstein, R.A., Maass, W.: Edge of chaos and prediction of computational performance for neural microcircuit models. *Neural Networks*, 323–333 (2007)
- [8] Van Hese, P., Martens, J.-P., Boon, P., Dedeurwaerdere, S., Lemahieu, I., Van de Walle, R.: Detection of spike and wave discharges in the cortical EEG of genetic absence epilepsy rats from Strasbourg. *Physical and Medical Biology* 48, 1685–1700 (2003)
- [9] Westerhuis, F., Van Schaijk, W., Van Luijtelaar, G.: Automatic detection of spike-wave discharges in the cortical EEG of rats. In: Measuring Behavior 1996, International Workshop on Methods and Techniques in Behavioral Research, Utrecht, The Netherlands, October 16-18 (1996)
- [10] Fanselow, E., Ashlan, P., Nicolelis, A.: Reduction of pentylenetetrazole-induced seizure activity in awake rats by seizure-triggered trigeminal nerve stimulation. *Journal of Neuroscience* 20, 8160–8168 (2000)
- [11] White, A., Willians, P., Ferraro, D., Clark, S., Kadam, S., Dudek, F., Staley, K.: Efficient unsupervised algorithms for the detection of seizures in continuous EEG recordings from rats after brain injury. *Journal of Neuroscience Methods* 152, 255–266 (2006)
- [12] Osorio, I., Frei, M., Wilkinson, S.: Real-time automated detection and quantitative analysis of seizures and short-term prediction of clinical onset. *Epilepsia* 39, 615–627 (1998)
- [13] Marescaux, C., Vergnes, M., Depaulis, A.: Genetic absence epilepsy in rats from Strasbourg - a review. *Journal of Neural Transmission* 35, 37–69 (1992)

Learning of Subgoals for Goal-Oriented Behavior Control of Mobile Robots

Sang Hyoung Lee¹, Sanghoon Lee¹, Il Hong Suh¹, and Wan Kyun Chung²

¹ The Department of Electronics Computer Engineering, Hanyang University, Seoul, Korea
`zelog@inco.l.hanyang.ac.kr, {shlee, ihsuh}@hanyang.ac.kr`

² The Department of Mechanical Engineering, POSTECH, Pohang, Korea
`wkchung@postech.edu`

Abstract. Subgoal learning is investigated to effectively build a goal-oriented behavior control rule with which a mobile robot can achieve a task goal for any starting task configurations. For this, states of interest are firstly extracted from successful task episodes, where the averaged occurrence frequency of states is used as threshold value to identify states of interest. And, subgoals are learned by clustering similar features of state transition tuples. Here, features used in clustering are produced by using changes of the states in the state transition tuples. A goal-oriented behavior control rule is made in such a way that proper actions are sequentially and/or reactively generated from the subgoal according to the context of states. To show the validities of our proposed subgoal learning as well as a goal-oriented control rule of mobile robots, a Box-Pushing-Into-a-Goal(*BPIG*) task is simulated and experimented.

1 Introduction

A goal-oriented behavior control rule can be considered as a sequence of behaviors to complete a task. Such behavior control rules are usually provided by planners, reinforcement learning, or human programmers. Planning or reinforcement learning technologies have been known to be partially effective in recovering working conditions by building new behavior plans or policies to cope with exceptional cases. Most of them, however, require building of a new plan or policy to go from a current situation to a goal, which would be costly when exceptional cases occur frequently.

If a robot is able to autonomously learn subgoals, it will be possible to complete a task by changing the order of subgoals [1], or, by producing a sub-plan or a sub-policy only for coping with the current subgoals. This is more time- and cost-efficient than newly planning or learning the entire task, saving efforts for building a new plan, or reducing search-space. Here, subgoals refer to the states which robots necessarily or frequently go through in the course of task execution. By learning subgoals, the robot is able to explore more effectively and accelerate learning in other task in the same or similar environments where the same subgoals are useful [2]. Subgoals are also useful in producing the nominal sequence of behaviors for executing tasks with behavior control rules.

Stolle *et al.* [3] developed a control approach that used a library of trajectories to establish a global control law or policy. A library of trajectories is used to create a global

policy by nearest-neighbor look up. However, since subgoals for a task have not been considered in their approach, a new library of trajectories will be required if a goal location is changed. Fukazawa *et al.* [4] proposed an algorithm that acquires intermediate goals between initial states and goal states for an agent executing multiple tasks. They focused on generating the priorities of subgoals. But, they have not discussed how to generate actions to reach subgoals. Sutton *et al.* [5], Mannor *et al.* [6], and Shen *et al.* [7] have proposed option-based reinforcement learning techniques for robot to learn behaviors more effectively for a task, where options imply trajectories between subgoals. However, they assumed that subgoals were available ahead of learning.

As we have seen above, there have been few works investigating subgoal learning. In this paper, we will propose an algorithm for subgoal learning to effectively build a behavior control rule, with which a mobile robot can achieve a task goal for any starting task configurations. In this case, a goal-oriented behavior control rule is built in such a way that proper actions are sequentially and/or reactively generated from subgoals according to the context of states.

The organization of this paper is as follows. In section II, details on subgoal learning will be described. In section III, a goal-oriented behavior control rule will be explained. In section IV, simulations and experimental results will be given to verify validities of the subgoal learning and the goal-oriented behavior control rule for a Box-Pushing-Into-a-Goal(*BPIG*) task. Finally in section V, concluding remarks will be provided.

2 Subgoal Learning

A goal-oriented behavior control rule of a task is defined as a sequence of *state-action* pairs which are reordered according to changes of states in such a way that a task is completed under several uncertainties due to nondeterministic environment, incomplete perception and/or probabilistic action effect.

To generate such a meaningful sequence of *state*(current state, next state)-*action* pairs for the task completion (See Fig. 1), it is required to extract *state-action* pairs of interest for the successful task completion from task episodes which include various types of successful task episodes. Then, some of those *state-action* pairs of interest are abstracted as attentive *state-action* pairs (here is after, subgoals) and are nominally ordered as a behavior sequence for the task completion. At the time of execution of the

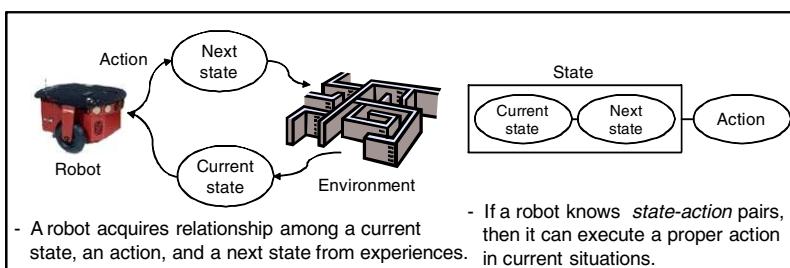


Fig. 1. A relationship of a *state* (current state, next state)-*action* pair

task, the nominal sequence could be reasonably reordered according to the context of the working environment [8].

2.1 Task Description Space(*TDS*) and Collection of Successful Episodes

TDS is a space for describing the task, in which a subgoal learner understands what task is given. Both a physical space (here is after, P_{space}) and a configuration space (here is after, C_{space}) [9] are composed of our proposed *TDS*. Tentative behavior sequences are found in a C_{space} and evaluated in a P_{space} . A state in the C_{space} of *TDS* may not be physically meaningful. Thus, a state needs to be checked whether the state is physically meaningful. This is done by using the P_{space} . Employing the C_{space} helps the subgoal learner to easily get many versatile successful task episodes from which a generic subgoal can be acquired. A state \mathbf{S} of the C_{space} for a task can be defined as

$$\mathbf{S} = [d_{O_1 O_2}, a_{O_1 O_2}, \dots, d_{O_{n-1} O_n}, a_{O_{n-1} O_n}] \quad (1)$$

where $d_{O_1 O_2}$, $a_{O_1 O_2}$, ..., $d_{O_{n-1} O_n}$, and $a_{O_{n-1} O_n}$ respectively, imply distance between the object O_1 and the object O_2 ($d_{O_1 O_2}$), angle between the object O_1 and the object O_2 ($a_{O_1 O_2}$), ..., distance between the object O_{n-1} and the object O_n ($d_{O_{n-1} O_n}$), angle between the object O_{n-1} and the object O_n ($a_{O_{n-1} O_n}$). A path in our *TDS* from an initial state (\mathbf{S}_{init}) to a goal state (\mathbf{S}_{goal}) can be found by applying the A^* algorithm in such a way that states to be tested are chosen in the C_{space} , and then those states are checked as physically possible states in the P_{space} .

2.2 Extraction of *TDS* States of Interest

All states of successful paths in *TDS* are *TDS* state vectors from which candidates of the *TDS* states of interest can be extracted. Here, we denote \mathbb{S}_{total} to be the total set including *TDS* state vectors. In \mathbb{S}_{total} , there can be a lot of identical *TDS* state vectors. On the other hand, there are many *TDS* states with a single occurrence. The objective of our subgoal learning is to extract generic subgoals from the cases with multiple occurrences. For this, $frequency(\mathbf{S}_i, \mathbb{S})$ is defined as

$$frequency(\mathbf{S}_i, \mathbb{S}) \triangleq number\ of\ satisfying\ (\mathbf{S} = \mathbf{S}_i) \wedge (\mathbf{S} \in \mathbb{S}) \quad (2)$$

where \mathbb{S} is the set including *TDS* state vectors, and \mathbf{S} and \mathbf{S}_i are a *TDS* state vector. And, now $\mathbb{S}_{duplicate}$ and \mathbb{S}_{unique} are define as

$$\mathbb{S}_{duplicate} = \{\mathbf{S}_i \mid (\mathbf{S}_i \in \mathbb{S}_{total}) \wedge (frequency(\mathbf{S}_i, \mathbb{S}_{total}) \geq 2)\} \quad (3)$$

and

$$\mathbb{S}_{unique} = \{\mathbf{S}_i \mid (\mathbf{S}_i, \mathbf{S}_j \in \mathbb{S}_{duplicate}) \wedge (\mathbf{S}_i \neq \mathbf{S}_j)\}. \quad (4)$$

In Eqs. (3) and (4), $\mathbb{S}_{duplicate}$ and \mathbb{S}_{unique} respectively, imply the set of *TDS* states having frequency greater than or equal to 2, and the set that does not allow the duplicated *TDS* state.

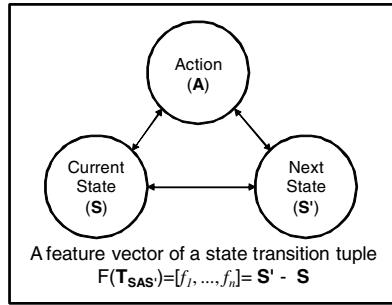


Fig. 2. A concept of a feature vector in a state transition tuple

Let f_{th} be the averaged value of occurrence frequency which can be computed as

$$f_{th} = \frac{|\mathbb{S}_{\text{duplicate}}|}{|\mathbb{S}_{\text{unique}}|} \quad (5)$$

where $|\mathbb{S}_{\text{duplicate}}|$ and $|\mathbb{S}_{\text{unique}}|$ imply the cardinality of the set $\mathbb{S}_{\text{duplicate}}$ and $\mathbb{S}_{\text{unique}}$. Subgoals will be made for the TDS states having occurrence frequency greater than or equal to f_{th} . A set of such candidate TDS states, TDS states of interest ($\mathbb{S}_{\text{interesting}}$), can be obtained as

$$\mathbb{S}_{\text{interesting}} = \{\mathbf{S}_i \mid (\mathbf{S}_i \in \mathbb{S}_{\text{duplicate}}) \wedge (frequency(\mathbf{S}_i, \mathbb{S}_{\text{duplicate}}) \geq f_{th})\}. \quad (6)$$

It is remarked that states of interest can be obtained by means of occurrence frequency, because there should be same or similar situation for the robot to encounter during the execution of the tasks from various initial configurations to the same goal configuration. Those frequently encountered situations in a real environment, or states in our TDS could be possible subgoals to achieve the goal.

2.3 State Transition Tuple and Clustering

We produce state transition tuples by using TDS states of interest, which are the candidates of subgoals. Let $\mathbf{T}_{SAS'}$ be the state transition tuple which is defined as

$$\mathbf{T}_{SAS'} \triangleq (\mathbf{S}, \mathbf{A}, \mathbf{S'}). \quad (7)$$

where \mathbf{S}' is a next state, \mathbf{S} is a current state, and \mathbf{A} is an action which forced \mathbf{S} to become \mathbf{S}' . From state transition tuples $\mathbf{T}_{SAS'}$, following information can be extracted (See Fig. 2). (1) Is the state vector changed? (2) If so, what elements of the state vector are changed? (3) Is the value of change increased or decreased? The feature vector of $\mathbf{T}_{SAS'}$ is now defined as

$$\mathbf{F}_{SAS'} = [f_1, \dots, f_n]. \quad (8)$$

In Eq. (8), for all $i = 1$ to n , f_i is defined as

$$f_i = \begin{cases} 1, & \text{if } s_i < s'_i, \\ 0, & \text{if } s_i = s'_i, \\ -1, & \text{if } s_i > s'_i. \end{cases} \quad (9)$$

It is recalled that there may be a lot of different $\mathbf{T}_{\text{SAS}'}$ for task episodes. And, different $\mathbf{T}_{\text{SAS}'}$ can have the same $\mathbf{F}_{\text{SAS}'}$. Thus, those state transition tuples are classified in such a way that $\mathbf{T}_{\text{SAS}'}$ having the same $\mathbf{F}_{\text{SAS}'}$ are included in the same cluster.

2.4 Generalization

It is necessary to find out the attentive (or representative) *TDS* state vector ($\mathbf{S}^{\mathbb{C}_i}$) for executing a task with each cluster. In this work, a concept learning approach is employed, where a general-to-specific ordering of hypothesis is applied to positive instances. The *more-general-than partial ordering algorithm* [10] is used to organize the search for an attentive *TDS* state vector $\mathbf{S}^{\mathbb{C}_i}$ with $\mathbf{T}_{\text{SAS}'}$ in a cluster. An attentive *TDS* state vector was found by using the algorithm. The algorithm will output the attentive *TDS* state vector.

3 Goal-Oriented Behavior Control Rule

3.1 Ordering of Subgoals

Each attentive *TDS* state vector representing a cluster will be regarded as a subgoal. Using subgoals, a goal-oriented behavior control rule is made in such a way that proper actions are sequentially and reactively generated from the subgoals according to states which could be expected from previous robot behaviors, or unexpected ones. For this, subgoals need to be grouped according to the distance from a subgoal to the goal.

Let $L(\mathbf{S}^{\mathbb{C}_i})$ be the distance from $\mathbf{S}^{\mathbb{C}_i}$ (the attentive *TDS* state vector of the i th cluster) to the goal state. If current *TDS* state \mathbf{S} of a tuple in a cluster \mathbb{C}_j is the same as next *TDS* state \mathbf{S}' of a tuple in other cluster \mathbb{C}_i , the distance of \mathbb{C}_i to the goal is longer than \mathbb{C}_j to the goal. $L(\mathbf{S}^{\mathbb{C}_i})$ is increased by one.

3.2 Action Selection Mechanism

N -best subgoals are selected by comparing probabilistic values of the current observed state \mathbf{Z} and attentive *TDS* state vector $\mathbf{S}^{\mathbb{C}}$ of subgoals as

$$\mathbb{C}_N = \sum_{i=1}^N \arg \max_{\mathbb{C}, \mathbb{C}_1 \notin \dots \notin \mathbb{C}_{i-1}} P(\mathbf{S}^{\mathbb{C}_i} | \mathbb{C}, \mathbf{Z}), \quad (10)$$

where \mathbb{C} implies a set of subgoals (clusters) and \mathbf{T} implies state transition tuples. And, the best matched tuple is selected by comparing the probabilistic value of \mathbf{Z} and all \mathbf{S} in state transition tuple of n -best subgoals as

$$\mathbf{T}_{\text{SAS}'} = \arg \max_{\mathbf{T}} P(\mathbf{S} | \mathbb{C}_N, \mathbf{Z}), \quad (11)$$

where \mathbb{C}_N implies selected n -best subgoals. Then, the action of the best matched tuple is activated sequentially to avoid attraction to a passed subgoal, and to take successful transition to a state toward next subgoal as

$$\mathbf{A} = \Phi(\mathbf{T}_{\text{SAS}'}) \quad (12)$$

where the function $\Phi()$ simply extracts an action in the selected tuple.

4 Simulations and Experiments

To show the validities of our proposed method of the subgoal learning for goal-oriented behavior control, a *BPIG* task is simulated and experimented. The *BPIG* task is having a robot push boxes to a goal across a room. To evaluate our subgoal learning for goal-oriented behavior control of the task, simulations are first performed, where simulations are executed in the same simulator used for P_{space} . But, starting configuration for each test is randomly selected so that it is totally different from those of episodes at the time of subgoal learning. A configuration space for the *BPIG* task, the C_{space} is a six-dimensional space. A state \mathbf{S} of the C_{space} for the *BPIG* task can be defined as

$$\mathbf{S} = [d_{rb}, a_{rb}, d_{rg}, a_{rg}, d_{bg}, a_{bg}] \quad (13)$$

Successful episodes were collected by using the A* algorithm. The results summarized in Table 1 show the process of subgoal learning by using the successful episodes. In Figure 3, the results of the simulation and the experiment are shown, where the *BPIG* task is executed with designed subgoals and our goal-oriented behavior control rule in section 3.

Table 1. Simulation Results of the *BPIG* task

BPIG task			
# of successful episodes	1,024	# of TDS states of interest	8,593
# of dimensions of C_{space}	6	# of subgoals	12
# of TDS states	13,500	-	-

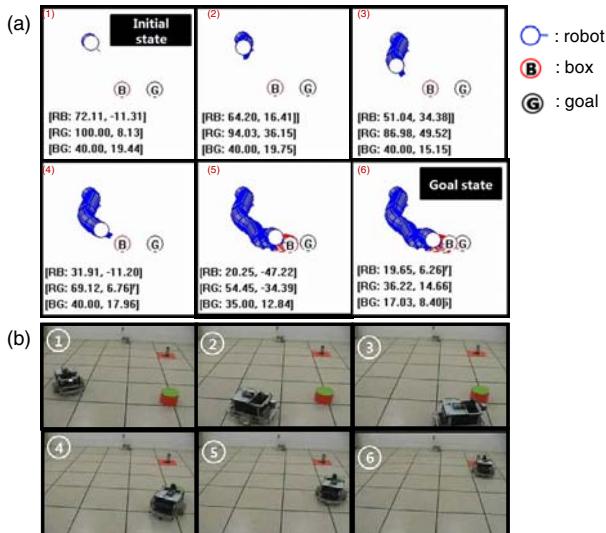


Fig. 3. Display of results for the *BPIG* task; (a) simulation results, (b) experimental results

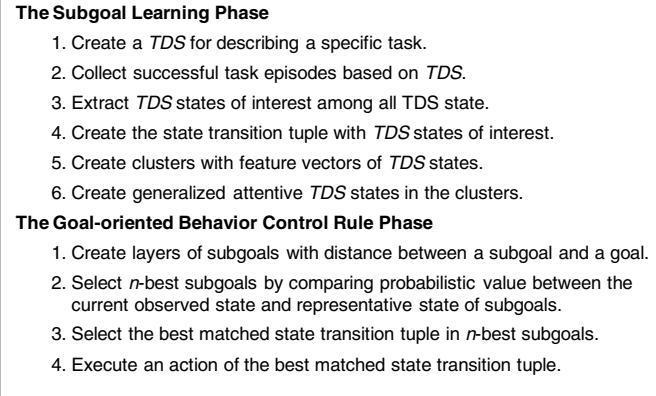


Fig. 4. Flow of Subgoal Learning for Goal-oriented Behavior Control Rule

To evaluate performances of our subgoal learning for goal-oriented behavior control, several experiments are carried out. Firstly, The *BPIG* task in an environment with sensing errors that have normal distribution $N(0, 1)$ and $N(0, 2)$. The task is observed for over 90% success rate in spite of sensing errors. Secondly, A robot executes a task in a dynamically changing environment by using subgoals for the *BPIG* task. The task was successfully completed even after boxes was added in the middle of task execution. This is an example showing that even in dynamically changing environments, a task can be completed by reordering subgoals that were already learned. Fig. 4 illustrates the entire flow of the previously-proposed subgoal learning for goal-oriented behavior control.

5 Conclusion and Further Works

We proposed a method for automatic subgoal learning to effectively build a goal-oriented behavior control rule with which a mobile robot can achieve a task goal for any starting task configurations. Our subgoals were made by extracting several attentive states from episodes, and by formulating a set of behaviors associated with such attentive states. Then, the states were organized to be an ordered collection of attentive states with their associated behavior sets, and were used to select a behavior from the set by means of the nearest neighbor look up. Thus, a goal-oriented behavior control rule could be robust since the subgoals have been designed by considering how to act for many possible states in episodes, which helps robots to cope with unexpected state transitions.

There are some points to be investigated in our future works. The subgoal learning for goal-oriented behavior control is to be applied to various kinds of service robot tasks. And, learning methods incorporating negative episodes and incremental learning technologies are to be investigated to deal with small size of episodes.

Acknowledgments. This work was supported by the IT R&D program of MKE/IITA. [2008-F-038-01, Development of Context Adaptive Cognition Technology].

References

1. Hue, C.W., Was, B.W., Chen, Y.: Subgoal ordering and Granularity Control for Incremental Planning. In: IEEE International Conference on Tools with AI (2005)
2. Goel, S., Huber, M.: Subgoal Discovery for Hierarchical Reinforcement Learning Using Learned Policies. In: International FLAIRS Conference, pp. 346–350 (2003)
3. Stolle, M., Atkeson, C.G.: Policies Based on Trajectory Libraries. In: IEEE International Conference on Robotics and Automation, Orlando, Florida, USA, pp. 3344–3349 (2006)
4. Fukazawa, Y., Trevai, C., Ota, J., Arai, T.: Acquisition of Intermediate Goals for an Agent Executing Multiple Tasks. *IEEE Transactions on Robotics* 22(5), 1034–1040 (2006)
5. Sutton, R., Precup, D., Singh, S.: Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. In: *AI*, vol. 112, pp. 181–211 (1999)
6. Mannor, S., Menache, I., Hoze, A., Klein, U.: Dynamic Abstraction in Reinforcement Learning via Clustering. In: The 21st International Conference on Machine Learning, Banff, Canada, pp. 560–567 (2004)
7. Shen, J., Gu, G., Liu, H.: Automatic Option Generation in Hierarchical Reinforcement Learning via Immune Clustering. In: Systems and Control in Aerospace and Astronautics, ISSCAA, pp. 19–21 (2006)
8. Suh, I.H., Kim, M.J., Lee, S., Yi, B.J.: A novel dynamic priority-based action-selection-mechanism integrating a reinforcement learning. In: IEEE Conference on Robotics and Automation, pp. 2639–2646 (2004)
9. LaValle, S.M.: Planning Algorithms, ch. 4. Cambridge Press (2006)
10. Mitchel, T.M.: Machine Learning, ch. 2. McGraw-Hill, New York (1997)

Coding Mechanisms in Hippocampal Networks for Learning and Memory

Yasuhiro Fukushima¹, Minoru Tsukada¹,
Ichiro Tsuda^{2,3}, Yutaka Yamaguti², and Shigeru Kuroda³

¹ Brain Science Institute, Tamagawa University, 6-1-1, Tamagawagakuen,
Machida, 194-8610, Japan

² Research Institute of Electronic Science, Hokkaido University, Japan

³ Department of Mathematics, Hokkaido University, Japan
y-fukusi@lab.tamagawa.ac.jp

Abstract. The following coding mechanisms in the CA3-CA1 hippocampal networks were examined. Firstly, the way in which the information of the spatio-temporal sequence of the hippocampal CA1 pyramidal cells was clarified by using the patch-clamp recording method. The input-output relations were analyzed by applying “spatial clustering index” and its “self-similarity” (Cantor-like coding) measure of the sequences. The membrane potentials were hierarchically clustered in a self-similar manner to the input sequences. The property was found to be present at one and two time steps retrograde in the sequences. The experimental results closely matched theoretical results of Cantor coding, reported by Tsuda and Kuroda (2001). Secondly, in the consolidation process, the spatiotemporal learning rule (STLR) composed of the spatial coincidence and its time history plays an important role in mapping the Cantor-like property onto synaptic weight space. The coexistence of STLR and Cantor-like coding in single pyramidal neuron of the hippocampal CA1 area is discussed from the viewpoint of coding mechanisms of reinforcement learning.

1 Introduction

The hippocampal network consists of three types of synapses that form a circuit. A spatial-signal serves as the input to the hippocampus and is transmitted through a synapse in the dentate gyrus (DG) to the CA3, then through another synapse to the CA1. There also exists a simultaneous input which directly connects to the CA1. The CA3 is also characterized by a distinct biological neural network which has a recurrent (feedback) connection. This circuitry compiles past memory into the present. On this subject, Nakazawa et al. (2002) have reported that after knocking out feedback in the CA3 of mice using genetic techniques, an extremely large number of cues become required to accomplish one action. According to these observations, it can be hypothesized that the hippocampal CA3 network forms a context of time sequence, while the CA1 network maps the spatiotemporal context into its synaptic weight space.

For the CA3→CA1 network, Tsuda (1996, 2001) proposed a computational model of chaos-driven contracting systems in which the unstable network (chaos-driven

network, CA3) forms a context of events via chaotic itinerary and the stable network (contracting dynamics, CA1) encodes its information via Cantor coding. In the CA1, Tsukada et al. (1996), proposed the SpatioTemporal Learning Rule (STLR), which maps spatiotemporal information onto CA1 synaptic weight space. Tsukada and Pan (2005) showed that STLR had the highest efficiency in discriminating pattern sequences. The novel points of this learning rule included the induction of cooperative plasticity without a postsynaptic spike and the time history of its input sequences. On the other hand, Hebb (1949) proposed that synaptic modification is strengthened only if the pre- and post-synaptic elements are activated simultaneously. We applied two rules to a single-layered neural network and compared its ability of separating spatiotemporal patterns with that of other rules, including the Hebbian learning rule and its extended rules. The simulated results (Tsukada and Pan, 2005) showed that STLR (non-Hebb) has the highest efficiency in discriminating spatiotemporal patterns. On the other hand HEBB has a natural tendency to attract analogous firing patterns into a representative one, put simply as “pattern completion” (Guzowski et al., 2004). From this it is concluded that STLR has a high ability in pattern separation, while HEBB has a high ability in pattern completion.

In this paper, we tested the following coding mechanisms in the CA3-CA1.

Coding mechanism 1: We aimed to clarify how the information of the spatio-temporal sequence of the hippocampal CA3 affects the postsynaptic membrane potentials of single pyramidal cells in the hippocampal CA1. To do so, we delivered spatio-temporal stimuli to Schaffer collaterals of the CA3 through a pair of electrodes, and recorded the post-synaptic membrane potentials using the patch-clamp recording method (Fukushima et. al, 2007). The input-output relations were analyzed by applying “spatial clustering” and its “self-similarity” (Cantor-like coding) measures of the sequences. The membrane potentials were hierarchically clustered in a self-similar manner to the input sequences. The property was significantly observed at one and two time steps retrograde in the sequences. The properties were maintained in two different stimulus conditions, weak and strong current stimulation. The experimental results were closely matched theoretical results of Cantor coding, reported by Tsuda and Kuroda (2001, 2004).

Coding mechanism 2: We have shown, experimentally, that both STLR and HEBB coexist in single pyramidal cells of the hippocampal CA1 area (Tsukada et al., 2007). In STLR, synaptic weight changes are determined by the “synchrony” level of input neurons and its temporal summation (bottom-up), while in HEBB, the soma fires by integrating dendritic local potentials or by top-down information such as environmental sensitivity, awareness, and consciousness (top-down). The role of soma spiking as top-down information raises a number of interesting computational predictions. Hippocampal theta is one of candidates for carrying top-down information which is driven by the medial septum (Buzsaki et al., 1983; Stewart and Fox et al., 1990). The theta stimulation of adult rat hippocampal synapses can induce LTP (Thomas et al., 1998). Second are the extrinsic modulators such as acetylcholine, serotonin, norepinephrine and dopamine. They can alter neuronal throughput and BAPs (so-called “meta-plasticity”) in such a way as to let the transmitters diffuse broadly (Tsubokawa and Ross, 1997; Sandra and Ross 1999; Pitler and Alger, 1992).

The coexistence of STLR and HEBB in single pyramidal neurons of the hippocampal CA1 area may support a coding mechanism of reinforcement learning.

At the last part of this paper, we explain the relation of these two learning rules. These two learning rules were mutually related, and laid as a basic mechanism of episodic memories. In the CA3-CA1 network, the two coding mechanisms above play an important role in the context formation of policy, reward, and value.

2 Cantor-Like Coding

2.1 Self-similarity and Spatial Clustering Property in Membrane Potentials of Hippocampal CA1 Pyramidal Neurons for a Spatio-temporal Input Sequence

In order to clarify how spatiotemporal sequence information from CA3 affects the CA1 pyramidal neurons hierarchically, we sequentially applied one of four spatiotemporal stimulation patterns via two independent pathways, and recorded the induced membrane potentials. Patch-clamp recording was made from pyramidal neuron of CA1 area in rat hippocampal slice. EPSPs were induced by extracellular electrical stimulation using two theta glass electrodes. The tips of the two electrodes were set to the Schaffer collaterals, proximal and distal sites to the soma. Successive ten pulses (33Hz) random sequence electrical stimulations, one of four patterns for 400 pairs. The four patterns of electrical stimulation were as follows: (1) electrical stimulation through both electrodes ("11" type), (2) electrical stimulation through one electrode ("10" type), (3) electrical stimulation through the other electrode ("01" type), and (4) no electrical stimulation ("00" type). The recorded responses were sequentially analyzed by using two measures; self-similarity and spatial clustering. "Cluster index" indicates how well the distribution of responses is clustered by the patterns of electrical stimulation.

A self-similarity index indicates a measure of hierarchical self-similarity. The four weighted centers for each time-history step of the spatial input pattern forms an observed pattern in amplitudes. The self-similarity measure is given by the number of neighboring permutations to get a target-ordered pattern corresponding to the standard pattern (the corresponding pattern to the 1st time-history step). The self-similarity index is normalized by the measure for the chance level. Our results indicate that statistically significant self-similarity occurred between two time-history intervals for sub- and supra-threshold conditions (Fig.1a).

The spatial clustering index indicates a pattern classification measure for the spatial input pattern ("11", "10", "01", and "00") at each time-history step. Each response was categorized into four groups by the pattern, one preceding pattern, or two preceding pattern of electrical stimulation. The clustering index indicates the probability that the distance between "its value and weighted-center of its own group" is not shortest among the four distances between "its value and weighted center of all four groups". Our results show significant clustering by two time-history steps for sub-threshold groups and by three for supra-threshold conditions (Fig.1b). These results suggest that CA1 neurons have the potential to code the level of spatial clustering and its self-similarity.

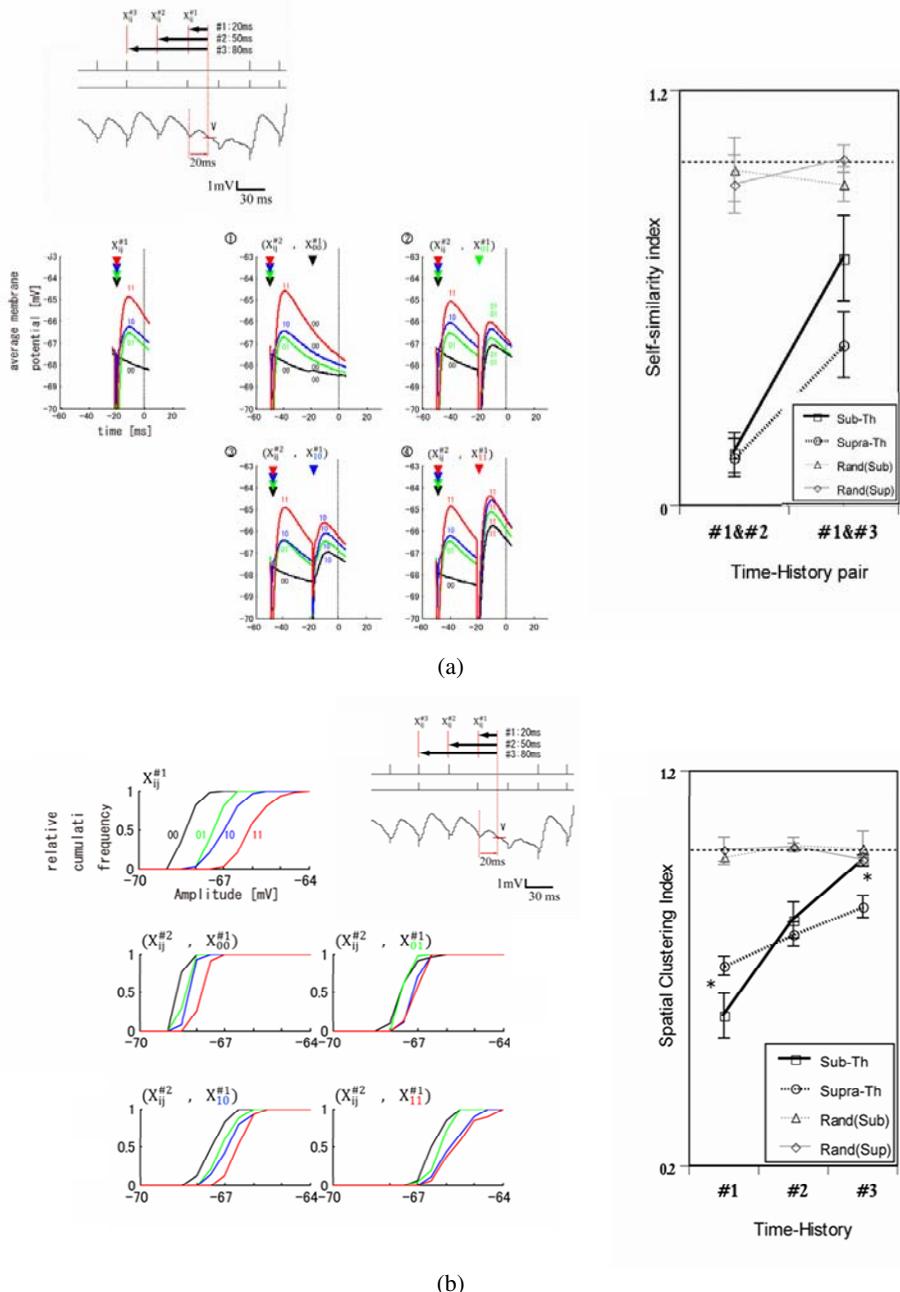


Fig. 1. (a) Self-similar property in hippocampal CA1 neurons. (b) Spatial clustering property in hippocampal CA1 neurons.

2.2 Comparison of Our Experimental Results with the Proposition of Cantor Coding

Tsuda and Kuroda reported the possibility of Cantor coding in hippocampal CA3-CA1 network (Tsuda, 2001; Tsuda and Kuroda 2001, 2003). Cantor coding is an information coding scheme for temporal sequences of events. In the theoretical Cantor coding model, each individual neuron in CA1 receives infinite chaotic inputs from CA3. In the model simulation, Tsuda and Kuroda showed Cantor coding in the membrane potential of CA1 neurons.

Because of experimental limitations, we used a spatio-temporal random input from CA3 to CA1 instead of infinite chaotic inputs to examine the spatial pattern classification and its self-similarity. In addition, the properties were maintained under two different stimulus conditions: weak and strong current stimulation (under sub- and supra-threshold conditions). Spatial clustering and its self-similarity were observed in one to three time history intervals. These results suggest that clustering and self-similarity are stable properties of CA1 neurons. These experimental results are very similar to the theoretical results of Tsuda and Kuroda (2001).

3 Hebb and Spatiotemporal Learning Rule (Non-Hebb) Interaction

3.1 Hebb and Non-Hebb Can Coexist in Single A1 Pyramidal Cells

The interaction between Hebb and non-Hebb occurs in a dendrites-soma system in a single pyramidal cell with many independent local dendrites in the CA1 area of the hippocampus. This system includes a spine structure, NMDA receptors, and sodium and calcium channels. The pyramidal cell integrates all of these local dendrite functions. The spatiotemporal learning rule and the Hebbian rule coexist in single pyramidal neurons of the hippocampal CA1 area (Tsukada and Pan, 2005; Tsukada et al., 2007). The Hebbian rule leads to the pattern completion and the spatiotemporal learning rule leads to the pattern separation.

Schematic illustrations were drawn in Figure-2abc. HEBB leads to pattern completion (Fig.2a). In contrast, STLR leads to pattern separation (Fig.2b). In the spatiotemporal learning rule, synaptic weight changes are determined by the “synchrony” level of input neurons (bottom-up) whereas in the Hebbian rule, the soma fires by integrating dendritic local potentials or by top-down information such as environmental sensitivity, awareness, and consciousness. The coexistence of the spatiotemporal learning rule (local information) and the Hebbian rule (global information) at the neuronal level may support this dynamic process that repeats itself until the internal model fits the external environment (Fig 2c). The dendrite-soma interaction (Fig 2c) in pyramidal neurons of the hippocampal CA1 area can play an important role in the context formation of policy, reward, and value in reinforcement learning.

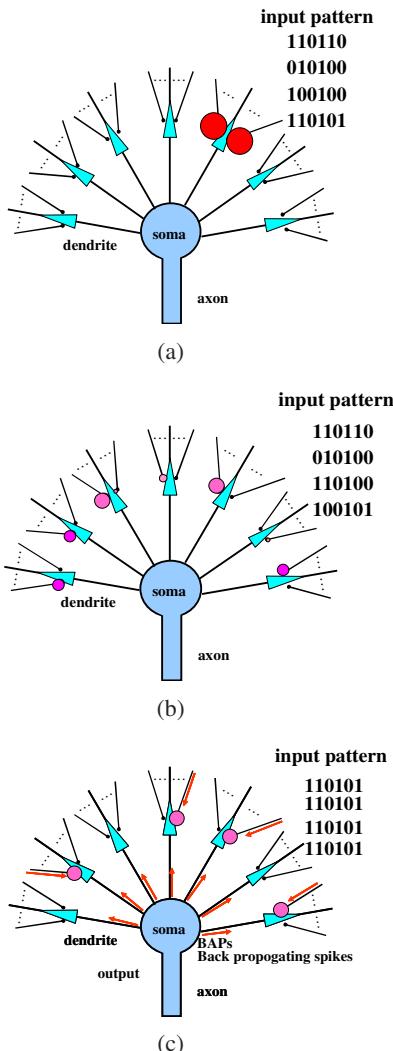


Fig. 2. (a) The Change in Synaptic weight according to the Hebbian Learning Rule. (b) The Change in Synaptic Weight according to the Spatio-temporal Learning Rule. (c) The Function of Local (dendrite) –Global (soma) Interaction and the Role of Back Propagating Spikes (BAPs).

3.2 Mechanisms of Reinforcement Learning in Single Cells

The role of soma spiking in relation to top-down information raises a number of interesting computational predictions. Hippocampal theta is one of the candidates for carrying top-down information which is driven by the medial septum (Buzsaki et al., 1983). The theta stimulation of adult rat hippocampal synapses can induce LTP (Thomas et al., 1998). Another candidate is extrinsic modulation by acetylcholine,

serotonin, norepinephrine and dopamine. They can alter neuronal throughput and BAPs (so-called “meta-plasticity”) in such a way as to let the transmitters diffuse broadly.

4 Relation between These Two Learning Rules

In the Cantor coding, the responses were the largest when the two input were applied at the same timing. This is corresponding to the first step of induction of STLR type LTP, because input at same timing to the neuron is very important in STLR learning. In additions, EPSP decay was directly affected to the Cantor coding, because the strong EPSP have high voltage after decay. STLR type LTP was affected by the decayed membrane response in Cantor-like coding. From these reasons, Cantor coding is the first step of short term memory consolidation, and affected to thereafter LTP by STLR type learning induction rule, and affects to the individual behavior related to episodic memory.

Acknowledgments. This study was supported by The 21st Century Center of Excellence Program(Integrative Human Science Program, Tamagawa Univ.)and by Grant-in-Aids for Scientific Research on Priority Areas- Integrative Brain Science Project and for KIBANN A NO.19200014 from Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. Buzsaki, G., Leung, L., Vanderwolf, C.H.: Cellular bases of hippocampal EEG in the behaving rat. *Brain Res. Rev.* 6, 169–171 (1983)
2. Fukushima, Y., Tsukada, M., Tsuda, I., Yamaguti, Y., Kuroda, S.: Spatial clustering property and its self-similarity in membrane potentials of hippocampal CA1 pyramidal neurons for a spatio-temporal input sequence. *Cognitive Neurodynamics* 1, 305–316 (2007)
3. Guzowski, J.F., Knierim, J.J., Moser, E.I.: Ensemble Dynamics of hippocampal regions CA3 and CA1. *Neuron* 44, 581–584 (2004)
4. Hebb, D.O.: *The Organization of Behavior*. John Wiley, New York (1949)
5. Nakazawa, K., Quirk, M.C., Chitwood, R.A., Watanabe, M., Yeckel, M.F., Sun, L.D., Kato, A., Carr, C.A., Johnston, D., Wilson, M.A., Tonegawa, S.: Requirement for hippocampal CA3 NMDA receptors in associative memory recall. *Science* 297, 211–218 (2002)
6. Pitler, T.A., Alger, B.E.: Postsynaptic spike firing reduces synaptic GABA_A responses in hippocampal pyramidal cells. *J. Neurosci.* 12, 4122–4132 (1992)
7. Sandler, V.M., Ross, W.M.: Serotonin modulated spike back-propagation and associated [Ca²⁺] changes in apical dendrites of hippocampal CA1 pyramidal neurons. *J. Neurophysiol.* 81, 216–224 (1999)
8. Stewart, M., Fox, S.E.: Do septum neurons pace the hippocampal theta rhythm? *Trends Neurosci.* 13, 163–168 (1990)
9. Thomas, M.J., Watabe, A.M., Moody, T.D., Makhinson, M., O'Dell, T.J.: Postsynaptic complex spike bursting enables the induction of LTP by theta frequency synaptic stimulation. *J. Neurosci.* 18(18), 7118–7126 (1998)

10. Tsubokawa, H., Ross, W.M.: Muscarinic modulation of spike back-propagation in the apical dendrites of hippocampal CA1 pyramidal neurons. *J. Neurosci.* 17, 5782–5791 (1997)
11. Tsuda, I.: A New type of self-organization associated with chaotic dynamics in neural networks. *Int. J. of Neural Sys.* 7, 451–459 (1996)
12. Tsuda, I.: Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems. *Behavioral and Brain Sciences* 24(5), 793–847 (2001)
13. Tsuda, I., Kuroda, S.: Cantor coding in the hippocampus. *Japan Journal of Industrial and Applied Mathematics* 18, 249–258 (2001)
14. Tsuda, I., Kuroda, S.: A complex systems approach to an interpretation of dynamic brain activity II: Does cantor coding provide a dynamic model for the formation of episodic memory? In: Érdi, P., Esposito, A., Marinaro, M., Scarpetta, S., et al. (eds.) *Computational Neuroscience: Cortical Dynamics. LNCS*, vol. 3146, pp. 129–139. Springer, Heidelberg (2004)
15. Tsukada, M., Pan, X.: The spatiotemporal learning rule and its efficiency in separating spatiotemporal patterns. *Biol. Cybern.* 92, 139–146 (2005)
16. Tsukada, M., Yamazaki, Y., Kojima, H.: Interaction between the spatiotemporal learning rule (STLR) and Hebb type (HEBB) in single pyramidal cells in the hippocampal CA1 area. *Cogn. Neurodyn.* 1, 157–167 (2007)

Developmental Stereo: Topographic Iconic-Abstract Map from Top-Down Connection

Mojtaba Solgi and Juyang Weng

Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, USA
{solgi,weng}@cse.msu.edu

Abstract. Engineering approaches to stereo typically use explicit search for the best matching between left and right sub-windows, which involves a high cost of search and unstable performance in the presence of binocular inconsistency and weak texture. The brain does not seem to conduct explicit search in the V1 and V2 cortex. But the mechanisms that the brain employs to integrate binocular disparity into 3-D perception is still largely a mystery. The work presented in this paper focuses on an important issue of integrated stereo: How the same cortex can perform recognition and perception by generating a topographic disparity-tuning map using top-down connections. As top-down connections with object-class supervisory signals result in topographic class maps, the model presented here clarifies that stereo can be processed by a unified in-place learning framework in the neural layers, and can generate iconic-abstract internal representation.

1 Introduction

It is largely unknown how the human visual system develops disparity tuned neurons in V1 [1], and how it integrates stereo with recognition. Existing methods for stereo disparity detection fall into three categories; (1) explicit matching (e.g., some discrepancy measure between the left and right patches [2], [3]), (2) using hand-designed features (e.g., using the phase information [4], [5]), and (3) network models that learn disparity from experience (e.g. [6], [7], [8], [9]). The categories (1) and (2) are generally called explicit matching approaches.

Among the different stages of the explicit matching approaches, the *correspondence problem* is believed to be the most challenging step; i.e. the problem of matching each pixel of one image to a pixel in the other [10]. Solutions to the correspondence problem have been explored using area-, feature-, pixel- and phase-based, as well as Bayesian approaches [2]. While those approaches have obtained limited success in special problems, it is becoming increasingly clear that they are not robust against wide variations in object surface properties and lighting conditions [4].

The network approaches in category (3) do not require a match between the left and right elements. Instead, the binocular stimuli are matched with binocular

neurons in the form of neuronal response. In the other words, both texture and binocular disparity are measured by a neuronal response - a great advantage of integration of binocular disparity and spatial pattern recognition.

However, existing networks that have been applied to binocular stimuli are either bottom-up SOM type or error-back propagation type. There has been no biological evidence to support error back-propagation, but Hebbian type of learning has been supported by Spike-Time Dependent Plasticity (SDTP) [11]. SOM type of networks that use both top-down and bottom-up inputs has not been studied until recently [12, 13, 14, 15]. In this paper we show that top-down connections that carry disparity information (e.g. when a monkey reaches an apple) enable neurons to self-organize according to not only bottom-up input, but also supervised disparity information. Consequently, the neurons that are tuned to similar disparities are grouped in nearby areas in the neural plane, forming what is called topographic disparity maps. Further, we experimentally show that such a disparity based internal topographic grouping leads to improved disparity classification.

Due to the top-down connection and the optimality of the LCA features used by the network, we are able to test stereo disparity detection from a large number of stereo views (rows) extracted from natural images. The network received 90% and 95% correct classification results from such images, depending on the computational resources used (i.e. number of neurons).

In the remainder of the paper, we first introduce the architecture of the network in Section II. Next, the implementation and results are presented in Section III. Section IV analyzes the developed networks. Finally, we provide some concluding remarks in Section V.

2 Network Architecture

The architecture of the multi-layer network is shown in Fig. 1. For biological plausibility, assume that the signals through the signal lines are all *non-negative*. Two types of synaptic connections are possible, excitatory and inhibitory. Only lateral connections can be excitatory and inhibitory. Bottom-up and top-down connections are all excitatory.

This is a recurrent network. Fig. 1 indicates how every neuron (*marked c* at the center) is connected to every other neuron in two ways. A fully connected network is used for the results presented here.

As the connection from neuron *c* to another neuron *c'* needs to pass a synapse, it is typical to consider the synapses as belonging to the post-synaptic neuron *c'*. Thus, we need only to consider input synapses (weights) in our modeling.

For each neuron *c*, at layer *l*, there are three types of connections, as shown in Fig. 1:

1. Bottom-up connections: connections that link from the neurons in the earlier layers (e.g., layer *l* – 1). All the sources of bottom-up inputs for a neuron are denoted as space *X*.

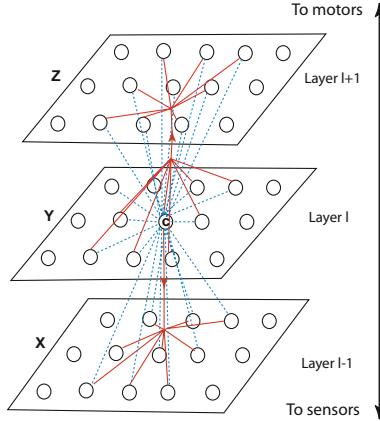


Fig. 1. The architecture of the proposed networks. A circle indicates a cell (neuron). Solid segment from each cell indicates its axon. A connection between a solid signal line and a cell indicates an excitatory or inhibitory connection. A connection between a dashed signal line and a cell indicates a dendrite (input) connection. Thus, all the connections are two-way.

2. Lateral connections: connections to neurons in the same layer. All the sources of later inputs for a neuron are denoted as space Y .
3. Top-down connections: connections to the neurons in the later layers (e.g., layer $l + 1$). All the sources of top-down inputs for a neuron are denoted as space Z .

Consider that this network computes at discrete times, $t = 0, 1, 2, \dots$, as a series of open-ended learning experiences after the beginning at time $t = 0$. Given each sensory input in Layer 1, all the neurons in the network respond and update in parallel. During the training phase, the responses of the motor layer are set by the given desirable output. During the test phase, the responses of the network are computed. When the response of layer l is being computed, each neuron in layer l uses the responses of other neurons at the previous time instance for lateral connections and top-down connections. As long as the entire network is updated significantly faster than the changes in the input layer and the motor supervision layer, the response of the network will track to the current input at the input layer and the current supervision at the motor layer. In the current work, we do not address the temporal change and the use of network dynamics for temporal context.

The activation levels of neurons are computed by the following formula

$$z = (1 - \beta) \frac{\mathbf{x}_b}{\|\mathbf{x}_b\|} \cdot \frac{\mathbf{v}_b}{\|\mathbf{v}_b\|} + \beta \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|} \cdot \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} \quad (1)$$

before competition among neurons in the same layer.

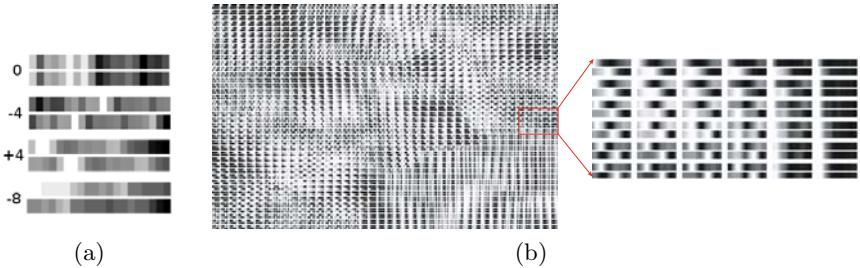


Fig. 2. (a) Examples of input, which consists of two rows of 20 pixels each. The top row is from the left view and the bottom row is from the right view. The numbers on the left side of the bars exhibit the amount of shift/disparity. (b) Bottom-up weights of 40×40 neurons in layer 1 using top-down connections. Connections of each neurons are depicted in 2 rows of each 20 pixels wide. The top row shows the weight of connections to the left image, and the bottom row shows the weight of connections to the right image.

3 Implementation and Results

The network has three layers: Layer 1 for input only without computation, Layer 2, and Layer 3 (or motor layer).

The input to the network is a pair of left and right rows, each 20 pixels wide. The image-rows were extracted randomly from 13 natural images (available from <http://www.cis.hut.fi/projects/ica/imageica/>). The right-view row position is shifted by -8, -4, 0, 4, 8 pixels, respectively, from the left-view row, resulting in 5 disparity classes. Fig. 2(a) shows some sample inputs. There were some image regions where texture is weak, which may cause difficulties in disparity classification, but we did not exclude them. During training the network was randomly fed with samples from different classes of disparity. The developed filters in Layer 2 are shown in Fig. 2(b). We conducted experiments to study the effect of top-down connections from Layer 3 (motor layer) to Layer 2. We define a probability for a neuron to respond to different classes of input as follows:

$$p(n, C_i) = \frac{f(n, C_i)}{\sum_{j=0}^m f(n, C_j)} \quad (2)$$

where n is the neuron, C_i represents class i , and $f(n, C_i)$ is the frequency for the neuron n to respond to the class C_i . These probabilities are shown in Fig. 3. After training the network was tested using a different set of test images that are different from training images. The performance is summarized in Fig. 5.

4 Analysis

4.1 Topographic Class Maps

As we see in Fig. 3, supervisory information conveyed by top-down connections resulted in topographically class-partitioned feature detectors in the neuronal

space, similar to the network trained for object recognition [16]. Since the input to a neuron in Layer 1 has two parts, the iconic input \mathbf{x}_b and the abstract (e.g. class) input \mathbf{x}_t , the resulting internal representation in Layer 1 is *iconic-abstract*. It is grossly organized by class regions, but within region it is organized by iconic input information. However, these two types of information are not isolated - they are considered jointly by neuronal self-organization.

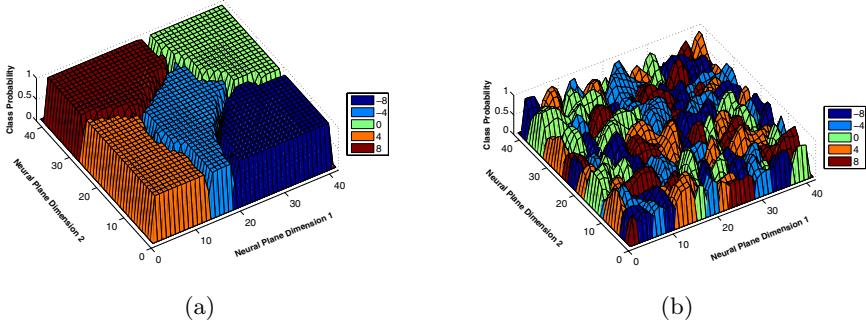


Fig. 3. The class probability of the 40×40 neurons of layer 1. (a) Top-down connections are active ($\beta = 0.5$) during development. (b) Top-down connections are not active ($\beta = 0$) during development.

To measure the purity of the neurons responding to different classes of disparity, we computed the entropy of the neurons as follows:

$$H = \sum_{i=1}^N -p(n, C_i) \log(p(n, C_i))$$

where $p(n, C_i)$ is defined in Eq. (2), and N is the number of classes. Fig. 4 shows that the topographic representation enabled by the top-down projections generalizes better and increases the neurons' purity significantly during training and testing.

4.2 The Effect of Top-Down Projection

As we see in Fig. 5, adding top-down projection signals improves the classification rate significantly. It can be seen that when $k = 50$ for the top- k updating rule, the correct classification rate is higher early on. This is expected as no feature detector can match the input vector perfectly. With more neurons allowed to fire, each input is projected onto more feature detectors. The population coding gives richer information about the input, and thus, also the disparity. When more training samples are learned, the top-1 method catches up with the top-50 method.

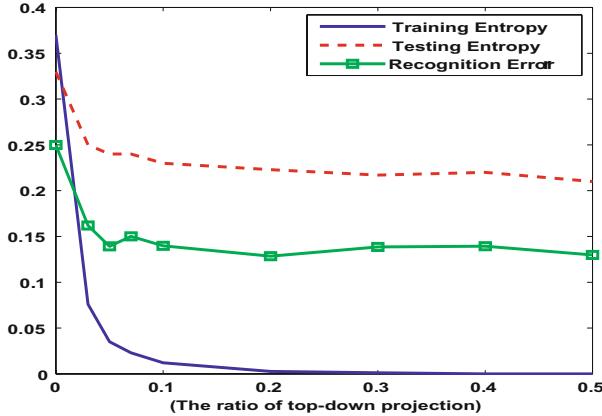


Fig. 4. The effect of top-down projection on the purity of the neurons and the performance of the network. Increasing β in Eq. (11) results in purer neurons and better performance.

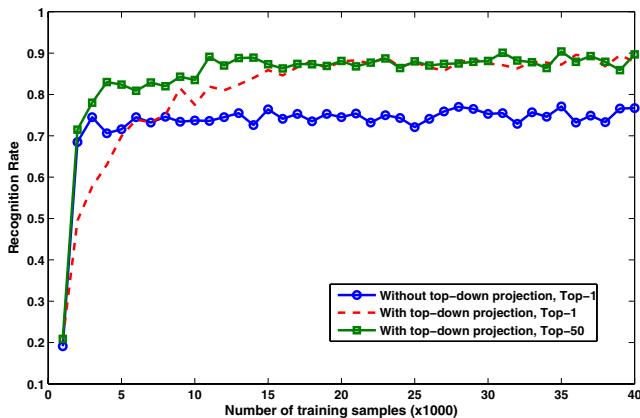


Fig. 5. The recognition rate versus the number of training samples. The performance of the network was tested with 1000 testing inputs after each block of 1000 training samples.

4.3 Developing Disparity-Tuned Receptive Fields

Assume that after learning, a neuron n is tuned to respond to a disparity class, C_d (i.e. the neuron purely responds to class C_d , as defined in section 3). The weight vector of a neuron n is

$$\mathbf{v}^n = (\mathbf{v}_{il}^n, \mathbf{v}_{rl}^n, \mathbf{v}_{rr}^n, \mathbf{v}_{ir}^n)$$

and an input vector \mathbf{x} is denoted by

$$\mathbf{x} = (\mathbf{x}_{il}, \mathbf{x}_{rl}, \mathbf{x}_{rr}, \mathbf{x}_{ir})$$

Consider the situation where the neuron receives a sample \mathbf{x} from class C_d . The activation level of the neuron is computed based on

$$\mathbf{v}^n \cdot \mathbf{x} = \mathbf{v}_{il}^n \cdot \mathbf{x}_{il} + \mathbf{v}_{rl}^n \cdot \mathbf{x}_{rl} + \mathbf{v}_{rr}^n \cdot \mathbf{x}_{rr} + \mathbf{v}_{ir}^n \cdot \mathbf{x}_{ir}$$

We know that $\mathbf{v}_{rl}^n = \mathbf{v}_{rr}^n$ and $\mathbf{x}_{rl} \approx \mathbf{x}_{rr}$. Therefore, we have:

$$\mathbf{v}^n \cdot \mathbf{x} = \mathbf{v}_i^n \cdot \mathbf{x}_i + 2\mathbf{v}_r^n \cdot \mathbf{x}_r$$

where $\mathbf{v}_i^n = (\mathbf{v}_{il}^n, \mathbf{v}_{ir}^n)$, $\mathbf{v}_r^n = (\mathbf{v}_{rl}^n, \mathbf{v}_{rr}^n)$, $\mathbf{x}_i = (\mathbf{x}_{il}, \mathbf{x}_{ir})$ and $\mathbf{x}_r = (\mathbf{x}_{rl}, \mathbf{x}_{rr})$.

Therefore, the network automatically elevates the effect of the relevant area in the neuron's response by a factor of 2.

Another mechanism to develop receptive fields is that LCA with top-down input develops discriminant features. It can be shown that if we scale up the relevant input subspace by a factor larger than 1, the self-organization mechanism develops neurons more along the relevant input subspace, which leads to more discriminant feature detectors that disregard irrelevant input information. A rigorous proof is beyond the scope of this paper due to space limit, but will appear elsewhere.

It is worth mentioning that due to spatial smoothness in natural scenes, the irrelevant part is not actually completely irrelevant to the relevant part. Hence, the term $\mathbf{v}_i^n \cdot \mathbf{x}_i$ partly contributes to the winner neurons.

5 Conclusion and Discussion

The problem of detecting disparity from stereo rows of natural images is quite challenging due to the wide variety of natural images. The developed network showed a good classification rate for 5 classes of disparity using a limited resource. With top-down connections, the neuronal layers are class-partitioned instead of class-mixed. As such class-partitioned representation (neuronal distribution and their weights) is tested with unseen visual stimuli, it enables the network to generalize significantly better. Thus, class-based topographic internal representation is not only biologically observed (e.g. fusiform face area (FFA)) and intuitively understandable, but also corresponds to superior internal representation in stereo disparity detection. This important property of internal representation is new for binocular networks.

Acknowledgements

The authors wish to sincerely thank Yan Wang for early discussion about this subject of network-based stereo. She supplied many references that this paper used. The authors also like to thank Paul Cornwell for his valuable comments and suggestions on this paper.

References

1. Parker, A.J.: Binocular depth perception and the cerebral cortex. *Nature Reviews Neuroscience* 8(5), 379–391 (2007)
2. Dhond, U.R., Aggarwal, J.K.: Structure from stereo - a review. *IEEE Transactions on Systems, Man and Cybernetics* 19(6), 1489–1510 (1989)
3. Zitnick, C.L., Kanade, T.: A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(7), 675–684 (2000)
4. Fleet, D.J., Jepson, A.D., Jenkin, M.R.M.: Phase-based disparity measurement. In: *CVGIP: Image Understand*, vol. 53, pp. 198–210 (1991)
5. Weng, J.: Image matching using the windowed Fourier phase. *International Journal of Computer Vision* 11(3), 211–236 (1993)
6. Lehky, S.R., Sejnowski, T.J.: Neural model of stereoacuity and depth interpolation based on a distributed representation of stereo disparity. *The Journal of Neuroscience* 70(7), 2281–2299 (1990)
7. Lippert, J., Fleet, D.J., Wagner, H.: Disparity tuning as simulated by a neural net. *Journal of Biocybernetics and Biomedical Engineering* 83, 61–72 (2000)
8. Wiemer, J., Burwick, T., Seelen, W.: Self-organizing maps for visual feature representation based on natural binocular stimuli. *Biological Cybernetics* 82(2), 97–110 (2000)
9. Franz, A., Triesch, J.: Emergence of disparity tuning during the development of vergence eye movements. In: *International Conference on Development and Learning*, pp. 31–36 (2007)
10. Marr, D.: *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Freeman, New York (1982)
11. Bi, G., Poo, M.: Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience* 24, 139–166 (2001)
12. Roelfsema, P.R., van Ooyen, A.: Attention-gated reinforcement learning of internal representations for classification. *Journal of Neural Computation* 17, 2176–2214 (2005)
13. Sit, Y.F., Miikkulainen, R.: Self-organization of hierarchical visual maps with feedback connections. *Neurocomputing* 69, 1309–1312 (2006)
14. Weng, J., Luciw, M.D.: Optimal in-place self-organization for cortical development: Limited cells, sparse coding and cortical topography. In: *Proc. 5th International Conference on Development and Learning (ICDL 2006)*, Bloomington, IN, May 31-June 3 (2006)
15. Weng, J., Luwang, T., Lu, H., Xue, X.: Multilayer in-place learning networks for modeling functional layers in the laminar cortex. *Neural Networks* 21, 150–159 (2008)
16. Luciw, M., Weng, J.: Topographic class grouping with applications to 3D object recognition. In: *Proc. International Joint Conf. on Neural Networks*, Hong Kong (June 2008)

An Analysis of Synaptic Transmission and Its Plasticity by Glutamate Receptor Channel Kinetics Models and 2-Photon Laser Photolysis

Hiroshi Kojima^{2,*} and Shiori Katsumata¹

¹ Computational Vision Science Laboratory, Tsukuba University, Tsukuba-shi,
Ibaraki 305-8573, Japan

² Laboratory for Cellular and Molecular Physiology, Tamagawa University,
Tokyo 194-8610, Japan
hkojima@lab.tamagawa.ac.jp

Abstract. We predicted the rate constants of AMPA receptor channels kinetics models of neural cells which could induce the change in efficacy of synaptic transmission by computer simulation. Excitatory postsynaptic currents (EPSCs) were reconstructed by computer calculation with the proposed kinetics models. Moreover, electrical responses were measured from the cells by using 2-photon laser uncaging. It was shown that the properties of the evoked current responses by photolysis and those obtained from spontaneous synaptic currents have the similar properties which indicates that the present AMPA receptor channel models mediate the evoked responses by laser photolysis. We investigated and proposed the possible rate constants, which could explain the changes in the EPSCs amplitude during LTP/LTD without changing the waveform, and corresponding physiological elements to them. Moreover, it is suggested that the present method based on kinetics models could be used for the investigation of other experiments evoked by laser photolysis.

1 Introduction

One of the most important synaptic receptor channels, AMPA receptor has been investigated in order to characterize the excitatory synaptic transmission and its plasticity in the central nervous system. Moreover, it has been suggested that changes in the efficacy of synaptic transmission such as long-term potentiation (LTP) and long-term depression (LTD), which are thought to be elementary mechanisms and substrates of learning and memory, are partly due to modifications and/or trafficking AMPA receptor channels in the postsynaptic membrane at the synapses. These kinetic models of receptor channels constructed from experiments have described precisely the physiological observations especially obtained by patch clamp technique. We estimated the rate constants in the kinetics models of AMPA receptor channels that might contribute to the enhancement and reduction of EPSC amplitudes during LTP and LTD by examining the computer generated EPSCs under an experimental constraint. At first

* Corresponding author.

the computer-generated EPSCs were reconstructed from the kinetics models proposed by other groups. They used the experimental results obtained from the membrane patches (in out-side out patch configuration) of both Pyramidal and Purkinje cells. We have recorded spontaneous miniature EPSCc and current responses activated by ultra-violet and 2 photon laser photolysis from the somata of neurons. The detailed analysis of the spontaneous (and/or miniature) EPSCs and laser-evoked current responses showed that they have the same properties and are mediated by the same types of AMPA receptor channels. We investigated and proposed the possible parameters from computer simulations that could explain the changes in the EPSCs waveform during LTP/LTD and also its corresponding biological factors. It was shown that the present method could be used for the investigation of synaptic plasticity in the central nervous system.

2 Model and Experimental Method

2.1 Computer Simulation of Synaptic Current Based on Kinetics Model

2.1.1 AMPA Receptor Channel Kinetics Model Obtained from CA3 Pyramidal Cell

The model has to predict the experimentally obtained ensemble current responses of outside-out patches to fast glutamate molecule applications (2). The model (figure 1) that we finally adopted assumes fast desensitization upon agonist binding before and after opening of the channel (4), where C_0 is the unliganded closed state, C_1 is the singly and C_2 is doubly liganded closed states, respectively. C_3 , C_4 and C_5 are desensitized, closed states, and O is the doubly liganded open state. c denotes the concentration of the glutamate molecules at the synaptic cleft and the vicinity of the receptor channels in the post-synaptic membrane. The rate constants used in the present simulation are not shown (10).

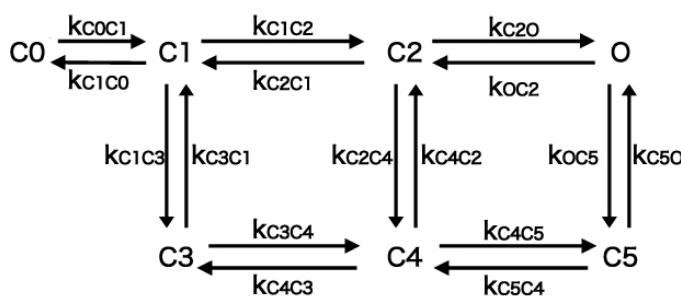


Fig. 1.

2.1.2 AMPA Receptor Channel Kinetics Model from Purkinje Cell

Kinetics models for AMPA receptor channels in cerebellum have been proposed from many laboratories (3, 8). A reaction scheme employed in the present study for the synapse between parallel fiber and Purkinje cell was basically the same as those

proposed previously by Haussner and Roth (5). The model has a total nine states, where C_0 is the unliganded closed state, C_1 is the singly liganded closed state, and C_2 is the doubly liganded closed state (c denotes glutamate concentration). C_3 to C_7 are desensitized (closed) states, with C_3 being singly liganded and C_4 to C_7 doubly liganded. O is doubly liganded open (conducting) state. The initial estimates for the rate constants in this model were the values in table 2 of Jonas *et al.* (7) together with estimates for the rate concerning to C_6 and C_7 .

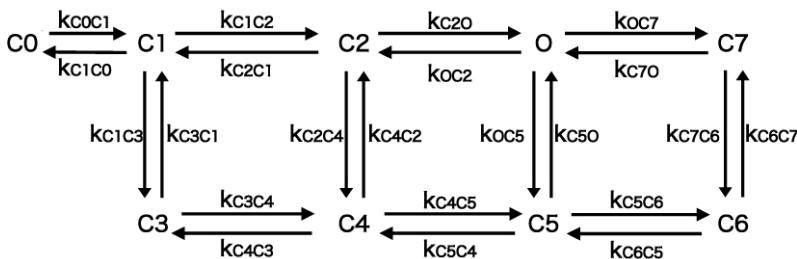


Fig. 2.

2.2 Design of the System for Photolysis

The present system for uncaging is originally designed in combination with a conventional confocal laser microscopy (LSM510 Meta, Carl-Zeiss, Jena, Germany) equipped with several visible wavelength laser. Ultraviolet (UV) light source for photolysis in the system is a commercial model (Enterprise II model 653, Coherent, CA) of an argon gas laser (351 and 364 nm, 80 mW) attached to the output exit of UV laser controls the output power of the ultraviolet laser beam. The preliminary experiment by using the present system were given in the previous paper.

2.3 Cerebellar Rat Slices, Hippocampal Neurons in Culture and Uncaging of MNI-Caged-L-Glutamate

Wistar rats (12-19 day old) were anaesthetized by diethylether and rapidly decapitated. Transverse cerebellar slices (270-300 μ m thickness) were cut in ice-cold ACSF (artificial cerebro-spinal fluid solution) using a vibrating slicer. They were moved to the recording chamber which was continuously perfused by normal ACFS at a rate of 2 ml/min bubbled with 95% O₂ and 5% CO₂. Patch pipettes were pulled from borosilicate glass tubing having a DC resistance of 4-7 M Ω . The internal solution had the composition of methanesulfonic acid (133 mM) based one. MNI-caged -L-glutamate was delivered to regions of cells for uncaging through a fine tube. Hippocampal pyramidal cells in culture were prepared by the protocol described elsewhere (9). The cultured cells were identified by Nomarski microscope inspection before use for experiments and transferred to a chamber which is continuously perfused with Tyrode solution. The internal solution of patch electrodes for recording had composition of CsCl (140mM) based one.

3 Results

3.1 EPSC Reconstructions from the AMPA/Kainate Receptor Channel Kinetics Model

The waveforms of the EPSCs reconstructed by these two kinetics models were investigated for LTP and LTD observed in CA1 pyramidal cell (Scheme 1) and cerebellar Purkinje cell (Scheme 2), respectively. The time course of transmitter release from the presynaptic terminal, which is represented by the concentration of glutamate molecules in the synaptic cleft, were given as two kinetics model (step -function (1.0 mM, 1.0 msec) and α -function (Fig. 3A, C)(3).

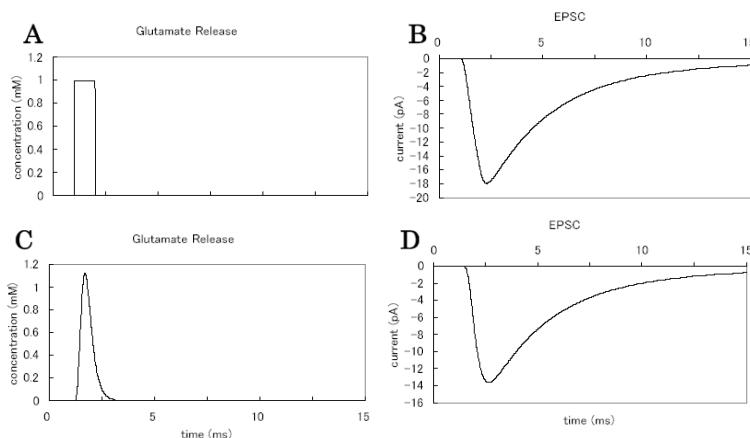


Fig. 3.

With these conditions and parameters, the reconstructed EPSC of pyramidal cell has a fast rise time less than 0.5 msec and a decay of single exponential function with time constant of 1.15 msec ($\tau = 1.15$ msec) which is shown in Fig. 3B and D. For simplicity, the profile of the concentration of transmitters at the synaptic cleft was assumed to be a step-function of which duration and amplitude are 1.0 msec and 1.0 mM, respectively. (Fig. 3A)

The reconstructed EPSC has the waveform of Fig. 3D for α -function and 3B for step function, respectively. The results suggests that two waveforms from the present two types of profiles of transmitter concentration give the same time course of EPSC and thus the simple step-like concentration change in the synaptic cleft is acceptable for transmitter release mechanism from the presynaptic terminal. Moreover, this assumption was also applicable for the EPSCs reproduced from the Purkinje cell type kinetics model of scheme 2 in Fig. 2 (result is not shown).

Next we have to test whether the present models could explain other sets of experimental results or not. Fig. 4 illustrates the computer generated paired-pulse stimulation in which two sequential transmitter pulses were applied to a larger outside out patches by using piezo driven theta tube. The result suggested that simulation of the paired pulse application of the glutamate molecules showed the depression of second current response that was already measured by patch-clamp experiment. The same kind of verification of the Purkinje cell kinetics model was also carried out. (data are not shown).

3.2 Change in Efficacy of Synaptic Strength and Rate Constants of Kinetics Models

From the experiments (whole-cell current recording), it has been reported that the waveforms of EPSCs before and during LTP/LTD have no change in their time course (rise time and decay time constant) except their amplitude. These three parameters determine uniquely the waveform of EPSC. Thus, we estimated the rate constants of the kinetics models that increase (for LTP) and decrease (for LTD) the amplitude of EPSCs without changing their time courses. The results of the estimation with 7-state model for LTP were listed in graph 5 which shows the change of three parameters (rise time τ_{rise} , decay time τ_{decay} , amplitude) by shifting a rate constants (C_0C_1 , C_1C_0 etc.) to the $10\times$ larger, and $0.1\times$ smaller value than control one without change in other all rate constants. Under the experimental condition, when the rate constant $k_{C_1C_2}$ was increased $10\times$ larger than control value, the rise time changes to 95% of control value, decay time to 104%, amplitude to 170%. The $k_{C_1C_2}$ determines

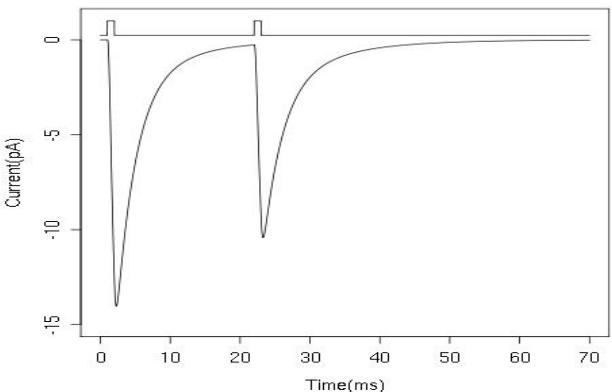


Fig. 4.

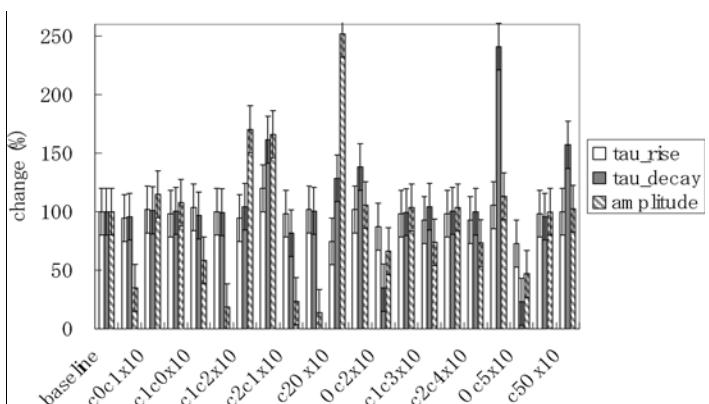


Fig. 5.

the binding speed of second agonist molecule to AMPA receptor channel which has been already singly occupied and is closed state and thus increase in this rate constant means that once the receptor is singly occupied state, the second agonist molecule is easily bind to the receptor. It is concluded from the present calculation based on the 7-states kinetics model that the rate constant k_{C1C2} would increase due to unknown mechanisms such as phosphorylation etc after a treatment which causes LTP and that this change in rate constant leads to the enhancement of the amplitude of synaptic current (EPSC) during LTP (1). In order to find the parameter that may cause LTD in Purkinje cell, the same estimation was conducted with the kinetics model proposed for AMPA receptor channels in the Purkinje cell membrane. The obtained results suggested that simultaneous modification of both k_{c2c4} (the rate constant of the transition from the state of the receptor having two agonists to desensitization state) and k_{Oc2} (the rate constant of the transition from the open state to the closed state) induces the reduction of the EPSC amplitude during LTD of cerebellar Purkinje cell (data is not shown).

3.3 Responses by Laser Photolysis and AMPA/Kainite Receptor Channel Kinetics Model

We measured the current responses from the soma of Purkinje cells by uncaging of MNI-caged glutamate using IR laser beam, which is focused (diameter of 3~5 μm) on a dentritic trees and soma of the cells. Figure 6A shows the location of the Purkinje cell soma on which the 2-photon (720 nm) laser beam was focused. The laser evoked current response measured from pipette accessed to the soma is shown in Figure 6B.

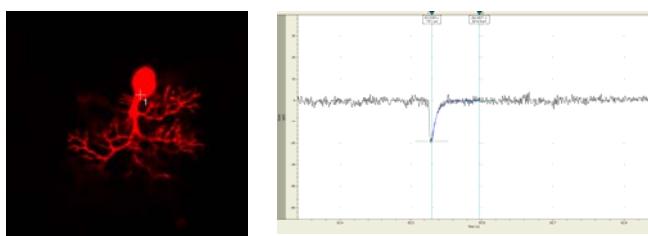


Fig. 6.

The decay time constant of the evoked response is 6.34 msec with 20 pA peak amplitude. Figure 6C is a typical example of a spontaneous miniature EPSC recorded from the soma of the same Purkinje cell. The decay phase of this miniature EPSC is fitted by a single exponential function having the decay time constant of 6.17 msec. Both recorded currents have the same decay phases that are well fitted by a single exponential function and the value of the time constants are almost equal. Together with pharmacological experiments (data is not shown), it is suggested that the current responses evoked by 2-photon laser uncaging are mediated by AMPA receptor channel which have the kinetic scheme shown in the previous section.

4 Conclusions and Summary

Firstly, by computer simulation, we reconstructed the synaptic currents (EPSC) from the proposed AMPA receptor channel kinetics model. Furthermore, we predicted the possible rate constants that could induce synaptic plasticity such as hippocampal LTP and cerebellar LTD. It was suggested from the present calculation that, for example, the rate constant k_{C1C2} in the kinetics model of pyramidal cells increases the speed of the second binding of agonist to receptor, which indicates that the change in affinity of glutamates to receptors partly causes LTP of hippocampus. The present method using reconstructed EPSC based on receptor channel kinetics model makes it possible to predict the physiological and molecular biological factors which are responsible for the synaptic plasticity. Secondary, the current responses measured by 2-photon laser uncaging have the same kinetics properties with those of the EPSCs recorder from the cells in slice preparations. Thus, the 2-photon laser photolysis can be widely applicable as a method for the stable and controllable simulation of fine areas of the neurons. Furthermore, the similar type of AMPA receptor channels models used in the present analysis mediate the evoked current responses by photolysis under voltage-clamp condition.

References

1. Andrasfalvy, B.K., Magee, J.C.: Changes in AMPA receptor currents following LTP induction on rat CA1 pyramidal neurons. *Journal of Physiology* 559(2), 543–554 (2004)
2. Colquhoun, D., et al.: Action of brief pulses of glutamate on AMPA/kainate receptor s in Patches from different neurones of rat hippocampal slices. *Journal of Physiology* 459, 261–287 (1992)
3. Destexhe, S., et al.: In: Koch, C., Segev, I. (eds.) *Methods in neuronal modeling*, 2nd edn., pp. 1–25. MIT Press, Cambridge (1998)
4. Dudel, J., et al.: Rapid activation, desensitization, and resensitization of synaptic channels of crayfish muscle after glutamate pulses. *Biophysical Journal* 57, 533–545 (1990)
5. Hausser, M., Roth, A.: Dendritic and somatic glutamate receptor channels in rat cerebellar Purkinje cells. *Journal of Physiology* 501(1), 77–95 (1997)
6. Jonas, P., Sakmann, B.: Glutamate receptor channels in isolated patches from CA1 and CA3 pyramidal cells in rat hippocampal slices. *Journal of Physiology* 455, 143–171 (1992)
7. Jonas, P., et al.: Quantal Components of Unitary EPSCs at themossy fibre synapse on CA3 pyramidal cells of rat hippocampus. *Journal of Physiology* 472, 615–663 (1993)
8. Kojima, H., et al.: Properties of AMPA receptor channels during long-term depression in rat cerebellar Purkinje cells. In: Kuba, et al. (eds.) *Slow Synaptic Responses and Modulation*, pp. 307–317. Springer, Heidelberg (2000)
9. Kojima, H., et al.: Ultraviolet laser beam and confocal microscopy –A system for rapid patterned photolysis. *IEEE Circuit & Devices, The eletronics & Photonics Magazine* 22(6), 66–74 (2006)
10. Lin, B., et al.: Long term potentiation alters the modulator pharmacology of AMPA-type glutamate receptors. *Journal of Neurophysiology* 87, 2790–2800 (2002)

A Biologically Inspired Neural CPG for Sea Wave Conditions/Frequencies

Leena N. Patel* and Alan Murray

University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh, EH9 5JL, UK
{L.Patel,A.F.Murray}@ed.ac.uk

Abstract. This paper shows that a biology-based neural network (called a central pattern generator (CPG)) can be re-evolved for sea conditions / frequencies. The fish's CPG operates at 1.74Hz to 5.56Hz, whereas we require performance to reach 0.05Hz to 0.35Hz (20s to 3s waves) for an alternative engineering problem. This is to enable adaptive control of wave energy devices, increasing their efficiency and power yield. To our knowledge, this is the first time a bio-inspired circuit will be integrated into the engineering domain (and for a completely different function). This provides great inspiration for utilising other neural network mechanisms for alternative tasks.

1 Introduction

Systems inspired by biology are becoming more widespread in the area of artificial intelligence, robotics and neural computation. This is in an effort to increase knowledge of the biological paradigms they are built upon and to develop intelligent systems. Of these, neural networks are applied fairly successfully. For example, at NASA, researchers use an artificial neural network (ANN) to organise and map aerodynamic changes during flight [2]. This information is then used by the flight control system to stabilise the aircraft. Further development of the ANN aims towards more direct control of decisions made, working alongside the flight control system.

Other ANNs mimic the behaviour of animals more closely, modelling the underlying neural architecture. As an example, [3] describes a hexapod robot, developed using coupled nonlinear oscillators. Their neural system is based on an abstracted version of central pattern generators (CPGs) governing insect locomotion.

CPGs are circuits, which control rhythmic activity such as walking, swimming, mastication and respiration. Importantly, these neural networks can self-modulate to meet the needs of changing environments. Inspired by this intelligent process, we investigate the effects of re-evolving CPG circuitry to solve a real problem in the engineering domain.

* Corresponding author.

The weights and neuron specific parameters of an isolated lamprey-CPG segment¹ are evolved using genetic algorithms to determine whether the network can still function in different ocean wave conditions. This work is essential to develop controllers for wave energy devices, which will optimise their efficiency and energy production capabilities. This is currently a very difficult problem for such non-adaptive units. The conditions and ranges at which the machines operate differ from those of the fish and therefore the neural circuit requires retuning to accommodate them.

In this paper, we report proof-of-concept experiments which evolve an isolated oscillatory network that can be used with single point absorbers or bobbing buoy devices such as AquaBuOY [4] rather than multi-unit articulated devices such as Pelamis [4].

2 Ocean Waves and Energy

Massive reserves of energy are stored in the motion of the seas. Technology to harness this energy is beginning to emerge (e.g. AquaBuOY in fig. 1(a), Pelamis), however, these devices operate at sub-optimal efficiency since they are unable to respond autonomously to changing sea conditions. They rely on inaccurate histories of wave measurements or “compromise” settings (such as damping levels) based on sea state models such as the Pierson-Moskowitz spectrum (fig. 1(b)).

Sea spectra are based on accumulated accelerometer measurements of waves and attempt to describe the sea state based on constant wind velocities blowing over a section of the ocean for a long period. While averaged data is useful for structural considerations and designing machines which operate within the boundaries of the bandwidth, they do not present a realistic picture of the sea state for devices that need to operate on a moment-by-moment basis. Typical wave conditions are much more complicated and variable. In fact, ocean waves are highly irregular and very difficult to predict.

Our strategy will use a bio-inspired approach, reacting to the sea in real time in the same way that a fish responds to changes in its environment. Similar motion characteristics form the basis of this idea; where both fish and machine perform managed oscillatory movements albeit for different purposes (*anguilliform* swimming vs. converting wave energy into usable electricity).

In this paper we describe a crucial preliminary goal, which is to develop controllers able to operate within the correct wave range. Single CPG modules are all that are needed at this stage, which when fully developed with feedback can be used with bobbing buoys such as AquaBuOY (shown in fig. 1(a)).

¹ The lamprey CPG comprises several linked copies of an oscillatory segment. Each segment can work in isolation to produce rhythmic patterns of activity. Coupling these units enables propulsion and coordinated movements of the entire fish’s body. A multisegment CPG is therefore only necessary for articulated wave energy devices where movement between segments of the machine need to be coordinated.

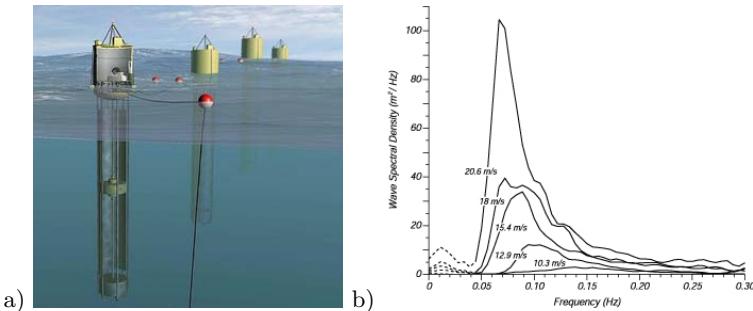


Fig. 1. a) AquaBuOY (a single point absorber developed by Finerva) b) the Pierson-Moskowitz Spectrum shows wave spectra for a fully developed sea [5]

3 Invertebrate Spinal Neural Network Model

The most extensively researched central pattern generator (CPG) for invertebrate swimming control belongs to the lamprey, an eel-like fish. It has been examined biologically, with *in-vivo* specimens innervated in a chemical bath [6,7] to determine detailed cell models. Computational experiments have produced a simplified model [8] (one oscillatory unit is shown in fig. 2) explored further in our previous work to determine its flexibility for swimming control [9].

In the network of fig. 2, rhythmic firing patterns (typical of CPG circuits) are maintained in the following way:

- 1) One side is dominantly active at any time. On this side, neuron type E excites cells on its own side while C inhibits all neurons on the opposite side.
- 2) After some time, a burst terminating mechanism causes control to switch sides. Burst termination is caused by L becoming active later in the cycle, suppressing the active C neuron and passing control from one side to the other [10].

Each population of neurons (shown in fig. 2) is modelled by a set of 1st order differential equations which represent their firing behaviour:

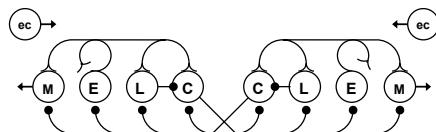


Fig. 2. Model showing a single oscillatory neural circuit governing burst activity for swimming in an invertebrate. There are four neuron types: E (Excitatory), C (Contralateral Inhibitory), L (Lateral Inhibitory) and M (Motorneurons) with inhibitory (dots) and / or excitatory (open forks) connections. ec (Edge Cells) provide feedback to the Central Pattern Generator.

$$\dot{\xi}_+ = \frac{1}{\tau_D} \left(\sum_{i \in \Psi_+} u_i w_i - \xi_+ \right), \quad (1)$$

$$\dot{\xi}_- = \frac{1}{\tau_D} \left(\sum_{i \in \Psi_-} u_i w_i - \xi_- \right), \quad (2)$$

$$\dot{\vartheta} = \frac{1}{\tau_A} (u - \vartheta), \quad (3)$$

$$u = \begin{cases} 1 - \exp \{(\Theta - \xi_+) \Gamma\} - \xi_- - \mu \vartheta & (u > 0), \\ 0 & (u \leq 0). \end{cases} \quad (4)$$

In this set of equations,

- ξ_+ and ξ_- are excitatory and inhibitory synaptic inputs respectively
- w_i their associated weights
- τ_D and τ_A represent time delays
- Ψ_+ and Ψ_- denote pre-synaptic excitatory/inhibitory neurons (eqns. 1-2).
- ϑ is a leakage term, included as delayed negative feedback (eqn. 3)
- Θ (threshold), Γ (gain) and μ (frequency adaptation) describe the dynamics of each neuron class. In the biological model, these are tuned to match response characteristics of each neuron type (see [8]).
- u represents the mean firing frequency of each neural population. Excitatory input is transformed by a transfer function providing saturation for high input levels (eqn. 4).

These equations model the behaviour and time-dependent functionality of each neuron class, which combined as in fig. 2 produce rhythmic oscillations.

4 Generating Parameters with Evolutionary Algorithms

Evolutionary techniques are frequently utilised to search and optimise parameters of a given problem. Of utmost importance is the construction of appropriate fitness functions which guide the algorithm through the search space.

Specific evolutionary algorithms include Particle Swarm Optimisation which is based on the flocking behaviour of birds, Ant Colony Optimisation, based upon how ants leave pheromone trails. The more commonly known Genetic Algorithm (GA) is based on Darwinian principles of natural selection. GAs are most widely used owing to their explorative and exploitative traversal through the search space, their reasonable success rate, usually within acceptable timeframes, and because they do not often get stuck in local minima.

Possible solutions are encoded as chromosomes with each gene representing one evolvable element of the problem. With each generation, the technique cycles through three main operations: selection, variation and elimination. *Selection* involves choosing two parent chromosomes (usually from fitter candidates) to produce a child solution. The genetic composition of each child chromosome solution is *varied* with operations such as mutation (where the values of genes are tweaked) and crossover (where parts of each parent are combined to make

an entire solution). Finally, weak solutions are *eliminated* and replaced with fitter candidates. The genetic approach is explained further in section 5 and exemplified by parameters and constructs used in this particular project.

5 Evolving a Sea Wave CPG Network

Central Pattern Generators (CPGs) are often optimised using GAs to determine neural weights and connections which mimic biology. We believe that this is the first example of optimisation towards an alternative engineering task. Integrating CPGs with wave power devices will enable them to react autonomously in varying sea conditions, and thus increase their efficiency and power output.

Motivated by this goal, our past work explored the flexibility of the lamprey neural network [910]. Evolved systems were simpler in design, yet performed with greater sophistication. This work explores controllers that can accommodate the different conditions they must operate in, that is, outside the biological range.

5.1 Sea Wave Genetic Algorithm Parameters

Chromosome solutions of the genetic process for this task comprise 43 genes, corresponding to weights, tonic input and parameters describing neuron behaviour, such as adaptation rate, threshold and gain. The underlying time delay values are increased by a factor of 15 (over those of the lamprey CPG) to suit the new environment². Due to the symmetric nature of the CPG model, it is only necessary to evolve half of the entire network.

Starting with an initial random population (with 100 candidates), operations of two-point crossover, mutation and pruning are applied each generation. Two-point crossover entails selecting two random locations (with 0.5 probability) in paired parents where genes are swapped between them. Mutation, also applied randomly (with 0.4 probability) involves varying some of the child's genes. Finally, pruning is a non-standard GA operation (but not a new one) which randomly (with 0.1 probability) sets genes to 0, eliminating their connection. This is applied to determine whether connectivity can be reduced and simpler configurations produced. A secondary pruning process is also applied at the end of the entire genetic process where weak connections (lower than 0.1, in decrements of 0.02) are eliminated provided that they do not produce inferior solutions. The GA is terminated after 400 generations as solutions converge by this point.

Fitness is guided by objectives which ensure 1) left and right output is out of phase, 2) regular single-peaked bursts of activity, 3) monotonically controllable frequency (using tonic inputs) and 4) reduced connectivity. Additionally, they must contain the range of frequencies to operate in sea waves (not addressed previously), requiring a whole new amended genetic process. Fitness functions are shown in table II

² Simply adjusting time delays does not accommodate the required operation range and attempts to evolve these parameters proved unsuccessful within reasonable timeframes.

Table 1. Fitness Evaluation Equations to Evolve Neural Controllers for Ocean Waves

Objective	Mathematical Definition	Bad	Good
Oscillations	$(zerosL + zerosR)/2$	2	12
Standard deviation	$\sqrt{\sum_{t=1}^n (U_l(t) - \bar{U}_l)^2} + \sqrt{\sum_{t=1}^n (U_r(t) - \bar{U}_r)^2}$	0.1	0.5
L-R period diff.	$\frac{\sum_{cycle=1}^c P_l(cycle) + \sum_{cycle=1}^c P_r(cycle)}{2c}$	0.15	0
Period diff.	$\frac{ P_l(c) - P_l(c-1) }{2P} + \frac{ P_r(c) - P_r(c-1) }{2P}$		
Signal diff. (consecutive)	$\left(\frac{\sum_{t \in lastcycle} U_l(t) - U_l(t-P) }{\sum_{t \in lastcycle} U_l(t) - U_l(t-P) } \right) / 2$ $+ \left(\frac{\sum_{t \in lastcycle} U_r(t) - U_r(t-P) }{\sum_{t \in lastcycle} U_r(t) - U_r(t-P) } \right) / 2$	0.15	0
Signal diff. (L-R bursts)	$\frac{\sum_{t \in lastcycle} U_l(t) - U_r(t-P/2) }{\sum_{t \in lastcycle} U_l(t) + U_r(t-P/2) }$	0.4	0.05
L-R signal diff.	$\frac{\sum_{t=1}^n U_l(t) - U_r(t) }{n_max_connections}$	0	0.8
Connectivity		1	0.3
Frequency range (Hz)	-	0.07	0.33
Tonic input range	-	0	2
Lowest wave frequency	-	0.33	0.07
Highest wave frequency	-	0.07	0.33

Boundaries for acceptable outcomes are set using values in the columns labelled ‘good’ and ‘bad’ in table 1. These are transformed into individual objective scores with eqn. 5 and then combined into an overall fitness value for each candidate solution.

$$F(x) = 0.95 * (x - Good) / (Good - Bad) + 1 \quad (5)$$

6 Results and Discussion

Results demonstrate that the parameters of the invertebrate central pattern generator can be evolved using a genetic algorithm, to operate in a different solution space. A ‘sea wave’ configuration is presented in table 2 together with the original bio-network.

6.1 Evolved Neural Architecture

Weights, synaptic connections and neural parameters differ substantially between the two networks. Although the denominations of the neuron groups have been maintained, it should be noted that in the evolved wave-energy network, these no longer perform the same function as they did in the biological network (of fig. 2). Therefore, a completely new network, with different relationships between neuron groups has emerged.

Interestingly, evolving controllers operating in the required range was not a simple task. Of 20 experiments, each starting with a random population, only 5 generated controllers met the set objectives. However, within these 5 experiments, several viable controllers with different configurations were found. This

Table 2. Configuration and Operating Ranges of CPGs Evolved for Sea Waves

Frequency Range(Hz) low - high	Conns (of 56)	Neural Parameters			Time Delays		Synaptic Weights							
		θ	Γ	μ	τ_D	τ_A	from:	EIN1	CINI	LINI	EINr	CINr	LINr	BS
Biological Prototype														
1.74 - 5.56	26	-0.2 0.5 8.0 0.1	1.8 1.0 0.5 0.3	0.3 0.3 0 0	30 20 50 20	400	EIN1 CINI LINI MNI	0.4 3.0 13.0 1.0	- - - -	- -1.0 - -	- -2.0 -1.0 -2.0	- - - -	2.0 7.0 5.0 5.0	
An Evolved Sea Wave Controller														
0.03 - 0.39	16	2.27 -0.1 -0.45 -1	1.53 0 0.19 1.97	0.77 0.300 0.750 0.300	450 3000 - -	6000	EIN1 CINI LINI MNI	-0.9 - -0.55 -	- - -13 -	14 - - -3.47	-2.54 - - -4.36	- - - -	-5 -5 -0.13 4.36	

discovery supports the need to generate suitable controllers in a stage-wise rather than ‘all-in-one-go’ manner. In the wave device problem, the number of evolvable parameters continues to increase and fitness functions will become more and more complex, which in turn expands the search space of the GA, resources and time spent testing solutions. Therefore, finding viable networks which operate in the correct conditions is crucial, before coupling them to machines and exploring adaptation criteria.

A final observation from table 2 is that connectivity is lower (16 vs 26 connections) in the evolved network. This simpler configuration is vital for hardware implementation and fulfils an objective guiding the GA process.

6.2 ‘Sea-Wave’ Performance Range

The operating ranges of this ‘sea wave’ controller are 0.03 - 0.39Hz (33s to 2.5s wave periods), substantially surpassing the requirements of current wave energy converters (WECs). This means that control modules may not need updating even as 1) technology improves and WECs are built to withstand harsher sea conditions, 2) climate conditions change (as they are doing), entire systems and power cables may not need to be relocated.

Other evolved networks for the wave-power problem contained frequency ranges much lower than 0.03Hz and some higher than 0.39Hz. Therefore, fitness components of the genetic algorithm can be altered to accommodate specific operating ranges of individual types of marine energy device.

7 Conclusion

This work shows that a bio-inspired neural network (a fish’s central pattern generator which governs swimming) can be re-evolved to operate in a new task domain, marine energy. CPGs are used for their ability to self-regulate their

motion in response to changing conditions. For wave energy devices, such an attribute is extremely useful, providing a form of intelligent control to an otherwise unintelligent operator. We believe this active and adaptive approach will produce more power more reliably in variable conditions. Developing a controller to operate in these different conditions, as described in this paper, forms a crucial step towards achieving this. Producing bio-mimetic systems in this way also has significant implications for other real tasks and challenges in engineering/control.

Acknowledgments. This research is supported by EPSRC Grant EP/C516583/1 and previously by SHEFC grant R36298.

References

1. Finerva.com, <http://www.finavera.com/en/wavetech/>
2. National Aeronautics and Space Administration (NASA), Intelligent Flight Control System (2005), <http://www.nasa.gov/centers/dryden/news/FactSheets/FS-076-DFRC.html>
3. Arena, P., Fortuna, L., Frasca, M., Sicurella, G.: An Adaptive, Self-organizing Dynamical System for Hierarchical Control of Bio-inspired Locomotion. *IEEE Trans. Syst. Man Cybern. B Cybern.* 34(4), 1823–1837 (2004)
4. Pelamis Wave Power, <http://www.pelamiswave.com/>
5. Moskowitz, L.: Estimates of the Power Spectrum for Fully Developed Seas for Wind Speeds of 20 to 40 Knots. *Jour. of Geophysical Research* 69(24), 5161–5179 (1964)
6. Cohen, A.H., Wallén, P.: Fictive Swimming Induced in an in vitro Preparation of the Lamprey Spinal Cord. *Experimental Brain Research* 41(1), 11–18 (1980)
7. Grillner, S., McClellan, A.D., Sigvardt, K., Wallén, P., Wilén, M.: Activation of NMDA Receptors Elicits fictive locomotion. *Lamprey Spinal Cord in vitro, Acta Physiologica Scandinavica* 113, 549–551 (1981)
8. Ekeberg, Ö.: A Combined Neuronal and Mechanical Model of Fish Swimming. *Biological Cybernetics* 69, 363–374 (1993)
9. Patel, L.N., Murray, A., Hallam, J.: Increased Swimming Control with Evolved Lamprey CPG Controllers. In: Int. Joint Conf. on Neural Networks, pp. 2195–2200 (2005)
10. Patel, L.N., Murray, A., Hallam, J.: Super-Lampreys and Wave Energy: Optimised Control of Artificially-Evolved. Simulated Swimming Lamprey, *Neurocomputing* 70, 1139–1154 (2007)

Feature Subset Selection Using Differential Evolution

Rami N. Khushaba, Ahmed Al-Ani, and Adel Al-Jumaily

Faculty of Engineering and Information Technology, University of Technology, Sydney
(UTS), Australia
`{rkhushab, ahmed, adel}@eng.uts.edu.au`

Abstract. One of the fundamental motivations for feature selection is to overcome the curse of dimensionality. A novel feature selection algorithm is developed in this chapter based on a combination of Differential Evolution (DE) optimization technique and statistical feature distribution measures. The new algorithm, referred to as DEFS, utilizes the DE float number optimizer in a combinatorial optimization problem like feature selection. The proposed DEFS highly reduces the computational cost while at the same time proves to present a powerful performance. The DEFS is tested as a search procedure on different datasets with varying dimensionality. Practical results indicate the significance of the proposed DEFS in terms of solutions optimality and memory requirements.

1 Introduction

One of the most important and indispensable tasks in any pattern recognition system is to overcome the curse of dimensionality problem. This forms a motivation for using a suitable dimensionality reduction method including either feature selection or projection. Feature selection (FS) is essentially a task to remove irrelevant and/or redundant features [1]. In simple words, feature selection techniques study how to select a subset of attributes or variables that are used to construct models describing data. The reason behind using FS techniques include reducing dimensionality, removing irrelevant and redundant features, reducing the amount of data needed for learning, improving algorithms' predictive accuracy, and increasing the constructed models' comprehensibility [2].

The selection of features can be achieved in two ways: One is to rank features according to some criterion and select the top k features, and the other is to select a minimum subset of features without learning performance deterioration. As a part of any feature subset selection algorithm, there are several factors that need to be considered, the most important are: the evaluation measure and the search strategies [3]. The typical evaluation measures include: filters and wrappers. Filter based feature selection methods are in general faster than wrapper based methods. This is due to the fact that the filter based methods depend on some type of estimation of the importance of individual features or subset of features. Comparing with the filter methods, wrapper based methods

are more accurate as the quality of the selected features in a subset is measured using a learning algorithm. On the other hand, a search strategy is needed to explore the feature space. Various search algorithms that differ in their optimality and computational cost have been developed to search the solution space. These methods include for example: Tabu Search (TS) [4], Simulated Annealing (SA) [5], Genetic algorithms (GA) [6], Ant Colony Optimization (ACO) [7], and Particle Swarm Optimization (PSO) [8].

The purpose of this chapter is to present a novel feature selection algorithm based on a modification of Differential Evolution (DE) optimization technique, in combination with a simple, yet powerful, statistical measure. The justification behind the introduction of the new algorithm is that it highly reduces the computational memory requirements while at the same time achieve good results. The reason for considering DE is that it has proved to perform extremely well on a wide variety of test problems presenting a powerful performance in terms of solutions optimality and convergence speed, as stated by [9]. However, the original DE optimization algorithm is a float number optimizer not suitable to be used directly in a combinatorial problem like feature subset selection. Thus certain modifications are required for DE to be suitable for a feature subset selection task.

This chapter is structured as follows: Section 2 introduces the reader to the concept of Differential Evolution. Section 3 describes the proposed DE-based feature selection algorithm. Practical results are presented in section 4. Finally, a conclusion is given in section 5.

2 Differential Evolution and Feature Subset Selection

The most well known parallel search algorithms are GA, PSO, and DE algorithms. In feature selection problems with both GA and PSO, binary strings are employed usually in which every bit represents an attribute. The value of '1' means that the attribute is selected while '0' means not selected. This increases the computational cost for large problems. As an example consider a problem with 3000 features. If a subset of 20 feature is required, then for a population of 50 elements, the total size of the population matrix for both GA and PSO will be 50×3000 with each member of the population containing 20 1's and 2980 0's. On the other hand, DE was introduced to solve problems with real values. The DE optimization technique can be viewed as an enhanced version of the real valued GA that employs a differential mutation operator with faster convergence properties. Currently, there is no well known binary version of DE that can be used for feature selection. In order to develop a new DE based feature selection algorithm, then it is a tempting task to study the possibility of utilizing the float optimizer in feature selection problems. This would results in a great saving in computational cost. Thus, for the example mentioned earlier, the size of the population matrix will be 50×20 , hence, a lower memory requirement than both GA and PSO.

2.1 Differential Evolution

Differential Evolution (DE) is a simple optimization technique having parallel, direct search, easy to use, good convergence, and fast implementation properties [9]. The crucial idea behind DE is a new scheme for generating trial parameter vectors by adding the weighted difference vector between two population members X_{r1} and X_{r2} to a third member X_{r0} . The following equation shows how to combine three different, randomly chosen vectors to create a mutant vector, $V_{i,g}$ from the current generation g :

$$V_{i,g} = X_{r0,g} + F \times (X_{r1,g} - X_{r2,g}) \quad (1)$$

where $F \in (0, 1)$ is a scale factor that controls the rate at which the population evolves.

Extracting both distance and direction information from the population to generate random deviations results in an adaptive scheme that has good convergence properties. In addition, DE also employs uniform crossover, also known as discrete recombination, in order to build trial vectors out of parameter values that have been copied from two different vectors. In particular, DE crosses each vector with a mutant vector, as given in Eq. (2):

$$U_{j,i,g} = \begin{cases} V_{j,i,g} & \text{if } \text{rand}(0,1) \leq C_r \text{ or} \\ X_{j,i,g} & \text{Otherwise} \end{cases} \quad (2)$$

where $U_{j,i,g}$ is the j 'th trial vector along i 'th dimension from the current population g . The crossover probability $C_r \in [0, 1]$ is a user defined value that controls the fraction of parameter values that are copied from the mutant. If the newly generated vector results in a lower objective function value (better fitness) than the predetermined population member, then the resulting vector replaces the vector with which it was compared [10].

To this end, DE optimization may not be used directly in feature selection problems. The next section identifies a modified DE to be utilized in feature selection problems.

3 A Float-Optimizer for Feature Selection

A new algorithm is presented in this chapter as one possible way to utilize a float number optimizer like DE in feature selection. The block diagram of the proposed DEFS algorithm is shown in Fig.1. The first step in the algorithm is to generate new population vectors from the original population. For each position in the population matrix, a new mutant vector is formed as mentioned in Eq.1. The mutant vector is then crossed with the original vector that occupies that position in the original matrix. The result of this operation is called a trial vector. The corresponding position in the new population will contain either the trial vector (or its corrected version) or the original target vector depending on which one of them achieved a higher fitness (classification accuracy).

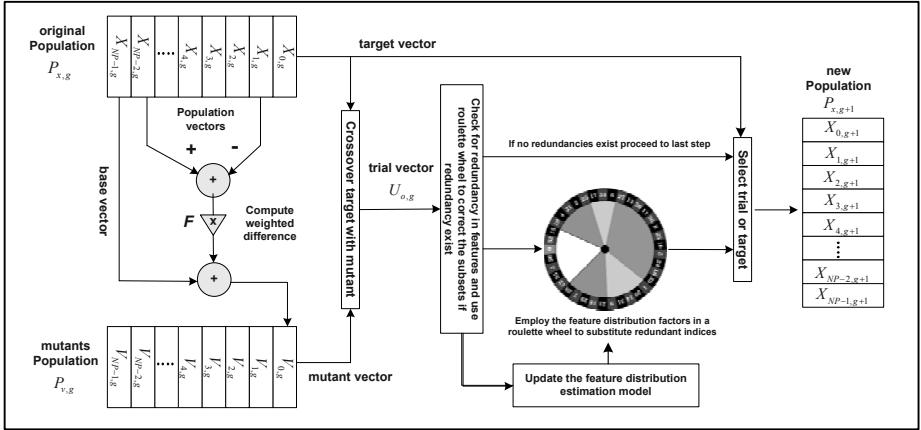


Fig. 1. Block diagram of the proposed DEFS algorithm

Due to the fact that a real number optimizer is being used, nothing will prevent two dimensions from settling at the same feature coordinates. As an example, if the resultant vector is [255.1325 20.5498 85.5464 43.4240 254.8665], then the rounded value of the resulting vector would be [255 21 86 43 255]. This result is completely unacceptable within feature selection problems, as a certain feature (feature index = 255) is used twice. In order to overcome such a problem, we propose to employ feature distribution factors to replace duplicated features. A roulette wheel weighting scheme is utilized. In this scheme a cost weighting is implemented in which the probabilities of individual features are calculated from the distribution factors associated with each feature. The distribution factor of feature f_i is given by Equ.3 below:

$$FD_i = a_1 \times \left(\frac{PD_i}{PD_i + ND_i} \right) + \frac{NF - DNF}{NF} \times \left(1 - \frac{(PD_i + ND_i)}{\max(PD_i + ND_i)} \right) \quad (3)$$

where a_1 is a suitably chosen positive constant that reflects the importance of features in PD_i . PD_i is the positive distribution factor that is computed from the subsets which achieved an accuracy that is higher than the average accuracy of the whole subsets. ND_i is the negative distribution factor that is computed from the subsets which achieved an accuracy that is lower than the average accuracy of the whole subsets. NF is the total number of feature, and DNF is the desired number of features to be selected. PD_i and ND_i are shown schematically in Fig. 2. The rationale behind Eq. 3 is to replace the replicated parts of the trial vectors according to two factors. The $PD_i/(PD_i + ND_i)$ factor indicates the degree to which f_i contributes in forming good subsets. On the other hand the second term in Eq. 3 aims at favoring exploration, where this term will be close to 1 if the overall usage of a specific feature is very low. Meanwhile, the $((NF - DNF)/NF)$ is utilized as a variable weighting factor for

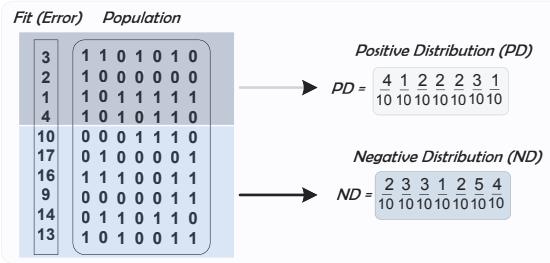


Fig. 2. The feature distribution factors

the second term. In such a case the importance of unseen features will be higher when selecting smaller number of features and smaller when selecting a larger number of features.

The distribution factors supplied to the roulette wheel is formed by the difference between the relative frequencies of distribution estimated from FD_g and FD_{g+1} . In such a case one would reduce the possibility of certain features dominating the estimation of the distribution factor.

As an example on this algorithm, consider the same example with redundancies above. The aim here is to correct the current trail vector [255 21 86 43 255] and replace duplicated feature with another one that is most relevant to the problem. Let's presume that the features ranked by the roulette wheel according to the highest distribution factors are [55, 255, 21, 210, 68, 74]. After excluding features that appear in the trial vector, the rest can be used to replace the duplicated features of the trail vector. Thus for our example, the trial vector would be represented by [255 21 86 43 55].

4 Experimental Results

In this section, the proposed DEFS will be compared with the well established binary GA [11], and the Binary PSO (BPSO) [12]. The implementations details for both GA, and BPSO are given as: both of these techniques utilize binary strings representing a feature subset with ones and zeros to indicated the selection and neglection of features respectively. A traditional GA was used with probability of crossover = 0.5 (chosen after running several tests), and probability of mutation = 0.02. The obtained strings are constrained to have the number of '1's matching a predefined number of desired features. The parameters of BPSO are selected as: the inertia weight was made to decrease linearly from 0.9 to 0.4; the maximum velocity was set to be clipped within 20% of the corresponding variable; and acceleration constants c_1 and c_2 are set to 2.0.

All of the DEFS, GA and BPSO were made to start from the same initial population and iterate to the same number of iterations. For all experiments described below, a population size of 50 was used by the three feature selection methods. The chosen fitness function was set to the classification error rates as

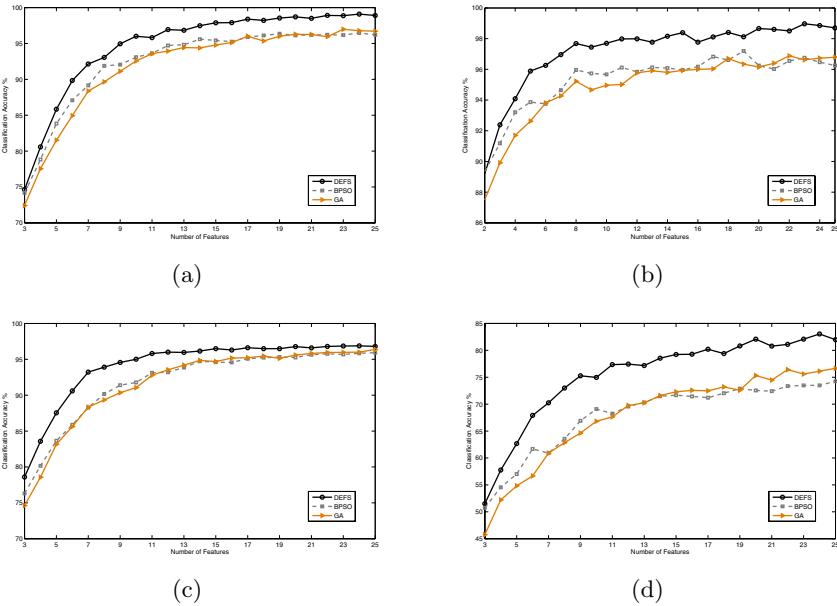


Fig. 3. Average classification accuracies across thirty runs for each subset size (a) Lung dataset (b) Colon dataset (c) Lymphoma dataset (d) NCI dataset

achieved by a suitable classifier. Four different datasets with varying dimensionality are utilized to check the performance of the proposed DEFS. These are available online from <http://research.janelia.org/peng/proj/mRMR/>, and the details for these datasets are given as: Lung Cancer (325 features), Colon (2000 features), Lymphoma (4026 features), and NCI (9712 features).

Due to the small number of samples associated with the above mentioned datasets then a 10 fold cross validation technique was used during the experiments on these datasets. Since the appropriate size of the most predictive feature set is unknown, the proposed DEFS was employed for various feature set sizes ranging from 3 to 25 with a step of 1. The methods were employed for thirty runs when searching for each specific feature subset size and the average is reported here. For example, when searching for a feature subset of size 9, each method (DEFS, BPSO, and GA) was utilized for thirty times to search for a subset of 9 features. The average of these thirty runs is reported as the classification accuracy in this chapter. The type of classifier employed with these datasets was chosen to be the K Nearest Neighbor classifier (KNN) with $K = 5$. The classifier type was fixed for the all of the search procedure (DEFS, BPSO, and GA).

The classification results for these five datasets are shown in Fig. 3. The results prove the effectiveness of the proposed DEFS in searching datasets with different number of features for the subsets that best interact together. Although that all of the methods starts from the same initial population, but the DEFS

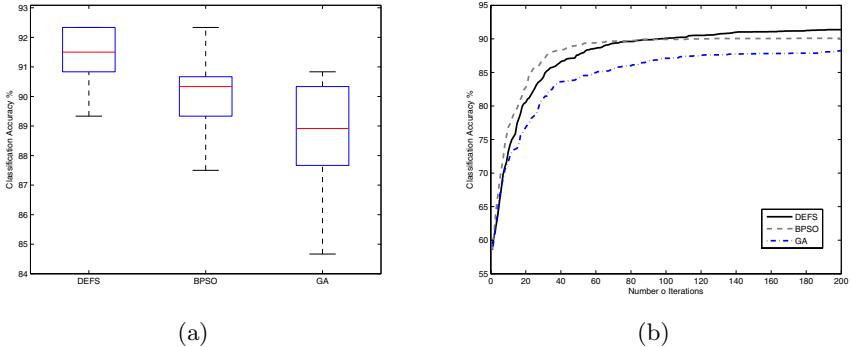


Fig. 4. Average classification accuracies across thirty runs for the Madelon dataset (a) Boxplot of the classification accuracies (b) Convergence of different algorithms for this specific dataset

performance outperforms the other methods due to the continuous exploration ability of the DEFS provided by the statistical probability based measure. In addition to the enhanced performance offered by the DEFS algorithm, the memory requirements of the DEFS algorithm are also much less than that of BPSO and GA. As an example consider the NCI dataset, when one try to select a feature subset size of say 10 features, both BPSO and GA will utilize a population of size 50×9712 , while the DEFS population size will only be 50×10 , i.e., a huge reduction in memory requirements, while maintaining better performance as proved by the practical results.

Due to the fact that distribution based algorithms which employs probability vectors usually allow only a very limited representation of dependencies between features, then it is a tempting task to validate the performance of DEFS on datasets with a large degree of dependency between features. For this task, the Madelon dataset from the UCI repository was chosen. It is a two-class classification problem with sparse binary input features. There are 500 features in this dataset from which only 5 are useful and the rest are either redundant or irrelevant. The original contributors of this dataset subdivided this dataset into a training set with 2000 patterns, and 600 patterns for validation. In this experiment also a KNN classifier was employed with $K = 5$. The experiments were repeated for 30 times also for each of the DEFS, BPSO, and GA. The classification accuracies achieved by DEFS, BPSO, and GA are illustrated using a boxplot as shown in Fig. 4a. The convergence of these algorithms is also shown in Fig. 4b. The figures shows that DEFS has a continuous exploration capability that leads toward the vicinity of the most promising solutions. This is mainly contributed to by the two terms in Eq. 3. This also proves that GA has the highest fluctuations among the three methods and it probably means that its performance is sensitive to the initial population when having large feature sets.

5 Conclusion

A new feature selection method was presented in this chapter based on the Differential Evolution optimization technique. The performance of the proposed algorithm was compared with GA and PSO. It was shown that the proposed DEFS required smaller memory than other methods which yields a reduction in the computational cost. Also, when testing on different datasets with huge number of features, the proposed algorithm managed to outperform both GA and PSO in terms of classification performance yielding higher accuracies across different number of features. All of the results presented proved the effectiveness of the proposed DEFS algorithm.

References

1. Liu, H., Motoda, H.: Computational Methods of Feature Selection. Taylor & Francis Group, LLC, Abington (2008)
2. Liu, H., Dougherty, E.R., Dy, J.G., Torkkola, K.A., Tuv, E., Peng, H.A., Ding, C.A., Long, F.A., Berens, M.A., Parsons, L.A., Zhao, Z.A., Yu, L.A., FOrman, G.A.: Evolving feature selection. *IEEE Intelligent Systems* 20, 64–76 (2005)
3. Al-Ani, A.: Feature Subset Selection Using Ant Colony Optimization. *Int. Journal of Computational Intelligence* 2, 53–58 (2005)
4. Tahir, M.A., Bouridane, A., Kurugollu, F., Amira, A.: Feature Selection Using Tabu Search for Improving the Classification Rate of Prostate Needle Biopsies. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), vol. 2, pp. 335–338 (2004)
5. Filippone, M., Masulli, F., Rovetta, S.: Supervised Classification and Gene Selection Using Simulated Annealing. In: International Joint Conference on Neural Networks, IJCNN 2006, pp. 3566–3571 (2006)
6. Frohlich, H., Scholkopf, B.: Feature Selection for Support Vector Machines by Means of Genetic Algorithms. In: 15th IEEE International Conference on Tools with Artificial Intelligence, November 3-5, pp. 142–148 (2003)
7. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, London (2004)
8. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, London (2001)
9. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Springer, Heidelberg (2005)
10. Palit, A.K., Popovic, D.: Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications. Springer, Heidelberg (2005)
11. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms, 2nd edn. John Wiley & Sons, Chichester (2004)
12. Firpi, H.A., Goodman, E.: Swarmed Feature Selection. In: Proceedings of the 33rd Applied Imagery Pattern Recognition Workshop (AIPR 2004), pp. 112–118 (2004)

Topology of Brain Functional Networks: Towards the Role of Genes

Mária Markošová¹, Liz Franz², and Ľubica Beňušková³

¹ Department of Applied Informatics, Faculty of Mathematics,
Physics and Informatics, Comenius University, Bratislava, Slovakia

² Department of Psychology

³ Department of Computer Science,
University of Otago, Dunedin, New Zealand

lubica@cs.otago.ac.nz

Abstract. We have extracted brain functional networks from fMRI data based on temporal correlations of voxel activities during the rest and task periods. The goal of our preliminary research was to study the topology of these networks in terms of small-world and scale-free properties. The small-world property was quite clearly evident whereas the scale-free character was less obvious, especially in the rest condition. In addition, there were some differences between the rest and task functional brain networks as well as between subjects. We discuss the relation of properties of functional brain networks to the topological properties of the underlying anatomical networks, which are largely dependent upon genetic instructions during brain development.

1 Introduction

Functional magnetic resonance imaging (fMRI) is a powerful noninvasive technique of brain activity imaging [1]. A measure of brain activity is the blood oxygenation level dependent (BOLD) signal recorded sequentially in slices of thousands of voxels of $\sim 3 \text{ mm}^3$ over the whole brain within the interval of 2–3 sec. In addition to valuable information on localization of various brain functions, it is possible to analyze the data to seek the underlying *functional brain networks*, i.e. networks of functional units that temporarily self-organize themselves to engage in a given task or to engage in spontaneous background activity during the rest condition.

In general, a network is a set of nodes linked by edges, oriented or not, mathematically described as a directed or undirected graph, respectively. Recently, Chialvo with co workers [2] [3] introduced the concepts from the graph theory to study the topological structure of functional brain networks. Topology is not concerned with metric properties such as physical distances between nodes, instead, topology involves the study of properties that describe how the nodes are assembled in space through their interconnections.

Basic concepts from the graph theory used to describe the network topology are the degree k , clustering coefficient C , average shortest path L and degree

distribution $P(k)$ [4] [5] [6]. The degree k of a node is the sum of all its efferent and afferent connections. Connected nodes are called neighbors in spite of their actual separation in physical distance. The clustering coefficient C is the average of actual existing connections between the node and its neighbors divided by the number of all possible connections the node could have with all the nodes in the network. Average shortest path L is the global mean of all the shortest distances between pairs of all the nodes in the network. The degree distribution $P(k)$ is the probability distribution of a node having the degree k . A *randomly connected network* has a binomial distribution with a single peak value of particular k (single-scale). The *scale-free network*, on the other hand, has a power-law degree distribution, i.e. $P(k) \sim k^{-\gamma}$, where γ is the power law scaling exponent. The scale-free network simply means that degrees are not grouped around one characteristic average degree (scale) but instead they are spread over a very wide range of values that can span several orders of magnitude. This implies a potentially rich internal organization with the so-called hubs that have many connections. Another possibility is an exponential degree distribution where higher degrees are less probable, which means the hubs are not so favoured in the internal organization. The *small-world networks* are characterized by high level of clustering, $C \gg C_{random}$, while the average short path is small, i.e. $L \approx L_{random}$ [7]. The canonical small-world network is a conjunction of many local clusters with occasional global interactions.

Eguíluz et al. [3] were interested in answering the question whether brain functional networks have small-world and scale-free properties. But how to construct a functional brain network from the fMRI data? They adopted voxels as natural candidates for nodes. Edges were defined as follows [3]: if the temporal BOLD activity in two voxels i and j is well correlated, then the link is established between them. For instance we can calculate the linear correlation coefficient as

$$r(i, j) = \frac{\langle V(i, t)V(j, t) \rangle - \langle V(i, t) \rangle \langle V(j, t) \rangle}{\sigma(V(i))\sigma(V(j))} \quad (1)$$

where $V(m, t)$ for $m = i, j$ is the activity in the voxel m at time t , $\sigma^2(V(m)) = \langle V(m, t)^2 \rangle - \langle V(m, t) \rangle^2$ and $\langle . \rangle$ represents the time average. If the temporal correlation exceeds some prescribed threshold value r_c , then the link between voxels i and j is established.

For their analysis Eguíluz et al. [3] seem to have used raw fMRI data as they do not mention any kind of preprocessing. Similar results were obtained for different tasks, namely a simple finger 1 and 2 opposing task, finger tapping and listening to music (although the temporal and other details of these protocols are not described). Network statistics did not significantly vary between 7 subjects nor for any of the total of 22 functional networks. The functional networks for $r_c = 0.5, 0.6, 0.7$ all had a clear scale-free degree distribution with average $\gamma = 2, 2.1, 2.2$, respectively. The functional networks also had a small-world property as the average scaling coefficient $C \sim 0.15 \gg C_{random} \sim 10^{-4}$, while $L \approx L_{random}$.

A more recent similar study was aimed at the topological statistics of the brain resting-state functional networks [8]. Van den Heuvel et al. focused on the observed low frequency ($\sim 0.1 - 1$ Hz) oscillations of the resting state BOLD signals, which are correlated between anatomically separated brain regions. They were interested whether these correlations result from physiological processes like breathing and cardiac oscillations or from the brain neural activity itself. Therefore they subjected their fMRI data from 28 subjects to an extensive pre-processing designed to remove all kinds of artifacts and to achieve the best possible temporal alignment between voxels. Their functional networks for threshold values $T = 0.55, 0.6, 0.65$ all had a clear scale-free degree distribution with average $\gamma = 1.8, 2.2, 2.5$, respectively. The resting-state functional networks were also of a small-world nature.

The goal of the presented preliminary study was to reproduce these procedures to verify whether the same statistics hold for the fMRI data from our laboratory. In addition, we wanted to take a closer look at differences between the rest and task functional brain networks as well as between inter-subject differences. The last objective was to seek explanation for our results and results from other laboratories as well as to set goals that should be investigated next.

2 Materials and Methods

2.1 Subjects and Data Acquisition

The fMRI data came from an experimental series of a sample database associated with fMRIotago, Dunedin, New Zealand. Normal healthy subjects were two females (subjects 1 and 3, aged 55 and 46 years, respectively) and two males (subject 2 and 4, aged 60 and 23 years, respectively). We used 16 seconds of initial rest, followed by 6 cycles of the task (20 s) and rest (20 s), with the first shorter rest period excluded from analysis. The task was a bimanual finger tapping, in which subjects tap once per second using thumb-index finger opposition movements of the left and right hands together according to a 1 Hz tone. The tone remains on for task and rest conditions with a change in pitch indicating task or rest condition. Each rest/task epoch contains 32,728 voxels [($Z = 1$ to 8 slices) \times ($X = 64$) \times ($Y = 64$)], with 10 acquisition cycles and 2 sec TR (repetition time).

2.2 Data Analysis

First, we averaged the rest data and task data for each subject, in order to emphasize the signal features that are characteristic for the rest and task condition of each subject. In both conditions, the absolute range of raw BOLD signal was in the interval from 0 to ~ 500 units. The signal differences between rest and task were only $\sim \pm 5\%$. Thus the activity of many voxels may actually on average decrease during the task relatively to the rest condition. Therefore we were interested in positive as well as negative suprathreshold correlations between the voxels. Then we calculated the correlation coefficients for 80 million randomly

chosen pairs of voxels. We took into account only those voxels whose activity was more than 100 or 200. As there was not a significant difference in results between these two activity thresholds, we did not raise this value further. When we did not apply this activity threshold, resulting functional networks had completely random structures. This is different from the study of Eguíluz et al. [3] who did not report any activity threshold; rather, they let the correlation threshold exclude the noisy random interactions. Another difference is they considered only positive interactions into their functional networks.

In our analysis, if the value of correlation coefficient is more than $r_c = 0.8$ (correlation), or less than $r_c = -0.8$ (anticorrelation) we consider the voxels significantly correlated and thus functionally connected. This way we created functional brain networks for all subjects. To extract the topology statistics of resulting functional networks we used the freely available Network Workbench Tool [9].

3 Results

Summarized results of the brain functional network analysis are shown in Table 1 (for activity threshold 100) and Table 2 (for activity threshold 200).

First let us look at the small world property. We have calculated $C_{random} = \langle k \rangle / N$, where $\langle k \rangle$ is the average degree of the network and N is the number of network nodes. $C_{random} = 10^{-3} - 10^{-4}$. For the estimate of the average shortest path in the corresponding random nets we used the formula $L_{random} \sim \ln(N) / \ln(\langle k \rangle)$, with the result of $L_{random} \approx 4$. The small-world index for all our networks is thus high, i.e. $\sigma = (C/C_{random})/(L/L_{random}) \sim 10$ to 100 [10].

With respect to the scale-free property, we can see that in all subjects and in both task and rest conditions, the functional networks have $1 < \gamma < 2.5$, which indicates a scale-free property. In addition, although we did not do any statistical evaluation, it does not seem there is any difference between the values of γ for the task and rest condition in individual subjects. However, there seems to be a difference between the subjects namely between the sample of subjects 1&4 ($\gamma \approx 2$) and 2&3 ($\gamma \approx 1.5$). To support this observation, however, we would have to collect more rest/task periods and perform the proper statistical evaluation. Nevertheless, this preliminary observation suggests there might be

Table 1. Activity threshold 100, correlation coefficient threshold $r_c = 0.8$. N = number of nodes, C = clustering coefficient, L = average shortest path, γ = power law scaling exponent

Subject	N	C		L		γ	
		task	rest	task	rest	task	rest
1	5530	0.038	0.038	5.12	5.10	1.99	2.04
2	5415	0.047	0.047	4.50	4.49	1.44	1.11
3	5720	0.047	0.047	4.31	4.04	1.39	1.39
4	5535	0.038	0.036	5.12	5.13	2.26	2.23

Table 2. Activity threshold 200, correlation coefficient threshold $r_c = 0.8$

Subject	N	C		L		γ	
		task	rest	task	rest	task	rest
1	4260	0.036	0.038	5.41	5.42	2.16	2.08
2	4420	0.047	0.047	4.74	4.79	1.54	1.33
3	5030	0.047	0.047	4.37	4.12	1.36	1.34
4	4265	0.040	0.038	5.41	5.40	2.05	2.38

individual differences between subjects, which may deserve further exploration in the future.

Surprisingly, the values of γ did not differ between the rest and task conditions, which may be related to the fact that finger tapping may not be such a demanding brain task. This hypothesis could be tested by performing analysis of fMRI data from subjects performing more cognitively demanding tasks. However, Eguíluz et al. [3] did not report any difference between the topology of for instance finger tapping and listening to music.

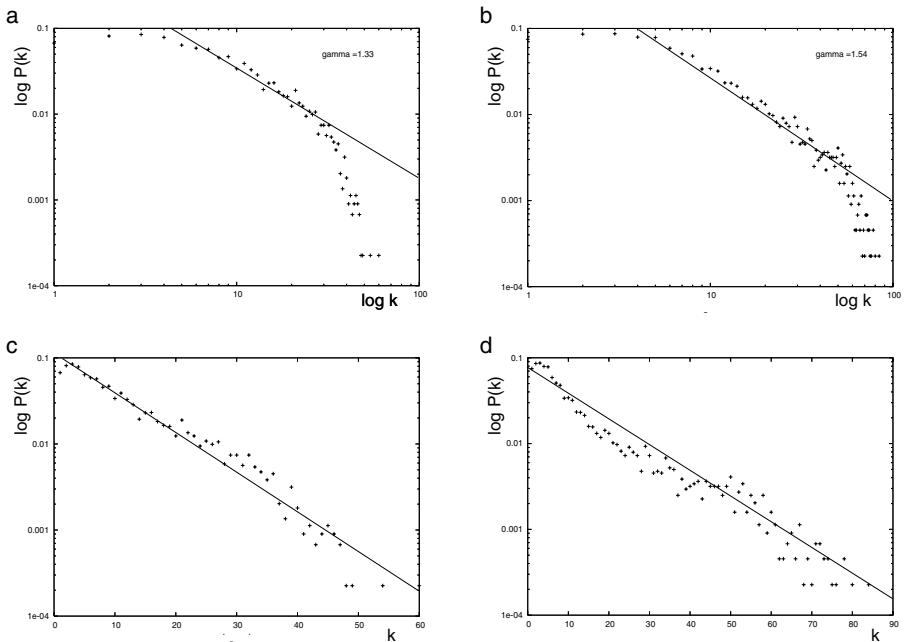


Fig. 1. Degree distributions for the averaged rest and task data in the functional brain networks of subject 2. (a) log–log plot for rest, (b) log–log plot for task, (c) log–lin plot for rest, (d) log–lin plot for task. We can see that the results are inconclusive with respect to the scale-free property – the rest network and perhaps also the task network may have rather an exponential degree distribution indicated by the linear fit in the log–lin plots.

Closer inspection of degree distribution plots reveals the most significant feature, common to all four subjects (we show only the plots for subject 2). There is a difference between the shape of the degree distribution for averaged rest and task periods. Scale-free structure, reflected by the linear part of the log–log plot of these distributions is more pronounced for the task period, see Fig. 1b. This means, that during the simple motor task of finger tapping, the functional brain network has more pronounced scale-free structure, i.e. has more high degree nodes than during rest. More high degree nodes for the task condition are clearly visible in the log–lin plots (compare Fig. 1c and 1d). In fact, for the rest period, the linear part of the log–log distribution was usually too short to calculate the scaling exponent (Fig. 1a); nevertheless we calculated them as well. For comparison we plotted the log–lin plots for both rest and task degree distributions (Fig. 1c,d). Despite these distributions cover many degrees (up to 50 for rest and up to 80 for task) they are roughly linear in the log–lin scale, especially the rest degree distribution. Linearity in the log–lin scale indicates the exponential degree distribution, and not the scale-free distribution. More detailed studies are necessary to decide whether the short linear part in the log–log plot is due to the lack of network scale-free structure, or due to the fact, that the analyzed functional brain networks did not have enough nodes or they indeed do not have a scale-free structure.

4 Discussion

In this work we studied functional brain networks of four healthy subjects. All of them were measured in two different conditions: during rest and during bimanual finger tapping task. All task and rest functional networks had the small-world property. However, results with respect to the scale-free property were not conclusive. Our results indicate that the functional networks may be on the edge between the exponential and scale-free degree distribution, with a slight shift towards the scale-free quality for the functional networks during the task period, and towards exponential distribution during the rest period.

The natural candidate where to look when trying to explain the topological structure of the functional brain networks is the topology of the underlying anatomical connectivity of the brain. Recently a new technique based on the diffusion MRI was used to generate large, comprehensive white matter connectional datasets of the entire brain for two human subjects [11]. Authors have found that global organization of individual human brain networks has the form of a small world. However, structural connectivity had an exponential degree distribution rather than a power-law distribution. That means hubs with many connections are not favoured. This finding is surprising in the light of previous findings that the functional networks have the power-law distribution of degrees [3] [8]. Our results are not so conclusive.

As Sporns et al. [12] argue, theoretical growth models for the scale-free networks are not biologically plausible, because they ignore the physical spatial relationships between nodes. For instance, $\gamma = 3$ corresponds to the Barabasi

and Albert model, where the network grows according to a preferential node attachment scenario [4] [6]: each new node is connected to n of the existing nodes one by one with a probability that is biased so that it is proportional to the number of links that the existing node already has. Additional local processes of edge rewiring cause the scaling exponent γ to be lower than 3. For instance, according to the Dorogovtsev and Mendes model [5], in addition to the preferential node attachment scenario, at each time step new edges are created between old nodes, with probability of this rewiring being proportional to the number of links that the existing nodes already have. This model results in a degree distribution in which $\gamma = 1.5$ for small values of k and $\gamma = 3$ for larger values of k . Markošová [13] suggested yet another local process of rewiring old nodes, namely that in addition to the step introduced by Dorogovtsev and Mendes, a certain proportion of old nodes is selected whose one existing edge end is detached and rewired again in a preferential manner. Depending on a particular proportion of re-wired nodes, we get $2 \leq \gamma < 3$ for larger values of k . On the other hand, Kaiser and Hilgetag modeled the development of frontal macaque cortex by a spatially embedded growing graph where preferential attachment occurs as an exponentially decaying function of spatial distance [14]. Amaral et al. modeled network growth where the node degree expansion is attenuated through node aging and energy limitations [15]. These two models resulted in networks with an exponentially decaying degree distribution. To summarize, it seems that the relationship between the underlying patterns of brain connectivity and emerging functional networks is not linear and must be studied further.

First, the effect of various types of fMRI data preprocessing upon resulting topology of functional networks must be investigated. An important task will be to label voxels and study the organization of these functional networks with respect to anatomy. This latter investigation can shed more light on the structural-functional relationships. It would be interesting to explore how the structure of functional brain network changes with different cognitive tasks. Will the difficulty or nature of task be reflected in the network parameters? Another interesting question is the influence of various neurological and psychiatric conditions upon parameters of brain functional networks. Will they change or remain invariant?

Specificity and precision required to achieve the correct brain wiring is extraordinary. Axons have to grow long distances bypassing numerous neurons on the way, before reaching and forming synapses on their proper targets. Some axons that grew early when the distances were short serve as scaffolds. Other axons grow along epithelial surfaces or extracellular matrices. Guidepost cells mark important sites. Growing axons respond to various external molecules with the help of an array of various receptors. And all this is orchestrated by the spatio-temporal patterns of gene expressions in the developing brain. The early connections provide the foundation of the adult structure, and disruption of these may be the source of many developmental flaws, such as cerebral cortical developmental disorders (including schizophrenia and autism) [16]. Thus, the future line of research in modeling brain functional networks can be based on the new

modeling approach of the so-called computational neurogenetic modeling, which models complex dynamics of neural networks with the help of the underlying dynamics of genes [17].

Acknowledgments. This research was supported by the Department of Computer Science, University of Otago, New Zealand, and by VEGA 1/2045/05, 1/3612/06 and 2/7087/27.

References

1. Functional magnetic resonance imaging (fMRI),
<http://en.wikipedia.org/wiki/FMRI>
2. Chialvo, D.R.: Critical brain networks. *Physica A* 340(4), 756 (2004)
3. Eguíluz, V., Chialvo, D.R., Cecchi, G.A., Baliki, M., Apkarian, A.V.: Scale-free brain functional networks. *Phys. Rev. Letters* 92, 018102 (2005)
4. Barabási, A.L., Albert, R.: Emergence of scaling in random network. *Science* 286, 509 (1999)
5. Dorogovtsev, S.N., Mendes, J.F.F.: Evolution of networks. *Adv. Phys.* 51, 1079 (2002)
6. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Rev. Modern Phys.* 74, 47 (2002)
7. Watts, D.J.: *Small Worlds*. Princeton University Press, Princeton (2004)
8. van den Heuvel, M.P., Stam, C.J., Boersma, M., Hulshoff Pol, H.E.: Small-world and scale-free organization of voxel-based resting-state functional connectivity in the human brain. *NeuroImage* (2005), doi:10.1016/j.neuroimage.2008.08.010
9. NWB Team, Network Workbench Tool, Indiana University, Northeastern University and University of Michigan, <http://nwb.slis.indiana.edu>
10. Humphries, M.D., Gurney, K., Prescott, T.J.: The brainstem reticular formation is a small-world, not scale-free, network. *Proc. Biol. Sci.* 273, 503–511 (2006)
11. Hagmann, P., Kurant, M., Gigandet, X., Thiran, P., Van Wedeen, J., Meuli, R., Thiran, J.-P.: Mapping human whole-brain structural networks with diffusion MRI. *PLoS ONE* 2(7), e597, doi:10.1371/journal.pone.0000597
12. Sporns, O., Chialvo, D., Kaiser, M., Hilgetag, C.C.: Organization, development and function of complex brain networks. *Trends Cog. Sci.* 8, 1364–6613 (2004)
13. Markošová, M.: Network model of human language. *Physica A* 387, 661 (2008)
14. Kaiser, M., Hilgetag, C.C.: Modelling the development of cortical systems networks. *Neurocomputing* 58–60, 297–302 (2004)
15. Amaral, L., Scala, A., Barthélémy, M., Stanley, H.: Classes of small-world networks. *Proc. Natl. Acad. Sci. USA* 97, 11149–11152 (2000)
16. Molnár, Z., Hoerder-Suabedissen, A., Wang, W.Z., DeProto, J., Davies, K., Lee, S., Jacobs, E.C., Campagnoni, A.T., Paulsen, O.: Genes involved in the formation of the earliest cortical circuits. In: Novartis Found Symp., vol. 288, pp. 212–224 (2007)
17. Benuskova, L., Kasabov, N.: Computational Neurogenetic Modeling. Springer, New York (2007)

Hybrid Design Principles and Time Constants in the Construction of Brain-Based Robotics: A Real-Time Simulator of Oscillatory Neural Networks Interacting with the Real Environment via Robotic Devices

Hiroaki Wagatsuma

Laboratory for Dynamics of Emergent Intelligence, RIKEN BSI,
2-1 Hirosawa, Wako-shi, Saitama
waga@brain.riken.jp

Abstract. One of most important concepts in robotics and artificial intelligence is the embodied approach, focusing on the importance of having a body that functionally connects to the external world. This setup suggests that the intelligence develops through sensorimotor skills and through situations that would actually be confronted in the environment. We support this concept and propose to further extend it to embodiment in the time domain. Nervous systems have variable processing times. The different time courses proceed in the nervous system in parallel, and individual circuits independently and cooperatively work under the constraints of temporal properties. We here propose an experimental platform of oscillatory neural networks having real-time communication with the environment through the robot's body. The synchronization mechanism of oscillations in neural activities have the advantage of synthetic controls known in motor coordination, but we extend this to circuits for cognitive functions like episodic memory formation and decision making of the robotic behavior by using the theta phase coding mechanism. A slow oscillation, like the theta rhythm, enables behavioral temporal sequences to be compressed in sequential firings during each oscillation cycle, and this helps to represent cognitive information in episodes composed of past-present-future structures. The temporal structure is crucial for recognition of the current context and adaptability in dynamic environments, and it smoothly controls sensorimotor local circuits with faster time scales. This work represents a tiny step towards constructing the brain by focusing on the temporal structure, yet this approach may elucidate the new nature of the brain-based intelligence.

1 Introduction

The issue of what intelligence is within the context of biological systems has long been a hot topic in the research field of artificial intelligence and the subject of many philosophical theories. Recently, it became a concrete target of the computational neuroscience field, especially in the synthetic approaches to brain-based robotics, enabling the combination of multiple functional units, like independent brain regions,

that function in the real world [1]. Such approaches for constructing a robot with a whole brain system with higher cognitive functions must still answer the question of what kinds of intelligence the brain possesses. Similar questions are frequently asked in the form of the ‘frame problem’ in the field of artificial intelligence [2]. The problem is how the brain quickly dissociates the necessary information for the current situation from bulk information in memory. However, the further question is whether this problem can be solved in the framework of simple cascade processes in a uniform time step. From the viewpoint of brain function for the figure-ground separation of ambiguous pictures or illusions, the coordination of the timing of ongoing activities in the system is necessary. In theoretical models [3,4], relevant features can be bound by the neural oscillation in local circuits, and the central circuit receives fragments of pictures and naturally modulates the local circuits to represent a possible picture by using oscillatory coupling with a slow change rather than the oscillation cycle. The advantage of a synchronization mechanism of oscillations in neural activities is well known not only in motor coordination, but also for other cognitive functions [5]. Recent evidence supports the functional roles of brain oscillation, by showing a long-range synchrony in the gamma-band oscillation distributed over the whole scalp of the human brain [6] and a long-range theta-band oscillation for the execution of mental calculations [7]. The interesting point of this evidence is that the brain has a variety of oscillations at different time scales, which may contribute to the integration of different types of information in the distributed brain areas. One plausible hypothesis to understand their functional role is that multiple brain oscillations underlie the intelligence of the brain, or at least how its speed. This is the key to thinking about parallel information processing in the brain.

To test this hypothesis, we considered the implementation of the neural mechanism with oscillator synchronizations [8,9,10] to a real robot and then constructed a simulation system to connect the robot working in the real environment as an on-line and real-time system [11]. It naturally has the physical constraint of the computer power for the simulation system. For example, the more equations are implemented, the more performance is required. In the theoretical studies, the neural oscillation is described by differential equations and conventionally simulated in the off-line system with numerical integration methods. We also used the same method but applied it to the on-line system. The serious problem is interruption of ongoing simulation by communication protocols with the robot. Therefore, a hybrid system between programming codes of the protocols and the mathematical description of the neural oscillators simulated by the numerical integration was adopted. We simply tested the online robot system via the sensorimotor skill of wall avoidance and a navigational task in the cross maze. Section 2 describes the mathematical description of the neural oscillation. Section 3 describes its implementation to the online robot system. Section 4 briefly introduces the experimental data. Finally, section 5 summarizes our research approach of brain-based robotics.

2 Relevant Neural Mechanisms

The ultimate goal is to make an online system to mimic the whole brain system, which consists of principal functional components that cooperatively act together. For

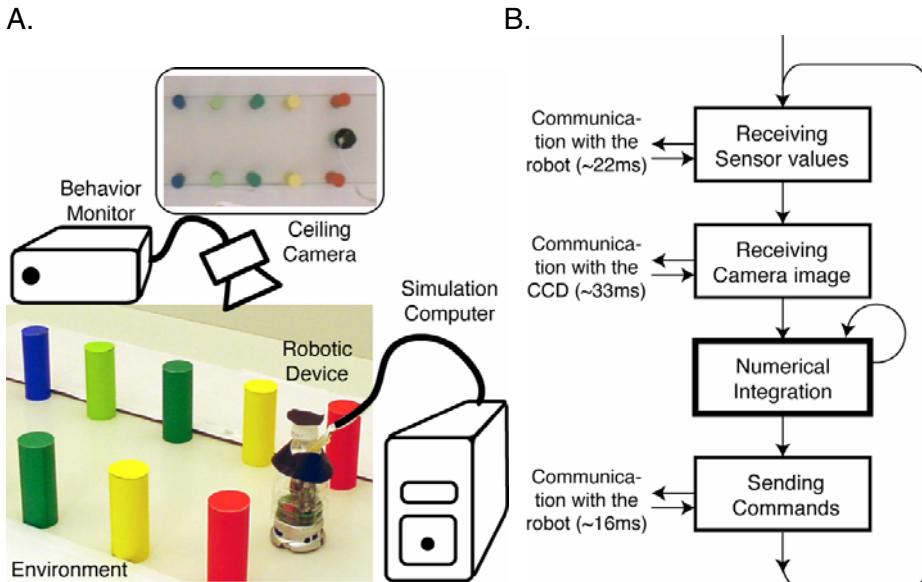


Fig. 1. A. The experimental platform of the neural network model which has real-time communication with the environment through the robot's body. The model described by differential equations like Eq.1 is simulated on the simulation computer by using numerical integration as an on-line system. The ceiling camera monitors the behavior of the robot during the experiment, independent of the simulation. B. In the simulation computer, the numerical integration is interrupted by communication with the robot for updates of sensor values and the CCD camera images (320x 240 pixels with RGB color plains) and for controlling the motor speed of the right and left wheels. Based on the time for communication with the robot, the sensory input is regarded as a constant value during the interval, ~70ms, in every iteration step.

this purpose, we have been interested in the synchronization mechanism frequently observed in biological systems [8,9,10,11]. Oscillations, such as the constant blinking of a fire-fly's light, smoothly synchronize with other oscillations via interactions. Interestingly, in the brain this property is not only observed in neuronal activities of local circuits, but also in the EEG relationship among multiple brain regions, which may contribute to bridge different cognitive functions [6,7]. We simply describe a coupling system of oscillatory units by the differential equation as follows:

$$\frac{d\phi_i}{dt}(t) = \omega_0 + \left\{ \beta_0 - I_i(t) - \sum_j w_{ij} f_j(t) \right\} \sin \phi_i(t), \quad (1)$$

$$f_i(t) = \max(\cos(\phi_i), 0), \quad (2)$$

where ω_0 is the intrinsic frequency constant, and β_0 is the stabilization constant to satisfy $(\omega_0 + 1) > \beta_0 > \omega_0 > 0$. This equation includes the coupling factor between the i -th and j -th units by the synaptic weight w_{ij} . The factor $\cos(\phi_i)$ represents the membrane potential of the i -th unit, so that the value stays at the resting state when

$I_i = 0$ and starts to oscillate when the input is coming, $I_i = 1$. By assuming the coupling factor, active units quickly synchronize together. If the intrinsic frequency ω_0 is different in units the population activity does not exhibit the whole synchronization but goes to partial synchronizations depending on coupling strengths and the variance among intrinsic frequencies. This kind of equation is conventionally simulated by numerical integration, such as a fourth-order Runge-Kutta method, taking four steps in each iteration step to avoid overshoot of the error estimation. Therefore, even if the simulation system evaluates the external input from the real environment with dynamical changes, the system is capable of updating in every iteration step. Thus, the input is required to be stable in intervals between the steps, or the simulation system works faster than the time constant of input changes, as is illustrated in Fig. 1.

3 How the Mechanisms Are Implemented into Robots

To save on the costs of hardware development in the first step, we choose a mobile robot platform that is easily controlled by remote computers. Khepera and e-puck commercial robot platforms have serial connection ports. The simulation system working on the remote computer receives sensory inputs and sends motor commands as an on-line system. Due to hardware constraints, it takes approximately 22ms to obtain

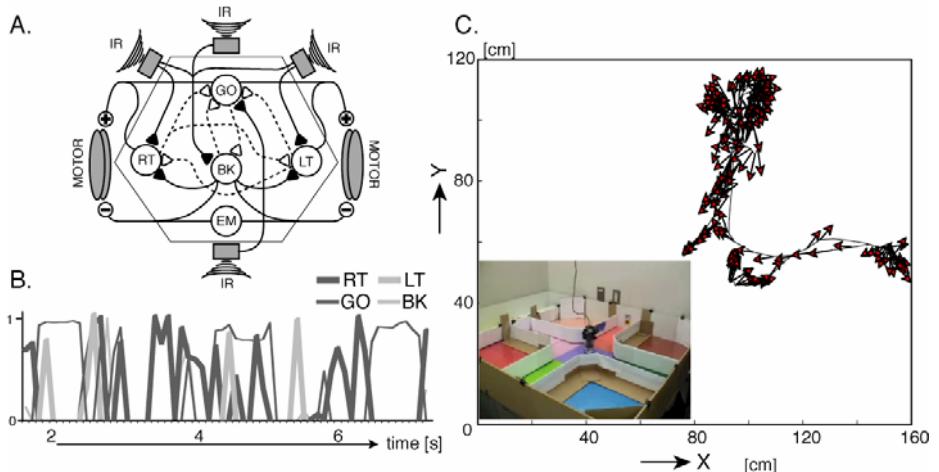


Fig. 2. A. Schematic illustration of the simple motor neural network model. The network has synaptic connections with sensors and motors and connections among neurons as mutual inhibition. GO, BK, RT, LT and EM, cells act as going forward and backward, right and left turns, and emergency back. IR represents distance sensor via infrared rays. B. The time evolution of the cell activities in 3s. Each iteration step corresponds to 0.1s, including the calculation time of the model. The GO cell tends to fire constantly for going straight and the activity is frequently interrupted by RT and LT activities for turns. C. The robot trajectory observed by the behavior monitor (Fig 1A), with a scene the moving robot. The arrow represents the heading of the robot. The wall avoidance navigation is successfully demonstrated.

the values from distance sensors attached on the girth of the robot's body, approximately 16ms to change the motor speeds of the wheels on the body, and approximately 33ms to transfer image data from the CCD camera in the top of the body.

An example is a sensorimotor system to avoid collisions with walls of the environment. The mobile robot is controlled by a simple neural network model that consists of five neurons, similar to those described by Eq.1, having a single response of firing, or a spike (Fig. 2A). According to synaptic connections, neurons of the model respond to the sensory information by generating spikes (Fig. 2B). To minimize calculation costs, we adopted a weighted summation of spiking activities in each time step to determine the speed parameters of the robot's right and left wheels, so that the robot goes back if the value is negative. As the result, the robot can avoid the walls, displaying a smooth trajectory in the cross maze, as shown in Fig. 2C. However, this mechanism depends on the calculation time for the numerical integration, which is affected by the size of the model, and this fact limits the maximum speed of movements for gaining time until the next step. In the next chapter, we describe how the sensorimotor system is coupled with oscillator networks to encode memory of the behavioral sequences and to make a decision as to which arm of the cross maze environment is suitable for the current situation.

4 What Experimental Data Can Be Obtained

How can we develop this system to encompass the whole brain network? We have proposed oscillator neural networks with "theta phase coding" [8,9,10], in accordance

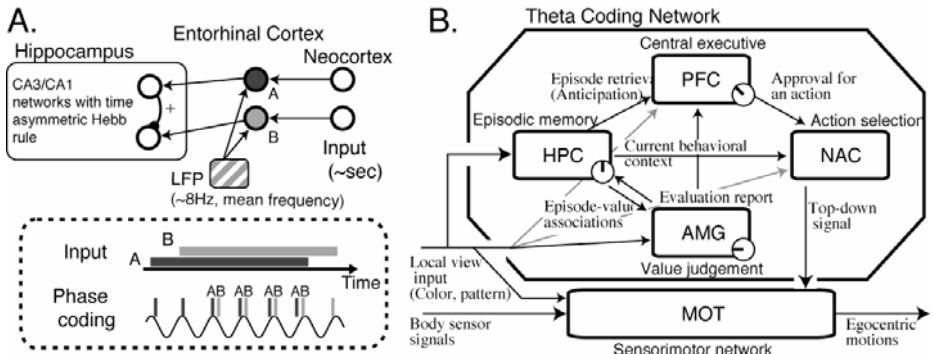


Fig. 3. A. The neural mechanism of theta phase coding, which is consistent with experimental observations [12,13]. In the entorhinal cortex, sensory inputs of A and B in a second are transformed into sequential firings of A and B in the compressed cycle of the theta rhythm, ~125ms. This enables the hippocampal network (HPC) to encode the sequence in the synaptic connection from A to B neurons, by using the Hebbian plasticity, which is known to occur at around 25ms according to the biological data [14]. Adapted from Wagatsuma and Yamaguchi, 2007 [9]. B. Schematic illustration of the extended neural network. Each box represents a network having oscillator neural units and a global oscillation of the local field potential (LFP), a circle with the bar. AMG, NAC, PFC and MOT represent the amygdala, the nucleus accumbens, the prefrontal and the primary motor cortex. (MOT is the same as Fig.2.). Simply, body sensor signals only enter the MOT. Adapted from Wagatsuma and Yamaguchi [11].

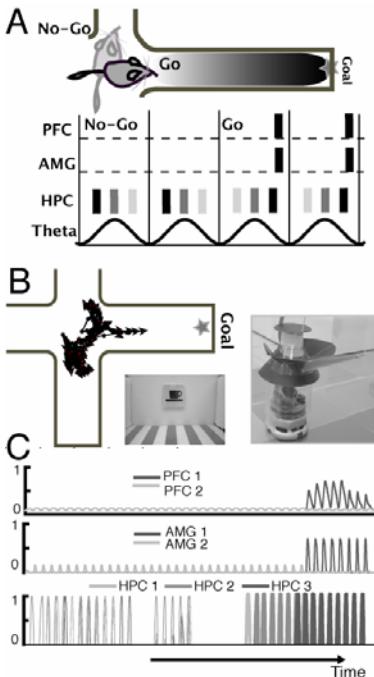


Fig. 4. A. A working hypothesis of the context-dependent adaptive behavior by using theta phase coding. The current context is discriminated by the temporal sequence in the cycle, determining in which direction to go. Schematically illustrated of Wagatsuma and Yamaguchi [11]. B-C. An example of the trajectory of the robot after encoding of episodes (B). The time evolution of neural activities of individual networks (C). The time course corresponds to that of the above trajectory. When facing the wrong arm of the maze, the PFC network does not activate despite HPC memory retrievals, while when facing the correct arm, the robot starts to get into the arm because of the PFC activation, *i.e.* decision-making signal, which is triggered by coincidental inputs from the HPC and the AMG. Note that the MOT network controls the spontaneous behavior of turning in the center without the top-down signals. Adapted from Wagatsuma and Yamaguchi [11].

with the neural mechanism in the hippocampus, which is known to be crucial for episodic memory formation in the human and is highlighted in the spatial map representation in animal cases. As shown in Fig. 3A, theta phase coding enables the network to encode memory of the running behavior by forming directional synaptic connections between neurons, even if the running experience happens only once. This is episodic memory. In comparison to reinforcement learning mechanisms, which reinforce successful memory through repetitive trials, the present mechanism generates a repetition of the current behavioral sequence in the form of sequential firing that appears in repeated oscillation cycles to retain the memory of the one-time experience. The functional role of episodic memory in adaptive behaviors in the brain study is still unclear; however, the mechanism allows for maintaining individuality in memory, apart from the accumulation of bulk memory, and to bind together other emotional aspects, which are evoked in the experience. We expect the mechanism to

have a functional role in finding the difference between the present and the past, even in similar situations, and to obtain a flexible solution learnt from the past memory by the recognition of the difference. Figure 4 shows experimental data of an extended network designed as a temporal hierarchical system, as shown in Fig. 3B [11]. Networks in the upper system have a slow time constant, which is determined by the cycle of LFP oscillations, requiring nearly 60 iteration steps. The slow variable governs the lower system with spiking activities in each step. Oscillator networks coordinate the coupling between activities in different time scales. In this hierarchical system, LFP oscillations of individual networks modulate all internal units and externally mediate global timings between networks by using the oscillator synchronization dynamics for information transfer. In this way, we demonstrated a real-time and real-world brain-based robot system, by focusing on the temporal structure of the neural mechanism. The neural network model used to construct our brain-based robot can be extended to encompass other functional networks relevant to other cognitive functions.

5 Concluding Remarks

One of the most important concepts in robotics and artificial intelligence is the embodied approach, focusing on the importance of having a body that actually connects to the external environment. This setup suggests that the intelligence develops through sensorimotor skills and the reality of situations that would be confronted in the environment. We support this concept and propose embodiment in the time domain. Nervous systems have a variety of time constants. A muscle jerk reflex is processed in local neurons and takes only around 40ms for the initiation of the leg movement. Sending a motor command from the brain after the recognition of visual cues takes around 150ms, as shown in the monkey case. Making a decision depending on the current context is considered to take at least a few seconds. These different time courses proceed in the nervous system in parallel, and individual circuits work independently and cooperatively within the constraints of being part of the hierarchical system in the time domain. We believe that adaptability to environmental situations emerges in such a system, and we propose the experimental platform of oscillatory neural networks having real-time communication with the environment through the robot's body. On the other hand, this approach to attempt to construct the whole brain system by focusing on the temporal properties has many difficulties. A principal problem is the limitation of the performance of the simulation system. By using a small robot working in a restricted environment, rescale of the time unit is applicable to the investigation. Giving a reduction in time scale simply makes a slow movement. However, constructing a system that naturally interacts with us, for instance playing games in the open field [17,18], is very difficult in our current platform. An advanced simulation system, requiring parallel processing in multiple computers and low-cost digitizing of oscillator simulations, should be considered in the near future.

Acknowledgments. The authors would like to thank Mayumi Haga, Yutaka Watanabe and Takayoshi Kogai for technical assistances of the platform development, and especially thank Yoko Yamaguchi for enormous supports for the robotic research projects. This work was supported by JSPS KAKENHI (19650073).

References

1. Krichmar, J.L., Edelman, G.M.: Brain-based devices for the study of nervous systems and the development of intelligent machines. *Artificial Life* 11, 63–77 (2005)
2. McCarthy, J., Hayes, P.J.: Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence* 4, 463–502 (1969)
3. Yamaguchi, Y., Shimizu, H.: Pattern recognition with figure-ground separation by generation of coherent oscillations. *Neural Networks* 7(1), 49–63 (1994)
4. Hirakura, Y., Yamaguchi, Y., Shimizu, H., Nagai, S.: Dynamic linking among neural oscillators leads to flexible pattern recognition with figure-ground separation. *Neural Networks* 9(2), 189–209 (1996)
5. Gray, C.M., König, P., Engel, A.K., Singer, W.: Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature* 338, 334–337 (1989)
6. Rodriguez, E., George, N., Lachaux, J.P., Martinerie, J., Renault, B., Varela, F.J.: Perception's shadow: long-distance synchronization of human brain activity. *Nature* 397, 430–433 (1999)
7. Mizuhara, H., Wang, L.Q., Kobayashi, K., Yamaguchi, Y.: A long-range cortical network emerging with theta oscillation in a mental task. *Neuroreport* 15(8), 1233–1238 (2004)
8. Wagatsuma, H., Yamaguchi, Y.: Cognitive map formation through sequence encoding by theta phase precession. *Neural Computation* 16, 2665–2697 (2004)
9. Wagatsuma, H., Yamaguchi, Y.: Neural dynamics of the cognitive map in the hippocampus. *Cognitive Neurodynamics* 1, 119–141 (2007)
10. Yamaguchi, Y., Sato, N., Wagatsuma, H., Wu, Z., Molter, C., Aota, Y.: A unified view of theta-phase coding in the entorhinal-hippocampal system. *Current Opinion in Neurobiology* 17, 1–8 (2007)
11. Wagatsuma, H., Yamaguchi, Y.: Context-dependent adaptive behavior generated in the theta phase coding network. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) *ICONIP 2007, Part II. LNCS*, vol. 4985, pp. 177–184. Springer, Heidelberg (2008)
12. O'Keefe, J., Recce, M.L.: Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus* 3(3), 317–330 (1993)
13. Skaggs, W.E., McNaughton, B.L., Wilson, M.A., Barnes, C.A.: Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus* 6(2), 149–172 (1996)
14. Bi, G.Q., Poo, M.M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience* 18(24), 10464–10472 (1998)
15. Jones, M.W., Wilson, M.A.: Phase precession of medial prefrontal cortical activity relative to the hippocampal theta rhythm. *Hippocampus* 15(7), 867–873 (2005)
16. Siapas, A.G., Lubenov, E.V., Wilson, M.A.: Prefrontal phase locking to hippocampal theta oscillations. *Neuron* 46(1), 141–151 (2005)
17. McKinstry, J.L., Edelman, G.M., Krichmar, J.L.: A cerebellar model for predictive motor control tested in a brain-based device. *Proc. Natl. Acad. Sci. USA* 103, 3387–3392 (2006)
18. Krichmar, J.L., Edelman, G.M.: Design Principles and Constraints Underlying the Construction of Brain-Based Devices. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) *ICONIP 2007, Part II. LNCS*, vol. 4985, pp. 157–166. Springer, Heidelberg (2008)

Part II

Neurodynamics

First Spiking Dynamics of Stochastic Neuronal Model with Optimal Control

Yongjun Wu^{1,2,*}, Jianhua Peng², and Ming Luo²

¹ Department of Engineering Mechanics, Shanghai Jiao Tong University,
Shanghai 200240, P.R. China

² School of Information Science and Engineering, East China University of Science
and Technology, Shanghai 200237, P.R. China
Tel. (Fax): 86-21-64252705
yongjun.wu@yahoo.com, jianhuapeng@hotmail.com,
mluo@ecust.edu.cn

Abstract. First-spiking dynamics of optimally controlled neuron under stimulation of colored noise is investigated. The stochastic averaging principle is utilized and the model equation is approximated by diffusion process and depicted by Itô stochastic differential equation. The control problems for maximizing the resting probability and maximizing the time to first spike are constructed and the dynamical programming equations associated with the corresponding optimization problem are established. The optimal control law is determined. The corresponding backward Kolmogorov equation and Pontryagin equation are established and solved to yield the resting probability and the time to first spike. The analytical results are verified by Monte Carlo simulation. It has shown that the proposed control strategy can suppress the overactive neuronal firing activity and possesses potential application for some neural diseases treatment.

Keywords: Optimal Control, Stochastic Averaging, First-Spiking, Colored Noise, Monte Carlo Simulation.

1 Introduction

Many experiments have shown stochastic behavior of central neurons. This stochasticity or randomness is due to the effect of noise, or stochastic stimulation. There are three kinds of noise affecting the neuron. One is thermal noise and one is the noise arising from the finite number of ion channels in a patch of neuronal membrane. Another one is the noise due to signal transmission and network effects [1]. When the neuron receives inputs, the membrane potential may be hyperpolarized or depolarized. The neuron fires a spike (or an action potential) whenever its membrane depolarization reaches a threshold and the membrane potential is reset to a new value. After a refractory period of time, the neuron is ready to fire again.

Stochastic neuronal model has been investigated by many researchers [2-6]. Usually, the noise is treated as Gaussian white noise (GWN) for mathematical

* Corresponding author.

tractability. But GWN is an ideal model and in practice, the noise should be treated as colored noise. Furthermore, studies have shown that the time-to-first-spike may play an important role in neural coding [7-9]. The brain perhaps does not have enough time to evaluate more than one spike from each neuron per processing step. In this case, the first spike contains most of the information [1, 7, 9].

On the other hand, the controlling of neuron is significant due to the potential applications of the treatment for some neurological and psychiatric disease, such as Parkinson's and epilepsy disease. Many studies have shown that some neurons in substantia nigra of a Parkinson's patient are overactive and maybe fire inappropriately, causing non-smooth movement and other symptoms [10]. A direct treatment way is to suppress the overactive neuronal firing activity. Some clinical operations have demonstrated that the suppression of the overactive neuronal firing activity can dramatically quiet the Parkinson's symptoms [11, 12].

Theoretically, this paper investigates first-spiking dynamics of optimally controlled neuron under stimulation of colored noise. Combining stochastic averaging principle with stochastic dynamical programming, the mathematical formulation of the control problem for maximizing the resting probability and maximizing the time to first spike are constructed. Example is given to illustrate the application of the proposed method. The theoretical results are verified by Monte Carlo simulation.

2 Stochastic Averaging Principle

The stochastic dynamics of a system with finite variables is described by the following stochastic differential equations

$$\dot{X}_i = \varepsilon f_i(X, t) + \varepsilon^{1/2} g_{ik}(X, t) \xi_k(t), \quad i = 1, 2, \dots, n; \quad k = 1, 2, \dots, m. \quad (1)$$

where ε is a small parameter, $X(t)$ is an n -vector stochastic process. f_i and g_{ik} satisfy certain requirements which are almost invariably met in practice. $\xi_k(t)$ are wide-band stationary and ergodic random processes with zero mean and correlation functions $R_{kk}(\tau)$ or spectral densities $S_{kk}(\omega)$, or $R_{kk}(\tau)$ decays with τ fast enough. The repeated subscript represents summation.

According to Stratonovich-Khasminskii limit theorem [13, 14], X converge weakly to n -dimensional diffusion processes in a time interval of ε^{-1} order as $\varepsilon \rightarrow 0$, which can be represented by the following Itô stochastic differential equations:

$$dX_i = \varepsilon m_i(X) dt + \varepsilon^{1/2} \sigma_{il}(X) dB_l(t), \quad i = 1, 2, \dots, n; \quad l = 1, 2, \dots, r. \quad (2)$$

m_i and σ_{il} are called drift and diffusion coefficients, respectively. $B_l(t)$ are standard Wiener processes. m_i and σ_{il} can be obtain from the following formula

$$\begin{aligned} m_i(X) &= \left\langle f_i(X, t) + \int_{-\infty}^0 \frac{\partial g_{ik}(X, t)}{\partial X_j} g_{jl}(X, t + \tau) R_{kl}(\tau) d\tau \right\rangle_t, \\ \sigma_{il}(X) \sigma_{jl}(X) &= b_{ij}(X) = \left\langle \int_{-\infty}^0 g_{ik}(X, t) g_{jl}(X, t + \tau) R_{kl}(\tau) d\tau \right\rangle_t. \end{aligned} \quad (3)$$

Herein,

$$\langle \cdot \rangle_t = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0}^{t_0+T} \langle \cdot \rangle dt \quad (4)$$

denotes the averaging with respect to t .

3 Controlled Neuron and Dynamical Programming Equations

Consider the following controlled stochastic neuronal model.

$$dx(t)/dt = f(x, t) + g_i(x, t)\xi_i(t) + u(x, t), \quad i = 1, 2, \dots, n. \quad (5)$$

where $x(t)$ is membrane potential, $f(x, t)$ is deterministic input usually possessing periodic current. $\xi_i(t)$ is stationary colored noise with zero mean and correlated function $R_{ik}(\tau)$. $u(x, t)$ is feedback control.

According to stochastic averaging principle, equation (5) can be approximated by the following Itô stochastic differential equation

$$dx(t) = [m(x) + u(x, t)]dt + \sigma(x)dB(t) \quad (6)$$

where

$$\begin{aligned} m(x) &= \left\langle f(x, t) + \int_{-\infty}^0 \frac{\partial g_i(x, t)}{\partial x} g_j(x, t+\tau) R_{ij}(\tau) d\tau \right\rangle_t, \\ \sigma^2(x) &= \left\langle \int_{-\infty}^{\infty} g_i(x, t) g_j(x, t+\tau) R_{ij}(\tau) d\tau \right\rangle_t, \quad i, j = 1, 2, \dots, n. \end{aligned} \quad (7)$$

The initial resting potential is x_0 and the threshold is x_{th} . When membrane potential reaches x_{th} for the first time, a spike (action potential) is generated and propagates alone the axon to the next neuron. We are interested in the probability of the controlled neuron remaining quiet, viz., resting probability. The other focus is the time to first spike of the controlled neuron, which is defined as $T = \inf \{t > 0, x(t) \geq x_{th}\}$.

Consider the control problem of maximizing the resting probability of the neuron. Namely, the probability of $x(t)$ being within the interval $[x_{lb}, x_{th}]$. Introduce the value function

$$V(x, t) = \sup_{u \in U} P\{x(s, u) \in [x_{lb}, x_{th}], \quad t < s \leq t_f \mid x(t, u) \in [x_{lb}, x_{th}]\} \quad (8)$$

Equation (8) implies that $V(x, t)$ is the probability of the optimally controlled neuron keeping quiet. $u \in U$ denotes the control constraint and t_f is the final control time. Based on the stochastic dynamical programming principle, the following dynamical programming equation can be established

$$\sup_{u \in U} \left\{ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2(x) \frac{\partial^2 V}{\partial x^2} + [m(x) + u(x, t)] \frac{\partial V}{\partial x} \right\} = 0 \quad 0 \leq t \leq t_f, x \in [x_{lb}, x_{th}] \quad (9)$$

with boundary conditions

$$V(x_{th}, t) = 0 \quad (10)$$

$$V(x_{lb}, t) = \text{finite} \quad (11)$$

and final time condition

$$V(x, t_f) = 1, x \in [x_{lb}, x_{th}] \quad (12)$$

The control problem of maximizing the time to first spike can be formulated similarly. Let $E[\tau(x, u)]$ denote the mean first spiking time of controlled system (5). Define the value function

$$V_1(x) = \sup_{u \in U} E[\tau(x, u)] \quad (13)$$

which implies V_1 is the time to first spike of optimally controlled system (5). Based on the dynamical programming principle, the following dynamical programming equation can be yielded

$$\sup_{u \in U} \left\{ \frac{1}{2} \sigma^2(x) \frac{d^2 V_1}{dx^2} + [m(x) + u(x, t)] \frac{dV_1}{dx} \right\} = -1, x \in [x_{lb}, x_{th}] \quad (14)$$

with boundary conditions

$$V_1(x_{th}) = 0 \quad (15)$$

$$V_1(x_{lb}) = \text{finite} \quad (16)$$

The optimal control law can be derived from maximizing the left side of equation (9) and equation (14) with respect to u . Assume the control is of the form

$$|u| \leq b \quad (17)$$

where b is a positive constant. Obviously, when $|u|=b$ and $u(x, t) \partial V / \partial x$ reaches maximum, the left side of equation (9) will be maximum. Because V is resting probability of the neuron, it is reasonable that V is smaller when x is closer to the threshold. So V is a monotonously decreasing function of x and $\partial V / \partial x < 0$. We can obtain the optimal control

$$u = b \text{sgn}(\partial V / \partial x) = -b. \quad (18)$$

Inserting u into equation (6), the final optimally controlled Itô stochastic differential equation is

$$dx(t) = \eta(x)dt + \sigma(x)dB(t) \quad (19)$$

where $\eta(x) = m(x) - b$, $B(t)$ is unit Wiener process.

Similarly, the optimal control law can be derived from maximizing the left side of equation (14). Because V is a monotonously decreasing function of x , we can conclude that V_1 is also a monotonously decreasing function of resting potential. This

conclusion can be justified by the following numerical example. So the same optimal control as the form of equation (18) can be obtained.

4 Backward Kolmogorov Equation and Pontryagin Equation

Equation (19) is the Itô stochastic differential equation representing optimally controlled system (5). The resting probability of the controlled system is defined as

$$P_{re}(t|x_0) = P\{x(s) \in [x_{lb}, x_{th}], 0 \leq s \leq t, x_0 \in [x_{lb}, x_{th}]\} \quad (20)$$

Then the probability of the neuron firing the first spike is defined as $P_f=1-P_{re}$. P_{re} is governed by the following backward Kolmogorov equation [15]

$$\frac{\partial P_{re}}{\partial t} = \frac{1}{2} \sigma^2(x_0) \frac{\partial^2 P_{re}}{\partial x_0^2} + \eta(x_0) \frac{\partial P_{re}}{\partial x_0} \quad (21)$$

with initial condition

$$P_{re}(0|x_0)=1, x_0 \in [x_{lb}, x_{th}] \quad (22)$$

and boundary conditions

$$P_{re}(t|x_{th})=0, \quad (23)$$

$$P_{re}(t|x_{lb})=\text{finite}. \quad (24)$$

Herein,

$$\begin{aligned} \eta(x_0) &= \eta(x)|_{x=x_0}, \\ \sigma(x_0) &= \sigma(x)|_{x=x_0}. \end{aligned} \quad (25)$$

Note that the boundary condition (24) is qualitative and should be quantified at boundary $x_0=x_{lb}$ as the following

$$\frac{\partial P_{re}}{\partial t} = \frac{1}{2} \sigma^2(x_0) \frac{\partial^2 P_{re}}{\partial x_0^2} + \eta(x_0) \frac{\partial P_{re}}{\partial x_0} \Big|_{x_0=x_{lb}}. \quad (26)$$

Usually, the partial differential equation (21) has no exact solution. It can be solved by finite difference method such as Crank-Nicolson scheme.

The time to first spike of the controlled system (5) is governed by the following Pontryagin equation

$$\frac{1}{2} \sigma^2(x_0) \frac{d^2 T}{dx_0^2} + \eta(x_0) \frac{dT}{dx_0} = -1, \quad (27)$$

The boundary conditions are

$$T(x_{lb})=\text{finite}, \quad (28)$$

$$T(x_{th})=0, \quad (29)$$

$\sigma(x_0)$ and $\eta(x_0)$ are defined as equation (25).

Assume that $x_0=x_{lb}$ is a reflecting boundary, the solution of equation (28) can be expressed as [15]

$$T(x_0) = 2 \int_{x_0}^{x_{th}} dy / \phi(y) \int_{x_{lb}}^y \phi(z) dz / \sigma^2(z) \quad (30)$$

where

$$\phi(z) = \exp \left\{ \int_{x_{lb}}^z 2\eta(z) / \sigma^2(z) dz \right\} \quad (31)$$

5 Numerical Example

Consider the following controlled leaky integrate-and-fire neuron, the dynamical equation is

$$dx(t)/dt = \lambda - x(t) + \xi(t) + u(x, t) \quad (32)$$

x is membrane potential. $\xi(t)$ is stationary colored noise with correlation function $R(\tau)=D\exp[-|\tau|/\tau_0]$. λ , D and τ_0 are constants. τ_0 is the correlation time of $\xi(t)$. u is feedback control.

Following the procedures shown in Section 2-4, we obtained some results. In Figs.1-2, — denotes theoretical results and \bullet \blacktriangle are results from Monte Carlo simulation. From Fig.1(a) and Fig.2, we can see that both the resting probability and the time to first spike are monotonously decreasing functions of initial resting potential x_0 , which justifies the derivation of equation (19). The resting probability can be enhanced by increasing the magnitude of the feedback control (see Fig. 1(b)). Namely, the higher first spiking probability P_f is decreased and the overactive neuronal firing activity is suppressed. Accordingly, the time to first spike is prolonged by increasing the control (see Fig. 2). All the theoretical results are in rather good agreement with those from Monte Carlo simulation.

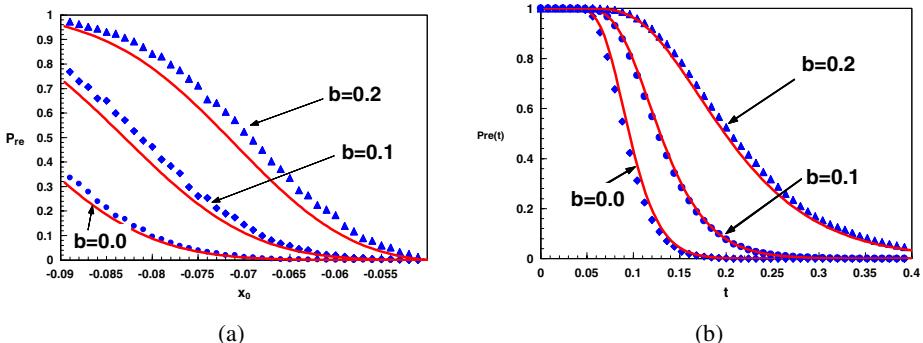


Fig. 1. Resting probability of controlled system (32). $x_{lb}=-0.09$ Volt, $x_{th}=-0.05$ Volt, $\lambda=0.3$, $D=0.05$, $\tau_0=0.01$, (a) $t=0.1172$ s. (b) t is a variable and $x_0=-0.0864$ Volt.

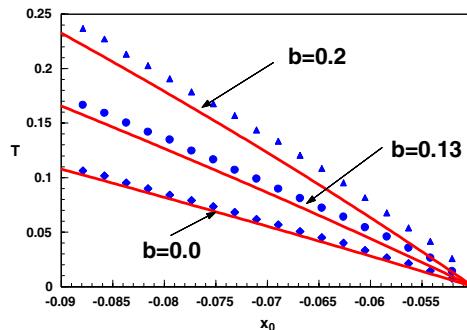


Fig. 2. Time to first spike of the controlled system (32). $x_{lb}=-0.09\text{Volt}$, $x_{th}=-0.05\text{Volt}$, $\lambda=0.3$, $D=0.05$, $\tau_0=0.01$.

6 Conclusions

In this paper, the stochastic averaging principle and dynamical programming principle are combined to investigate the first-firing dynamics of optimally controlled neuron stimulated by colored noise. After stochastic averaging, the neuronal model equation is substituted with Itô stochastic differential equation. The dynamical programming equations for maximizing the resting probability and time to first spiking are established. Based on the dynamical programming principle and the control constraint, the optimal control law is determined and the Itô stochastic differential equation describing the optimal control problem is obtained. The backward Kolmogorov equation and Pontryagin equation are established and are solved to yield the resting probability and time to first spiking of the controlled neuronal system. The resting probability and the mean time to first spike can be enhanced by the feedback control. The analysis of the result shows the overactive neuronal firing activity can be suppressed by increasing the control, which is significant for the treatment of some neurological disease. The agreement between the theoretical result and Monte Carlo simulation justifies the validity of the presented method.

The advantages of the proposed method are obvious. Because the noise is non-white, the method of diffusion process can not be used directly. After stochastic averaging, the method of diffusion process can be used. Another merit of the proposed control strategy is that the optimal control law can be obtained without solving the dynamical programming equation, which reduces the difficulty of mathematical manipulation.

Acknowledgements

The work reported in this paper was supported by the National Natural Science Foundation of China (Grant No. 10802030) and the Specialized Research Fund for Doctoral Program of Higher Education of China (Grant No. 200802511005).

References

1. Gerstner, K.: Spiking Neuron Models-Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (2002)
2. Barbi, M., Chillemi, S., Di, G.A.: The Leaky Integrate-and-Fire with Noise-A Useful Tool to Investigate SR. Chaos, Chaos Soliton Fract. 11, 1849–1853 (2000)
3. Fourcaud, N., Brunel, N.: Dynamics of the Firing Probability of Noisy Integrate-And-Fire Neurons. Neural Comput. 14, 2057–2110 (2002)
4. Brunel, N., Latham, P.E.: Firing Rate of the Noisy Quadratic Integrate-And-Fire Neuron. Neural Comput. 15, 2281–2306 (2003)
5. Tuckwell, H.C., Wan, F.Y.M.: Time to First Spike in Stochastic Hodgkin–Huxley Systems. Physica A 351, 427–438 (2005)
6. Lo, C.F., Chung, T.K.: First Passage Time Problem for the Ornstein-Uhlenbeck Neuronal Model. In: King, I., et al. (eds.) ICONIP 2006, Part I. LNCS, vol. 4232, pp. 324–331. Springer, Heidelberg (2006)
7. Johansson, R.S., Birznieks, I.: First Spikes In Ensembles Of Human Tactile Afferents Code Complex Spatial Fingertip Events. Nat. Neurosci. 7, 170–177 (2004)
8. Rullen, R.V., Gautrais, J., Delorme, A., Thorpe, S.: Face Processing Using One Spike Per Neurone. BioSystems 48, 229–239 (1998)
9. Thorpe, S., Fize, D., Marlot, C.: Speed of Processing in the Human Visual System. Nature 381, 520–522 (1996)
10. Lang, A.E., Lozano, A.M.: Parkinson’s Disease-Second of Two Parts. New Engl. J. Med. 339(16), 1130–1143 (1998)
11. Luo, J., Kaplitt, M.G., Fitzsimons, H.L., et al.: Subthalamic GAD Gene Therapy in a Parkinson’s Disease Rat Model. Science 298(5592), 425–429 (2002)
12. Han, X., Boyden, E.S.: Multiple-Color Optical Activation, Silencing, and Desynchronization of Neural Activity, with Single-Spike Temporal Resolution. PLoS One 2(3), e299 (2007)
13. Stratonovich, R.L.: Topics in the Theory of Random Noise, vol. 1. Gordom and Breach, New York (1963)
14. Khasminskii, R.Z.: A Limit Theorem for Solution of Differential Equations with Random Right Hand Sides. Theory Probab. Appl. 11, 390–405 (1966)
15. Zhu, W.Q.: Nonlinear Stochastic Dynamics and Control-Hamiltonian Theoretical Framework. Science Press, Beijing (2003) (in Chinese)

BCM and Membrane Potential: Alternative Ways to Timing Dependent Plasticity

Johannes Partzsch*, Christian Mayr**, and Rene Schüffny

Chair for Parallel VLSI Systems and Neural Circuits,

University of Technology Dresden, Germany

{partzsch, mayr, schueffn}@iee.et.tu-dresden.de

Abstract. The Bienenstock-Cooper-Munroe (BCM) rule is one of the best-established learning formalisms for neural tissue. However, as it is based on pulse rates, it can not account for recent spike-based experimental protocols that have led to spike timing dependent plasticity (STDP) rules. At the same time, STDP is being challenged by experiments exhibiting more complex timing rules (e.g. triplets) as well as simultaneous rate- and timing dependent plasticity. We derive a formulation of the BCM rule which is based on the instantaneous postsynaptic membrane potential as well as the transmission profile of the presynaptic spike. While this rule is neither directly rate nor timing based, it can replicate BCM, conventional STDP and spike triplet experimental data, despite incorporating only two state variables. Moreover, these behaviors can be replicated with the same set of only four free parameters, avoiding the overfitting problem of more involved plasticity rules.

1 Introduction

One of the major research areas of neurobiology is long term learning (i.e. plasticity) of synapses in neural tissue [1234]. Synapses are the contact points between neurons, where information from the sending neuron arrives at the so-called presynaptic side and is transmitted via the synapse as a postsynaptic current (PSC) pulse to the receiving neuron. The concept of long term plasticity is used to describe the phenomenon that certain types of pre- and/or postsynaptic stimuli can have long lasting effects on the efficacy of this transmission [256], i.e. the size of the PSC, ranging from days up to a year. These phenomena are called long term depression (LTD) for diminished synaptic responses respectively long term potentiation (LTP) for enhanced responses. Long term plasticity has defied easy modeling, with a host of *in vivo* and *in vitro* research showing diverse expressions of this plasticity [23]. Depending on the induction protocol, spike rates [67], different spike patterns [8910], or membrane voltage control [3710] have been found to elicit changes in plasticity.

* To whom correspondence should be addressed.

** The first two authors contributed equally to the research described in this manuscript.

Various models have tried to incorporate these findings e.g. in implementations of the classical rate-based BCM rule [5][11][12] or the newer spike-based STDP rule [2][13]. Since both rules describe phenomena which have been well established experimentally as well as theoretically, several models try to achieve a synthesis of both rules. Deriving BCM from STDP formulations is relatively easy [14], but those rules cannot explain nonlinear STDP without introducing secondary state variables [4][9][15]. Trying to reach from BCM to STDP results in STDP with biologically unrealistic characteristics [11][12]. In contrast, in section 2.1 we derive a model of BCM which exhibits major BCM characteristics like LTD at low postsynaptic frequency and LTP at high ones, as well as a frequency threshold (section 3.2). At the same time, this model can be linked analytically to conventional STDP formulations, so the parameters of our model can be computed from those derived experimentally for STDP (section 2.2). Additionally, in section 3.3 we show that this model can replicate experimental data on higher order STDP effects, namely the triplet experiments by Froemke and Dan [9].

2 Methods

2.1 Model

Our plasticity rule is motivated by the original BCM rule formulation [5]. In this model, the weight m of a synapse changes dependent on presynaptic activity $d(t)$ and postsynaptic activity $c(t)$:

$$\frac{dm}{dt} = \phi(c(t) - \Theta_M) \cdot d(t) \quad (1)$$

The activities are instantaneous variables denoting changes of spiking frequency around a mean value, so that they can take on positive as well as negative values. $\phi(\cdot)$ is an arbitrary function that changes sign at zero, so that the weight m moves towards d for $c > \Theta_M$, whereas it moves in opposite direction for $c < \Theta_M$.

Since some of the classical BCM studies [7] as well as newer spike-timing studies [3][10] have found a dependence of synaptic plasticity on the postsynaptic membrane potential, we re-interpret the meaning of the variables in the BCM rule. We make two observations: First, the presynaptic activity $d(t)$ at a certain location of a neuron's dendrite (down to a single synapse) can be read out via conductance changes in the postsynaptic membrane that are mediated by presynaptic neurotransmitter release. Often this is modelled by an exponentially decaying synaptic conductance variable (see [16] for an overview). Second, postsynaptic activity $c(t)$ is directly related to postsynaptic membrane potential: Below spiking threshold, the more excitatory input a neuron receives, the higher is its membrane potential. Additionally, the membrane potential involves a similar temporal asymmetry with respect to postsynaptic spikes as standard STDP rules: shortly before a postsynaptic spike, the membrane potential usually is above rest (depolarized), whereas it is below rest (hyperpolarized) shortly after a spike.

Motivated by the first observation, we introduce a presynaptic activity variable, which is a low-pass filtered version of the presynaptic spike train:

$$s(t) = \hat{S} \cdot e^{-\frac{t-t_j^{\text{pre}}}{\tau_s}}, \quad t_j^{\text{pre}} \leq t < t_{j+1}^{\text{pre}}, \quad (2)$$

where τ_s is the decay time constant and \hat{S} denotes the amplitude of the response to a presynaptic pulse.

Following the second observation, we use a simple spike-response model [16] of the membrane potential as postsynaptic activity, denoted as $u(t)$. This model consists of a Dirac pulse and an exponential decay to account for hyperpolarization after a spike:

$$u(t) = U_p \cdot \delta(t - t_n^{\text{post}}) + U_{\text{refr}} \cdot e^{-\frac{t-t_n^{\text{post}}}{\tau_{\text{refr}}}}, \quad t_n^{\text{post}} - 0 < t < t_{n+1}^{\text{post}} - 0 \quad (3)$$

In this equation, U_p represents the area under the pulse curve, determining the pulse amplitude. $U_{\text{refr}} < 0$ is the amplitude of post-spike hyperpolarization and τ_{refr} is the membrane time constant. Sub-threshold variations of membrane potential before a spike are neglected for simplicity.

For most experimental protocols, we included an attenuation of the postsynaptic spike. This attenuation was implemented by weighting the action potential amplitude U_p with the negative membrane potential:

$$U_p \rightarrow U_p(t^{\text{post}}) = U_p \cdot (1 - \alpha_{\text{att}} \frac{u(t^{\text{post}} - 0)}{U_{\text{refr}}}) \quad (4)$$

With this formulation, a postsynaptic spike occurring shortly after a previous one will have amplitude $U_p \cdot (1 - \alpha_{\text{att}})$. Note that, due to setting the membrane voltage to U_{refr} after each pulse, the pulse amplitude will only depend on the time course of the membrane voltage since the last action potential.

The resulting voltage-based rule reads as follows:

$$\frac{dm}{dt} = (u(t) - \Theta_u) \cdot s(t), \quad (5)$$

where Θ_u represents the threshold between LTP and LTD. This equation is similar to that of an ion channel with reversal potential Θ_u , activated by presynaptic pulses. Interestingly, similar rules have been synthesized for solving image processing tasks in VLSI realizations of neural networks [17].

Following our motivation, s should be a conductance, i.e. $[s] = \text{A/V} = \text{S}$, and u should be a potential, i.e. $[u] = \text{V}$, so that $[U_p] = \text{Vs}$. Thus, the weight m has units of a charge: $[m] = \text{As}$. This may be interpreted as the amount of neurotransmitters released at the synapse. LTP would thus raise the available neurotransmitter amount, whereas LTD would lower it.

2.2 Explicit Expression and Parameter Fitting

Equation (5) can be integrated to arrive at an explicit expression for the weight at time t . For determination of the model's parameters, we will derive such

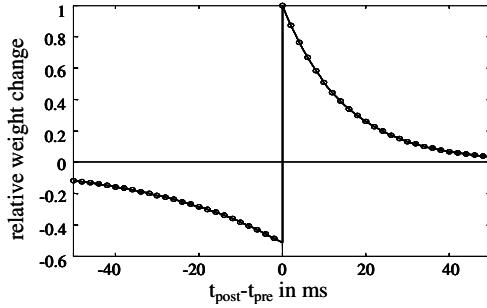


Fig. 1. Normalized STDP window of our rule for the parameters of Froemke and Dan [9] (see Fig. 2), derived analytically (solid line) and via simulations (circles, 60 pairings at 1Hz, protocol of [8])

expressions for a standard pairing protocol (see e.g. [8]). For clarity, we set $\Theta_u = 0$ and calculate the weight change triggered by a single pair of spikes with timing difference $\Delta t_n = t_n^{\text{post}} - t_n^{\text{pre}}$. Since the pairings have a low repetition frequency, the attenuation of U_p can be neglected. For pre-post pairings ($\Delta t_n > 0$) the following equation for times $t > t_n^{\text{post}}$ holds:

$$\Delta m_n(t) = \hat{S} \left(U_p + \frac{U_{\text{refr}}}{\frac{1}{\tau_s} + \frac{1}{\tau_{\text{refr}}}} \right) e^{-\frac{|\Delta t_n|}{\tau_s}} - \frac{\hat{S} U_{\text{refr}}}{\frac{1}{\tau_s} + \frac{1}{\tau_{\text{refr}}}} \cdot e^{-\frac{t-t_n^{\text{pre}}}{\tau_s}} \cdot e^{-\frac{t-t_n^{\text{post}}}{\tau_{\text{refr}}}} \quad (6)$$

Post-pre pairings ($\Delta t_n < 0$) lead to a similar relationship for times $t > t_n^{\text{pre}}$:

$$\Delta m_n(t) = \hat{S} \cdot \frac{U_{\text{refr}}}{\frac{1}{\tau_s} + \frac{1}{\tau_{\text{refr}}}} e^{-\frac{|\Delta t_n|}{\tau_{\text{refr}}}} - \frac{\hat{S} U_{\text{refr}}}{\frac{1}{\tau_s} + \frac{1}{\tau_{\text{refr}}}} \cdot e^{-\frac{t-t_n^{\text{pre}}}{\tau_s}} \cdot e^{-\frac{t-t_n^{\text{post}}}{\tau_{\text{refr}}}} \quad (7)$$

For small pairing frequency, i.e. $t \rightarrow \infty$, the terms dependent on t diminish in equations (6) and (7), so that the final weight change Δm_n merely depends on the timing difference Δt . Due to the exponential relationship, our model can be directly fitted to the exponential time window of standard STDP rules, parameterized by the amplitudes $A_{+/-}$ and time constants $\tau_{+/-}$ for LTP and LTD, respectively:

$$U_{\text{refr}} = A_- \cdot \left(\frac{1}{\tau_s} + \frac{1}{\tau_{\text{refr}}} \right) \frac{1}{\hat{S}} \qquad \tau_{\text{refr}} = \tau_- \quad (8)$$

$$U_p = (A_+ - A_-) \frac{1}{\hat{S}} \qquad \tau_s = \tau_+ \quad (9)$$

We have set $\hat{S} = 1\text{nS}$ and added units fAs to the amplitudes $A_{+/-}$ to arrive at membrane potential variations in the order of biological values.

Our analytical findings are confirmed by simulations of our BCM model with a low-frequency STDP pairing protocol, see Figure 1. As expected, the agreement with the exponential time window as derived above is excellent.

3 Results

3.1 Model Behaviour

Our model employs different mechanisms for LTP and LTD: While LTP results from sampling of presynaptic activity at postsynaptic spikes, LTD is continuously activated by coincidence of presynaptic activity and membrane hyperpolarization directly after a postsynaptic spike, see Figure 2A. Amplitudes of postsynaptic spikes are attenuated if they occur shortly after each other, so that for high postsynaptic firing rates the amount of LTP per spike is reduced. This counteracts the reduced amount of LTD per postsynaptic spike that results from shortened hyperpolarization periods in this regime.

It is important to note that our model parameters, being strongly linked to biophysical parameters, take on biologically realistic values when they are derived from standard STDP parameters (see section 2.2). The PSC time constant τ_s equals the STDP time constant for LTP, which is in the order of 20ms, a value compatible with NMDA synapse conductance changes [16,13]. The STDP time constant for LTD equals the membrane time constant τ_{refr} , which was found to be in the order of 10-40ms [1]. If we scale our membrane model, so that $U_{\text{refr}} = -5\text{mV}$, we arrive at a value $U_p = 151\mu\text{Vs}$ for realistic STDP parameters (see Fig. 2B, [1]), corresponding to a (rectangular) pulse of length 2ms and height 75mV, which are reasonable values for action potentials [1].

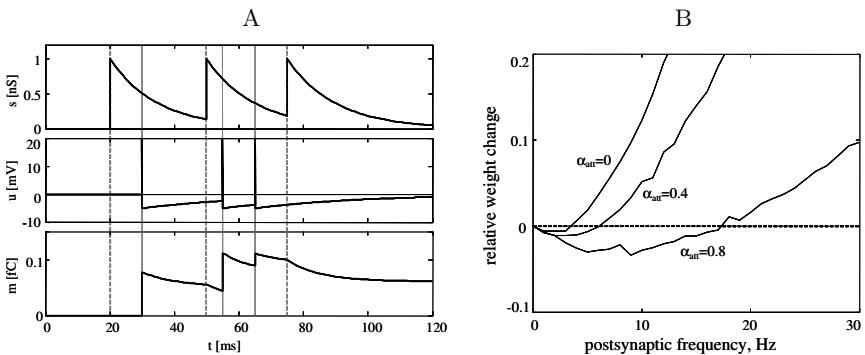


Fig. 2. A: progression of presynaptic activity s , postsynaptic activity u and weight m for a sample spike train. B: Weight change for different attenuation factors α_{att} when stimulating with Poisson spike trains at varying postsynaptic rates. Parameters for equation (5) as derived from Froemke and Dan [9] ($\tau_+ = 14.8\text{ms}$, $\tau_- = 33.8\text{ms}$, $A_+ = 1.01$, $A_- = -0.52$) via equations (8) and (9).

3.2 Frequency Dependence

In a first experiment, we applied our rule to a protocol recently used by Izhikevich and Desai [14] to test STDP models on their BCM characteristics. Thereby,

random Poisson spike trains of length 200s were generated, with the presynaptic firing rate held constant at 10Hz, but varying postsynaptic firing rate.

Figure 2B shows resulting frequency curves for different degrees of action potential attenuation. They all show typical BCM behaviour, depending on the attenuation factor α_{att} : The amount of LTD as well as the threshold between LTD and potentiation could be increased by raising α_{att} . This is because the postsynaptic action potential is the only source of LTP, so that its attenuation will directly result in a higher amount of LTD.

On the other hand, a positive threshold Θ_u would add a constant amount of LTD to the curves in Figure 2B, which shifts the entire curve down instead of changing its slope as for α_{att} . This is consistent with experimental findings on the plasticity-depolarization relationship [10]. Also, due to this shifting property, the frequency threshold Θ_M between LTD and LTP of the original BCM rule is directly related to the voltage threshold Θ_u in our model.

3.3 Triplet Protocols

We tested the performance of our model on triplet protocols using the data of Froemke and Dan [9]. Thereby, either two presynaptic and one postsynaptic or one presynaptic and two postsynaptic pulses were generated, and weight changes monitored dependent on the two independent time differences in the protocol. Such experiments are only poorly replicated by standard STDP models [4]. We did not fit our model especially to the triplet data, but used the STDP parameters measured in their control experiment, as before (see Figure 2B).

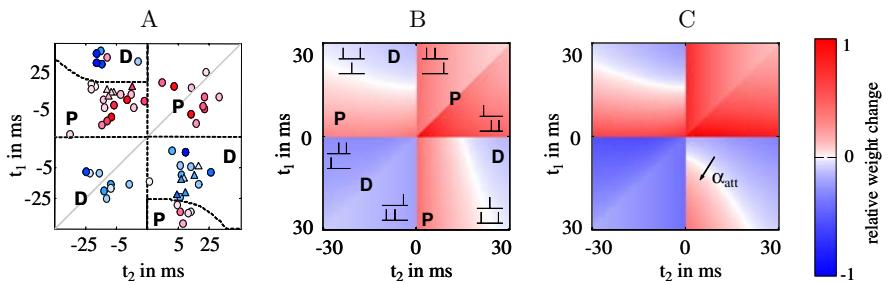


Fig. 3. Triplet experiments. Upper left part: 2 pre-, 1 post-synaptic spike; lower right part: 1 pre-, 2 post-synaptic spikes; A: experimental data of Froemke and Dan [9]; B: our basic model, using parameters from Froemke and Dan [9] (see Fig. 2); C: same as B, but with additional attenuation of postsynaptic action potentials, $\alpha_{att} = 0.8$.

Figure 3 shows relative weight changes after application of the spike triplets. The time differences are measured post-minus-pre, as in [9]. Our basic model matches the experimental results for most triplets, comparable to the biophysical model in [13]. Particularly, it reproduces the responses to pre-post-pre pairings (upper left quarter), but fails to account for the post-pre-post responses.

We can make our model consistent with the post-pre-post triplets if we add attenuation of postsynaptic spikes, see Figure 3C: The amount of LTD for the post-pre-post triplets is adjusted by the attenuation α_{att} (see arrow), so that our model can replicate the experimental data of [9] very well.

4 Discussion

In this paper, we have introduced a simple learning rule that is motivated by the original BCM formulation and membrane potential variations of biological neurons. Using a simple spike response neuron, we can directly map our model to standard pair-based STDP rules. Like iterative implementations of STDP, our model samples presynaptic activity at postsynaptic spikes for LTP. In contrast to that, postsynaptic hyperpolarization after action potentials is continuously integrated with presynaptic activity to arrive at a mechanism for LTD.

When replicating experimental data with such models, major challenges have been the integration of triplets [9] and higher order pairings [10,4] as well as frequency effects [14,15,6]. Several higher-order models with secondary state variables [15,9,4] have been proposed to reproduce some of these effects. However, incorporating no secondary state variables, our model can account for BCM effects, conventional STDP and triplet experiments with only four free parameters. The generalization capacity of our model can be seen from the fact that all these effects can be generated with one biologically realistic parameter set, while even higher order models have to fit several parameter sets to account for different experimental data [15,13]. In addition, our model is able to link the parameters of an STDP model to the ones of the neuron model and to BCM parameters.

Our model is also suited for implementation in neuromorphic hardware. Implementing a membrane-based Hebb rule similar to ours requires only low circuit complexity [17]. Compared to more involved realizations of neurobiological behavior, which replicate both STDP and complex membrane dynamics [18], our model is easier to implement. This is because in those models, the dynamic variables required by our model are already contained as membrane potential and synaptic conductance, so that the weight change may be directly derived from these quantities, requiring no separate circuitry for STDP state variables.

Because in our model, plasticity is directly linked to depolarization, experimental protocols employing this variable [10,3] could be qualitatively replicated, an advantage that purely spike-based models are lacking. Further analysis will also try to create a stronger link between the frequency threshold of conventional BCM and our voltage threshold. The metaplasticity of this threshold [5,11] might be accounted for by variations in resting potential [10,10] caused by metabolic changes in the dendritic membrane.

Acknowledgements. This work is supported by funding under the Sixth Framework Programme of the European Union under the grant no. 15879 (FACETS). We would thank W. Gerstner for fruitful discussions on plasticity.

References

1. Koch, C.: Biophysics of computation. Information processing in single neurons. In: Computational Neuroscience. Oxford University Press, Oxford (1999)
2. Morrison, A., Diesmann, M., Gerstner, W.: Phenomenological models of synaptic plasticity based on spike timing. *Biological Cybernetics* 98, 459–478 (2008)
3. Lisman, J., Spruston, N.: Postsynaptic depolarization requirements for LTP and LTD: a critique of spike timing-dependent plasticity. *Nature Neuroscience* 8(7), 839–841 (2005)
4. Pfister, J.P., Gerstner, W.: Triplets of spikes in a model of spike timing-dependent plasticity. *Journal of Neuroscience* 26(38), 9673–9682 (2006)
5. Bienenstock, E., Cooper, L., Munro, P.: Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience* 2(1), 32–48 (1982)
6. Dudek, S., Bear, M.: Homosynaptic long-term depression in area CA1 of hippocampus and effects of N-methyl-D-aspartate receptor blockade. *PNAS* 89, 4363–4367 (1992)
7. Artola, A., Bröcher, S., Singer, W.: Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature* 347, 69–72 (1990)
8. Bi, G.Q., Poo, M.M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience* 18(24), 10464–10472 (1998)
9. Froemke, R., Dan, Y.: Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature* 416, 433–438 (2002)
10. Sjöström, P., Turrigiano, G., Nelson, S.: Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* 32, 1149–1164 (2001)
11. Shouval, H., Bear, M., Cooper, L.: A unified model of NMDA receptor-dependent bidirectional synaptic plasticity. *PNAS* 99(16), 10831–10836 (2002)
12. Kurashige, H., Sakai, Y.: BCM-type synaptic plasticity model using a linear summation of calcium elevations as a sliding threshold. In: King, et al. (eds.) ICONIP 2006. LNCS, vol. 4232, pp. 19–29. Springer, Heidelberg (2006)
13. Badoval, M., Zou, Q., Davison, A., Rudolph, M., Bal, T., Fregnac, Y., Destexhe, A.: Biophysical and phenomenological models of multiple spike interactions in spike-timing dependent plasticity. *International Journal of Neural Systems* 16(2), 79–97 (2006)
14. Izhikevich, E., Desai, N.: Relating STDP to BCM. *Neural Computation* 15, 1511–1523 (2003)
15. Lu, B., Yamada, W., Berger, T.: Asymmetric synaptic plasticity based on arbitrary pre- and postsynaptic timing spikes using finite state model. In: Proceedings of International Joint Conference on Neural Networks (2007)
16. Gerstner, W., Kistler, W.: Spiking neuron models: single neurons, populations, plasticity. Cambridge University Press, Cambridge (2002)
17. Schreiter, J., Ramacher, U., Heittmann, A., Matolin, D., Schüffny, R.: Cellular pulse coupled neural network with adaptive weights for image segmentation and its VLSI implementation. In: Proceedings 16th International Symposium on Electronic Imaging: Science and Technology, vol. 5298, pp. 290–296 (2004)
18. Schemmel, J., Brüderle, D., Meier, K., Ostendorf, B.: Modeling synaptic plasticity within networks of highly accelerated I&F neurons. In: ISCAS 2007 (2007)

A Novel Hybrid Spiking Neuron: Response Analysis and Learning Potential

Sho Hashimoto and Hiroyuki Torikai

Graduate School of Engineering Science, Osaka University,
1-3 Machikaneyama, Toyonaka, Osaka, 560-8531 Japan
hashimoto@hopf.sys.es.osaka-u.ac.jp

Abstract. In this paper, we propose a hybrid spiking neuron which can exhibit various bifurcation phenomena and response characteristics of inter spike intervals. Using a discrete/continuous-states hybrid map, we can clarify typical bifurcation mechanisms and can analyze the response characteristics. In addition, we propose a learning algorithm of the hybrid spiking neuron and show that the neuron can approximate given response characteristics of inter spike intervals.

1 Introduction

Various spiking neuron models have been proposed based on analog nonlinear dynamics [1]-[3]. In addition, a digital spiking neuron (DSN) has been proposed based on discrete-state dynamics [4]-[6]. In this paper, first, we propose a hybrid spiking neuron (HSN) which is a generalized version of the DSN. We then derive a discrete/continuous-states hybrid map which can describe dynamics of the HSN without any approximation. Second, using the hybrid map, we can clarify typical bifurcation mechanisms of the HSN. Based on the bifurcation analysis, we can analyze the response characteristics of inter spike intervals. Third, we consider a learning problem. A HSN (teacher) exhibits various responses for a stimulation input. We give some responses as teacher signals to another HSN (student). We then propose a learning algorithm and show that the student HSN can approximate the response characteristics of the teacher HSN. Novelties and significances of this paper include the following points. (a) The DSN has no stimulation input and is described by a discrete-state map [6]. On the other hand, the HSN has the stimulation input and is described by the discrete/continuous-states hybrid map. The hybridness enables the HSN to exhibit various bifurcations and response characteristics of the inter spike interval. (b) The HSN is a wired system of shift register generators and the learning algorithm is based on changes of the wiring pattern. Hence the HSN and the learning algorithm are suitable for implementation on a reconfigurable chip such as FPGA [7]. (c) Ref [6] presents a learning algorithm of the DSN for approximation of some given spike-trains. On the other hand, this paper presents the learning algorithm which enables the HSN to approximate a given response characteristics. Such a learning is important not only as a fundamental research topic but also to develop applications

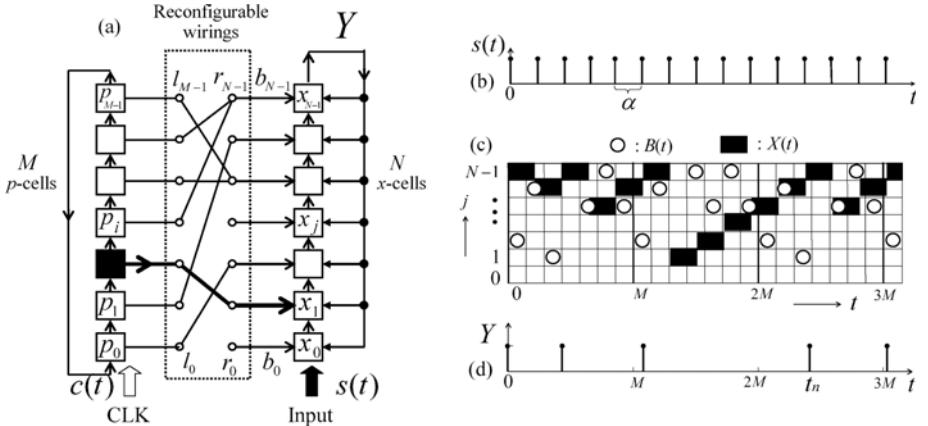


Fig. 1. (a) HSN model. (b) Stimulation input spike-train $s(t)$ whose period is α . (c) Time waveforms. (d) Output spike-trains $Y(t)$.

of the HSN., e.g., pulse-coupled neural network of the HSNs. (d) The approach of this paper will be extended for approximation of response characteristics of a biological neuron. Such an extended system will be applicable for neural prosthesis, e.g., substitution of a damaged biological neuron by an FPGA-implemented HSN which can autonomously obtain appropriate response characteristics. The results of paper will be fundamentals for such a future work.

2 Hybrid Spiking Neuron

2.1 Novel Hybrid Spiking Neuron

In this section we propose a *hybrid spiking neuron* (HSN) which has a stimulation input spike-train $s(t)$ and explain its basic behavior. As shown in Fig. 1(a), the HSN has M pieces of p -cells, where $M \geq 2$. Each p -cell has a digital state $p_i(t) \in \{0, 1\}$. One of the p -cells has a digital state "1", and the others have states "0". We use a positive integer $P(t) \in \{0, 1, \dots, M-1\}$ to indicate which p -cell has a state "1" at a time t , i.e., $p_i(t) = 1$ implies $P(t) = i$. The p -cells accept a internal clock $c(t)$ whose period is 1. Then the dynamics of $P(t)$ is governed by

$$P(t+1) = \begin{cases} P(t) + 1 & \text{if } P(t) \neq M-1, \\ 0 & \text{if } P(t) = M-1, \end{cases} \quad (1)$$

where $P(0) = 0$. $P(t)$ operates on a discrete time $t = 0, 1, 2, \dots$. Then $P(t)$ oscillates periodically with period M . As shown in Fig. 1(a), the HSN has one-way reconfigurable wirings from the left terminals (l_0, \dots, l_{M-1}) to the right terminals (r_0, \dots, r_{N-1}), where $N \geq 2$. We use a wiring function $A(i)$ to describe a wiring pattern: " $A(i) = j$ " means " l_i is wired to r_j ". In addition, we introduce the following representation method about the wiring function:

$A = (A(0), A(1), \dots, A(M - 1))$. The HSN in Fig. II(a) has the following wiring function: $A = (2, 5, 1, 6, 4, 6, 4)$. The right terminals (r_0, \dots, r_{N-1}) output a base signal $(b_0, b_1, \dots, b_{N-1})$. We use a positive integer $B(t) \in \{0, 1, \dots, N - 1\}$ to indicate which base signal b_j has a state "1" at a time t , i.e., $b_j(t) = 1$ implies $B(t) = j$. $B(t)$ is described by

$$B(t) = A(P(t)). \quad (2)$$

In Fig. II(c), white circles represent $B(t) = j$. The base signal $(b_0, b_1, \dots, b_{N-1})$ is input to the N x -cells. Each x -cell has a digital state $x_j(t) \in \{0, 1\}$. We use a positive integer $X(t) \in \{0, 1, \dots, N - 1\}$ to indicate which x -cell has a state "1" at a time t , i.e., $x_j(t) = 1$ implies $X(t) = j$. X may correspond to a membrane potential of a neuron model [3]. The x -cells accept the stimulation input spike-train $s(t)$ which is periodic with period α . Then dynamics of $X(t)$ is governed by

$$X(t + \alpha) = \begin{cases} X(t) + 1 & \text{if } X(t) \neq N - 1, \text{ (Shift)} \\ B(t + \alpha) & \text{if } X(t) = N - 1. \text{ (Reset)} \end{cases} \quad (3)$$

$X(t)$ operates on a discrete time $t = 0, \alpha, 2\alpha, \dots$. Basic Dynamics of $X(t)$ is shown in Fig. II(c). A black box at (t, j) represents $X(t) = j$. If the black box reaches the highest position at $t = t_n$ ($X(t_n) = N - 1$), the black box at $t = t_n + \alpha$ is reset to the position of the white circle at $t = t_n + \alpha$. At this reset moment, the HSN outputs a spike $Y(t_n) = 1$. Repeating such *shift-and-reset* dynamics, the HSN generates a spike train

$$Y(t) \equiv \begin{cases} 1 & \text{if } X(t) = N - 1, \quad t = 0, \alpha, 2\alpha, \dots \\ 0 & \text{if } X(t) \neq N - 1, \end{cases} \quad (4)$$

as shown in Fig. II(d). Y may correspond to a firing spike-train of a neuron model [3]. As a result, the dynamics of the HSN is described by Equations (1), (2), (3) and (4). The HSN is characterized by the parameters $M, N, A(i)$ and α .

Remark 1. We have analyzed dynamics of a digital spiking neuron (DSN) where the period of the stimulation input $s(t)$ is $\alpha = 1$, i.e., the stimulation input $s(t)$ to the x -cells is synchronized with the internal clock $c(t)$ to the p -cells. In this paper, we firstly formulate the dynamics of the HSN where $\alpha \neq 1$. The HSN can have richer dynamics than the DSN as we investigate in the following subsections.

2.2 Hybrid Phase Map

Let t_n be the n -th spike position as shown in Fig. II(c), where $n = 1, 2, 3, \dots$. The spike position t_n is governed by the following spike position map.

$$t_{n+1} = f(t_n) = t_n + \alpha(N - B(\text{Int}(t_n + \alpha \text{ (mod}M\text{))))). \quad (5)$$

Let the interval between t_n and t_{n+1} be $\Delta_n = t_{n+1} - t_n \in \{\alpha, 2\alpha, \dots, M\alpha\} = \Delta$. We refer to Δ_n as *inter spike interval* (ISI). The ISI Δ_n is given by

$$\Delta_n = \alpha(N - B(\text{Int}(t_n + \alpha \text{ (mod}M\text{))))). \quad (6)$$

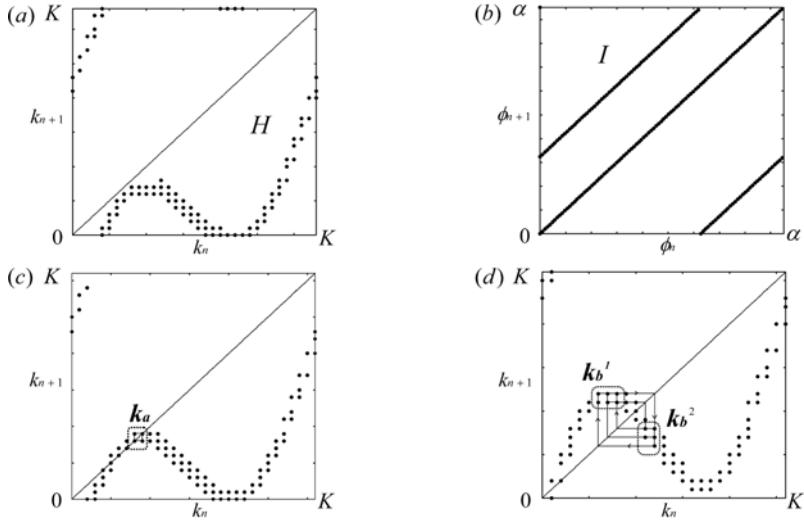


Fig. 2. (a) The map H . Period of the stimulation input is $\alpha \simeq 0.95$. (b) The map I for $\alpha \simeq 0.95$. (c) The map H for $\alpha \simeq 1.02$. (d) The map H for $\alpha \simeq 1.21$.

By using the period M of the base signal $B(t)$, the spike phase $\theta_n = t_n \pmod{M}$ is defined. The spike phase θ_n can be expressed by using an integer value $k_n \in \{0, 1, \dots, K = \text{Int}(M/\alpha)\} = \mathbf{K}$ and a continuous value $\phi_n \in [0, \alpha) = \Phi$ as follows.

$$\theta_n = \alpha k_n + \phi_n. \quad (7)$$

By using k_n and ϕ_n , the ISI Δ_n can be described by

$$\Delta_n = G(k_n, \phi_n) = \alpha(N - B(\text{Int}(\alpha(k_n + 1) + \phi_n))), \quad G : \mathbf{K} \times \Phi \rightarrow \Delta. \quad (8)$$

Then the dynamics of k_n and ϕ_n is governed by the following maps.

$$\begin{aligned} k_{n+1} &= H(k_n, \phi_n) = \text{Int}((\alpha k_n + \phi_n + \Delta_n)/\alpha), \\ \phi_{n+1} &= I(k_n, \phi_n) = \alpha k_n + \phi_n + \Delta_n \pmod{\alpha}, \\ H &: \mathbf{K} \times \Phi \rightarrow \mathbf{K}, I : \mathbf{K} \times \Phi \rightarrow \Phi. \end{aligned} \quad (9)$$

Fig. 2(a) and (b) show examples of the maps H and I , respectively.

Remark 2. The dynamics of the HSN can be governed by the hybrid map (H, I) defined by Equation (9) without any approximation. By using the hybrid map (H, I) , we analyze response characteristics of the HSN in the next subsection.

2.3 Bifurcation and Response Characteristics

In order to simplify the consideration, the number of p -cells and x -cells are assumed to be identical: $M = N$. Then we refer to M as *system size* hereafter.

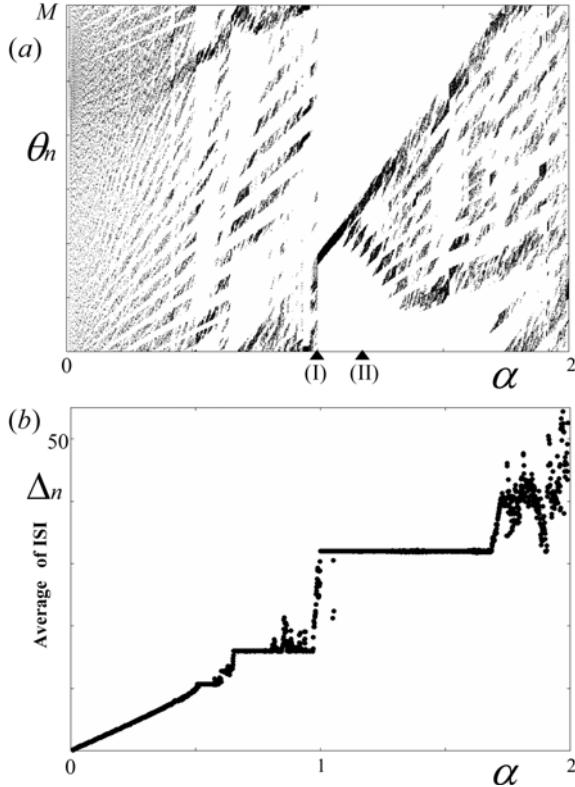


Fig. 3. (a) Bifurcation diagram of the spike phase. (b) Response characteristics (average of the ISI).

We use a sinusoidal-like base signal $B(t)$ which is defined by the following wiring function.

$$A^*(i) = \text{Int}((11 \sin(2\pi i/M) + 11) + 0.5), \quad i = 0, 1, \dots, M - 1. \quad (10)$$

Fig. 3(a) shows a bifurcation diagram of the spike phase $\theta_n = \alpha k_n + \phi_n$. We focus on the two bifurcations at the point $\alpha \simeq 1$ (bifurcation point (I)) and $\alpha \simeq 1.1$ (bifurcation point (II)). The mechanisms of the bifurcations can be explained as follows.

- The bifurcation at (I) is similar to the tangent bifurcation of a continuous state map [8]. Fig. 2(a) shows the map H just before the bifurcation point (I) and Fig. 2(c) shows the map H just after the point (I). As shown in Fig. 2(a), the band of the map H does not cross the diagonal line $k_{n+1} = k_n$ before the bifurcation point (I). However as shown in Fig. 2(c), the band of the map H is tangent to the diagonal line $k_{n+1} = k_n$ after the bifurcation point (I). Then the map H has a closed orbit composed of the set \mathbf{k}_a of points. The set \mathbf{k}_a corresponds to a fixed point of a continuous state map.

- The bifurcation at (II) is similar to the period-doubling bifurcation of a continuous state map [8]. As shown in Fig. 2(d), the orbit of k_n visits alternatively two sets \mathbf{k}_b^1 and \mathbf{k}_b^2 of points after bifurcation point (II). The two sets \mathbf{k}_b^1 and \mathbf{k}_b^2 correspond to two-periodic points of a continuous state map. What it comes down to is that the change of the orbit (i.e., $\mathbf{k}_a \rightarrow (\mathbf{k}_b^1, \mathbf{k}_b^2)$) corresponds to the period-doubling bifurcation of a continuous state map.

Fig. 3(b) shows a characteristics of the average of ISIs. We refer to the characteristics as response of the HSN to the stimulation input $s(t)$. Based on the above mentioned bifurcation mechanisms, we can give the following explanations.

- We can see that the response characteristics in Fig. 3(b) shows discontinuity at the bifurcation point (I). When the band of the map H in Fig. 2(a) does not cross the diagonal line $k_{n+1} = k_n$, there is no solution corresponding to \mathbf{k}_a in Fig. 2(c). However the solution \mathbf{k}_a exists when the map H is tangent to the diagonal line $k_{n+1} = k_n$. The existence of the solution \mathbf{k}_a discontinues at the bifurcation point (I). This discontinuity causes the discontinuity of the response characteristics in Fig. 3(b).
- The response characteristics does not show such discontinuity at the bifurcation point (II). A qualitative explanation is as the following. The set \mathbf{k}_a of points corresponds to one kind of ISI Δ_a . The two sets \mathbf{k}_b^1 and \mathbf{k}_b^2 of points correspond to two kinds of ISIs Δ_b^1 and Δ_b^2 . By a nature of the period-doubling bifurcation, we suppose $(\mathbf{k}_b^1 + \mathbf{k}_b^2)/2 = \mathbf{k}_a$. In addition, we assume that the relation between \mathbf{k}_a and Δ_a is approximately linear. Then we obtain $(\Delta_b^1 + \Delta_b^2)/2 = \Delta_a$. This means that the response characteristics does not change at the bifurcation point (II).

Remark 3. By adjusting the wiring function $A(i)$, we can realize various response characteristics of ISIs. Note that the response characteristics is one of the most important factors to characterize encoding functions of neuron models. We consider an inverse problem in the next section, i.e., approximation of a given response characteristics by a learning of the HSN.

3 Learning

In this section, we consider a fundamental question: *can an HSN approximate the response characteristics of another HSN (teacher) by dynamically adjusting the wiring function A ?* In order to simplify the consideration, the system size M is assumed to be $M = 8$. Then we use the following \tilde{A} as a teacher wiring function.

$$\tilde{A} = (3, 1, 0, 1, 3, 5, 6, 5). \quad (11)$$

where the tilde “~” represents “teacher” hereafter. Fig. 4(a) shows the response characteristics of the teacher. We select three kinds of stimulation inputs $s^1(t)$, $s^2(t)$ and $s^3(t)$, whose periods are α^1 , α^2 and α^3 . We input these stimulation inputs $s^1(t)$, $s^2(t)$ and $s^3(t)$ to the teacher HSN and sample 30 ISIs respectively. Let these ISI sequences be $\tilde{\Delta}^1$, $\tilde{\Delta}^2$, and $\tilde{\Delta}^3$. We use the student HSN whose system size is $M = 8$ and let its wiring function be represented by A^{old} . We

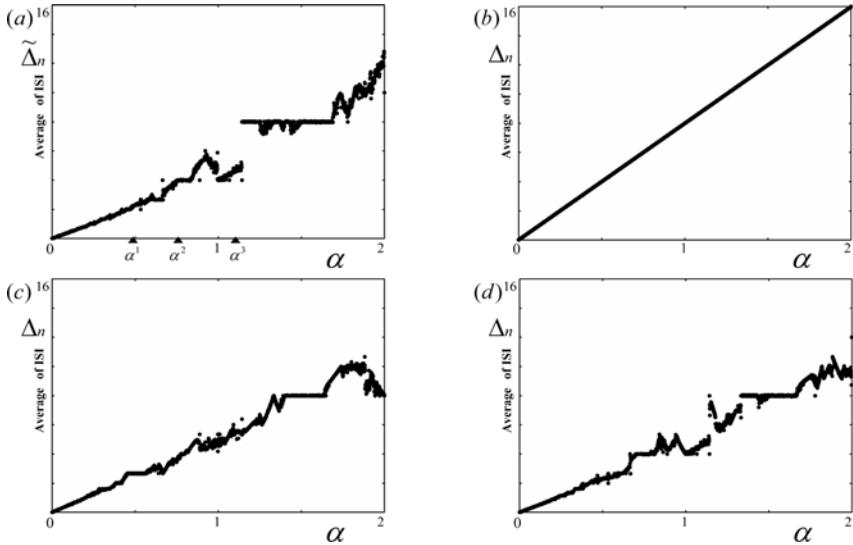


Fig. 4. (a) Response characteristics of the teacher. The teacher wiring function is characterized by Equation (11). (b) Response characteristics of the student just after initialization ($k = 0$). The cost function is $C = 292$ and the wiring function is $A^{old} = (0, 0, 0, 0, 0, 0, 0)$. (c) Response characteristics of the student after $k = 20000$ learning iterations. The cost function becomes $C = 82$ and the wiring function is $A^{old} = (3, 3, 2, 3, 3, 7, 5, 3)$. (d) Response characteristics of the student after $k = 50000$ learning iterations. The cost function becomes $C = 35$ and the wiring function is $A^{old} = (3, 1, 3, 1, 3, 5, 6, 3)$.

input $s^1(t)$, $s^2(t)$ and $s^3(t)$ to the student HSN and sample 30 ISIs respectively. Let these ISI sequences be Δ^1 , Δ^2 , and Δ^3 . Then we apply the following learning algorithm to the student HSN.

Step1. initialization: Initialize the wiring function as $A^{old} := (0, 0, \dots, 0)$.

Step2. re-wiring: Change one of the wirings (i.e., one of the elements in A^{old}) randomly and let the new wiring function be A^{new} .

Step3. selection: We propose the following cost function which measures degree of difference between the teacher ISI sequences and the student ISI sequences:

$$C = \sum_{n=1}^{30} \left(\frac{\|\tilde{\Delta}_n^1 - \Delta_n^1\|}{\alpha^1} + \frac{\|\tilde{\Delta}_n^2 - \Delta_n^2\|}{\alpha^2} + \frac{\|\tilde{\Delta}_n^3 - \Delta_n^3\|}{\alpha^3} \right). \quad (12)$$

If $C^{old} \leq C^{new}$ then go to Step 4. If $C^{old} > C^{new}$ then go to Step 5.

Step4. update: Update the wiring function as $A^{old} := A^{new}$. Go to Step 5.

Step5. termination: Let K be a given maximum iteration number. Increment the counter k by one. If $k < K$, then go to Step 2. If $k \geq K$, then terminate the algorithm.

Fig. 4(b) shows the response characteristics of the student just after the initialization in Step 1 of the learning algorithm, where the wiring function is $A^{old} = (0, 0, 0, 0, 0, 0, 0, 0)$. In this case the cost function is $C = 292$ and the response characteristics of the student in Fig. 4(b) is totally different from that of the teacher in Fig. 4(a). Fig. 4(c) shows the response characteristics after $k=20000$ learning iterations. In this case the cost function becomes $C = 82$ and the wiring function is $A^{old} = (3, 3, 2, 3, 3, 7, 5, 3)$. The response characteristics of the student in Fig. 4(c) looks a little like that of the teacher. Fig. 4(d) shows the response characteristics after $k=50000$ learning iterations. In this case the cost function becomes $C = 35$ and the wiring function is $A^{old} = (3, 1, 3, 1, 3, 5, 6, 3)$. The response characteristics of the student in Fig. 4(d) is similar to that of the teacher well. We can conclude that the learning algorithm can approximate the characteristics of ISI with some error. A quantitative analysis of the approximation error is a future problem.

4 Conclusions

We have proposed the hybrid spiking neuron (HSN). We have explained bifurcation mechanisms of the HSN and analyzed the response characteristics by using the discrete/continuous-states hybrid map. In addition, we have proposed the learning algorithm and shown that the student HSN can approximate the response characteristics of the teacher HSN. Future problems include: (a) detailed analysis of the hybrid map, (b) detailed analysis to the response characteristic, (c) development of the learning algorithm for better approximation, and (d) FPGA-implementation of the HSN and the learning algorithm. The authors would like to thank Professor Toshimitsu Ushio of Osaka Univ. for valuable discussions.

References

1. Izhikevich, E.M.: *Dynamical systems in neuroscience*. MIT Press, Cambridge (2006)
2. Izhikevich, E.M.: Which model to use for cortical spiking neurons? *IEEE Transactions on Neuronal Networks* 15(5), 1063–1069 (2004)
3. Nakano, H., Saito, T.: Grouping synchronization in a pulse-coupled network of chaotic spiking oscillators. *IEEE Trans. Neural Networks* 15(5), 1018–1026 (2004)
4. Torikai, H., Hamanaka, H., Saito, T.: Reconfigurable Digital Spiking Neuron and its Pulse-Coupled Network: Basic Characteristics and Potential Applications. *IEEE Trans. CAS* 53(8), 734–738 (2006)
5. Torikai, H., et al.: Digital spiking neuron and its learning for approximation of various spike-trains. *Neural Networks* (2008), doi:10.1016/j.neunet.2007.12.045
6. Torikai, H., Hashimoto, S.: A Hardware-oriented Learning Algorithm for a Digital Spiking Neuron. In: Proc. IEEE-INNS/ IJCNN (2008)
7. FPGA and HDL software package can be obtained, <http://www.xilinx.com/>
8. Devaney, R.L.: *An Introduction to Chaotic Dynamical Systems*, 2nd edn. Addison-Wesley Publishing Company, Reading (1995)

Event-Related Desynchronisation/Synchronisation of Spontaneous Motor Actions

Somnuk Phon-Amnuaisuk

Special Interest Group: Perception and Simulation of Intelligent Behaviours,
Faculty of Information Technology, Multimedia University,
Jln Multimedia, 63100 Cyberjaya, Selangor Darul Ehsan, Malaysia
somnuk.amnuaisuk@mmu.edu.my

Abstract. Event-related potentials (ERP) reflect external/internal stimuli. It is of our interest in the current study to investigate ERPs from spontaneous motor actions. Spontaneous motor actions are voluntary motor tasks carried out at an individual's own pace. Various properties of spontaneous motor actions are revealed through event-related desynchronisation and event-related synchronisation. These properties are reported and discussed in this paper.

1 Introduction

Electroencephalogram (EEG) is a potential generated from the sum of the excitatory and inhibitory post-synaptic potential of neurons. Therefore, event-related potentials (ERP) may manifest mental activities. Recent progress in brain computer interface (BCI) has been very promising. Researchers have successfully shown that appropriate features extracted from EEG signals (aka brain wave) could be used to signify intentions of an individual. It is exciting to imagine that we may, in a few decades, use our brain wave in neuroprosthetics, edutainments and even in military applications.

However, using brain wave signals for BCI applications is not a trivial matter. The challenge is that an EEG signal measured from the scalp has a very low signal to noise ratio. This makes accurate interpretation of EEG signal very challenging. Although recent dichotomise of EEG signals for cued left and right motor tasks could have up to 100 percent accuracy (experiments done with healthy subjects) when the signal is properly prepared and appropriate features are selected. Noncue-based responses have been somewhat neglected by researchers. It is interesting to note that most actions are driven by internal intentions and not by external cues. Therefore, it would be interesting to study features and other characteristics of noncue-based spontaneous motor tasks (see [5]).

In this study, we investigate ERP, ERD and ERS patterns observed from a subject engaged in a spontaneous motor task. While ERP gives an overall EEG response, ERD and ERS reveal changes in power which could be desynchronised (i.e., power decreases during motor actions) or synchronised (i.e., power increases

during motor actions). Here, we focus on left and right hand click-hold-release tasks. We organise the presentation according to the following topics: (i) Introduction, (ii) Background, (iii) Materials and Methods, (iv) Analysis and (v) Conclusion.

2 Background

The EEG signal is generated by neuronal dipoles created within elongated dendritic fields (e.g., cortical pyramidal cells). During stimuli, the neurons fire and the flow of current creates a dipole which exhibits oscillatory potentials. This signal could be monitored using non-invasive electrodes although it has a low signal to noise ratio.

It is known that the *Parietal* cortex area plays a major role in motor planning and actions. It is therefore recommended that electrodes are placed in the parietal area (i.e., C3, C4 in the standard 10-20 nomenclature) for EEG data acquisition of motor related tasks.

Previous studies in motor ERP reveal that when an individual's attention is directed towards the planning of actions, a negative deflection is observed from ERP [1], [3]. This negative deflection is termed *contingent negative variation* (CNV) [1], [8]. CNV is believed to be an expression of a preparation process for the actions. The late CNV wave is also believed to be of the same class with readiness potential (RP) [4].

While ERP reveals synaptical responses, ERD/ERS reveals power changes in a specific frequency band. It is believed that this change in power originates from the desynchronisation/synchronisation of activities of underlying neurons. In other words, ERPs represent the summed responses of cortical neurons while ERD/ERS represents changes in interactions between cortical neurons. ERD/ERS is therefore topological and frequency specific. Pfurtscheller suggests that ERD indicates the amount of attention paid to the tasks in the memory task. The higher the attention paid to the task (this may be inferred that complex task requires more attention) the higher the ERD magnitude [6].

3 Materials and Methods

3.1 Experimental Setup

Spontaneous motor tasks. A healthy male subject aged 45 participated in the two-hour session experiment. Four bipolar EEG channels were acquired on the following positions: C3, C4, Fz and Cz (in 10-20 nomenclature). The subject was seated in a comfortable position with eyes open, approximately 0.5 meter from the computer screen. The subject was instructed to look at a blank computer screen. The hair cross would appear on the screen for 7.5 seconds and disappear for 2 seconds. During the hair cross appearance on screen, the subject performed the following task at his own pace (i) use the left index finger to move the mouse (using mouse pad) and double click on the target, hold the mouse for a while

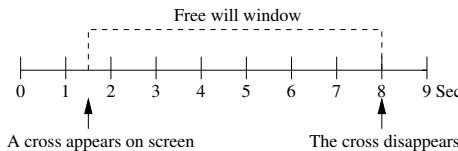


Fig. 1. Flow of experiment

(approx. 1-3 seconds) then release; (ii) use the right index finger to perform the same task. The subject was to choose the left or right hand at his own discretion. Figure 1 below shows the timing diagram of the experiment.

Rationale for the study. Most realistic BCI applications are not cue-driven but intention driven. That is, the needs to move limbs (e.g., in neuroprosthetics applications) are not cued, but are from an individual's intention. This experiment is set up to investigate the patterns in this kind of voluntary free pace motor actions. An understanding of the nature of the task would give us a better idea of how to process the information acquired from the task.

3.2 Data Acquisition and Processing

Four EEG data channels were acquired from four areas C3, C4, Fz and Cz. A single ground electrode was affixed at FPz. The cup electrodes were used and placed in a bipolar fashion, for example, channel C3 was acquired from two electrodes, one positive electrode connected 2.5 cm from C3 toward FC3 and one negative electrode was connected 2.5 cm from C3 toward CP3. The hair around the area was shaved. The impedance of each electrode was monitored before and after the sessions to confirm that the impedance was lower than 2.5Kohm . The specifications of the acquisition hardware claim that C3 and C4 acquired the EEG data with 256 Hz sampling rate, 0.5 to 100 Hz bandpass and with a sensitivity of $+/- 500\mu\text{V}$, while Fz and Cz acquired EEG data with the same sampling rate, but with 0.01 to 100 Hz bandpass and with a sensitivity of $+/- 2000\mu\text{V}$.

We carried out twelve sessions for a spontaneous motor action task. The number of actions from left hand and right hand actions in each session are not equal. Raw EEG data were inspected and contaminated data were removed from the data set. After removing contaminated epochs, 250 left hand and 220 right hand epochs were saved for further evaluation. In our evaluation, EEGLAB [2] package is employed for the evaluation of ERP. The evaluation of ERD/ERS is implemented in MATLAB. Detailed calculations of ERP, ERD/ERS are given below.

Calculate ERP and ERD/ERS. ERP is the average of all M epochs (see equation 1). $X_{ERP}^n(i,j)$ denotes a vector of length n from EEG channel i and from epoch j . ERD/ERS is also the average of all epochs but for a specific frequency band f (see equation 2, 3 and 4). $P_{ERD/S}^n(i,j)$ denotes an average power of an EEG channel in a specific frequency band. $X_{ERD/S}^n(i,j)$ shows the

changes in power with respect to a reference P_{ref} . In this experiment P_{ref} for press, hold and release (see figure 2, 3 and 4 respectively) are $P_{ERD/S}$ taken from (i) one second before the epoch (ii) the whole epoch and (iii) one second after the epoch respectively.

$$X_{ERP}^n(i, j) = \frac{1}{M} \sum_{j=1}^M X^n(i, j) \quad (1)$$

$$X_f^n(i, j) = \text{bandpass}(X^n(i, j)) \quad (2)$$

$$P_{ERD/S}^n(i, j) = \frac{1}{M} \sum_{j=1}^M (X_f^n(i, j))^2 \quad (3)$$

$$X_{ERD/S}^n(i, j) = \frac{P_{ERD/S}^n(i, j) - P_{ref}^n}{P_{ref}^n} \quad (4)$$

4 Analysis

In this section, we provide descriptive analyses of ERP and ERD/ERS in three scenarios: (i) at the press time, (ii) during the press and hold and (iii) at the

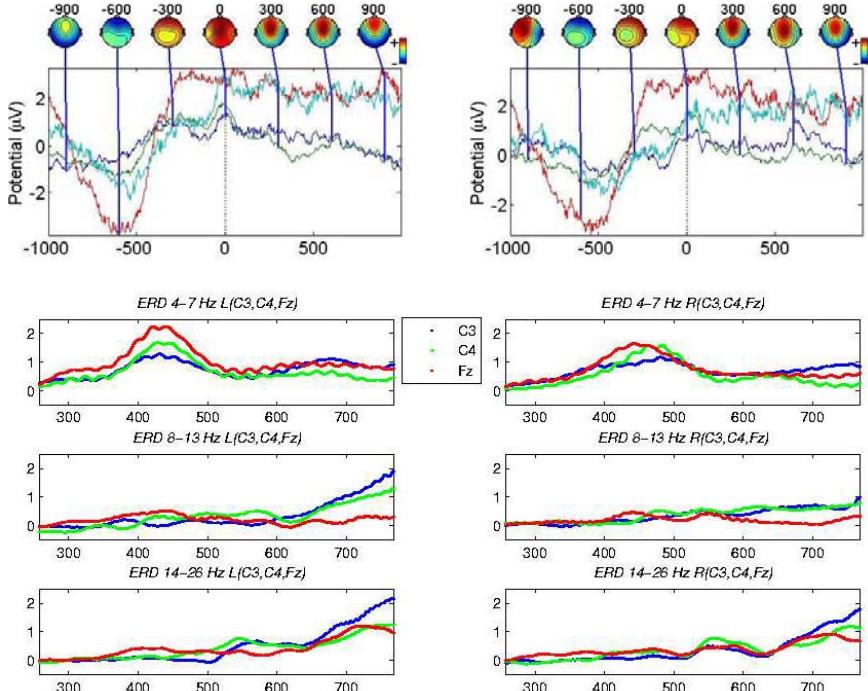


Fig. 2. Averaged ERP and ERD/ERS for left and right fingers action (press and hold at 0 sec, correspond to ERD/ERS sample number 512)

release time. Since the actions here are voluntary but free pace, for scenario (i), a time-lock response from one second before and one second after the press is shown, for scenario (iii) a time-lock response from one second before and two seconds after releasing the press is shown. Also note that the label for x-axis for ERD/ERS is in sample numbers; y-axis is the power changes ratio, and scale of 1 means 100% increment from P_{ref} .

Figure 2 (top most) shows outstanding negative deflections in all the channels (color codes: C3-blue, C4-green, Fz-red and Cz-cyan). Their corresponding ERD/ERS for the following frequency bands are provided: theta, alpha and beta (4-7 Hz, 8-13 Hz and 14-26 Hz respectively). There is a clear readiness potential 500 ms before the onset of the pressing of the buttons and it is most pronounced for Fz. However, no clear differences between the responses from the left and the right hand actions are observed.

A clear synchronisation in alpha and beta bands are observed in C3 and C4. This is in contrast to the ERD/ERS from the theta band where the synchronisation of all the channels is most pronounced 500 ms before the onset time (Figure 2, second row).

Figure 3 shows both ERP and ERD/ERS during the hold. There is no clear pattern in ERP data but ERD/ERS shows a transition from ERD to ERS in

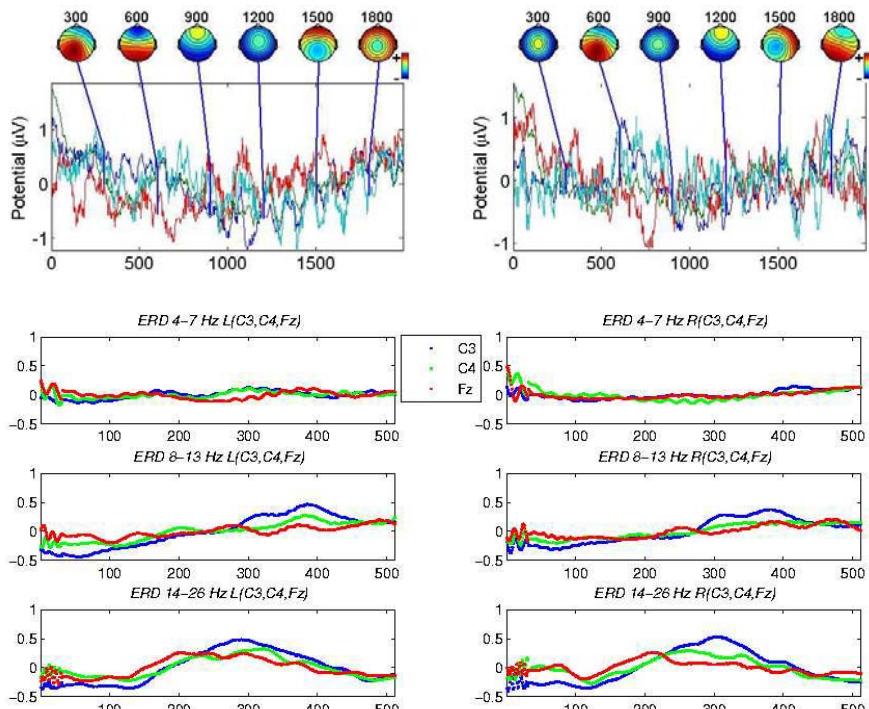


Fig. 3. ERP and ERD/ERS for left and right fingers (hold the press)

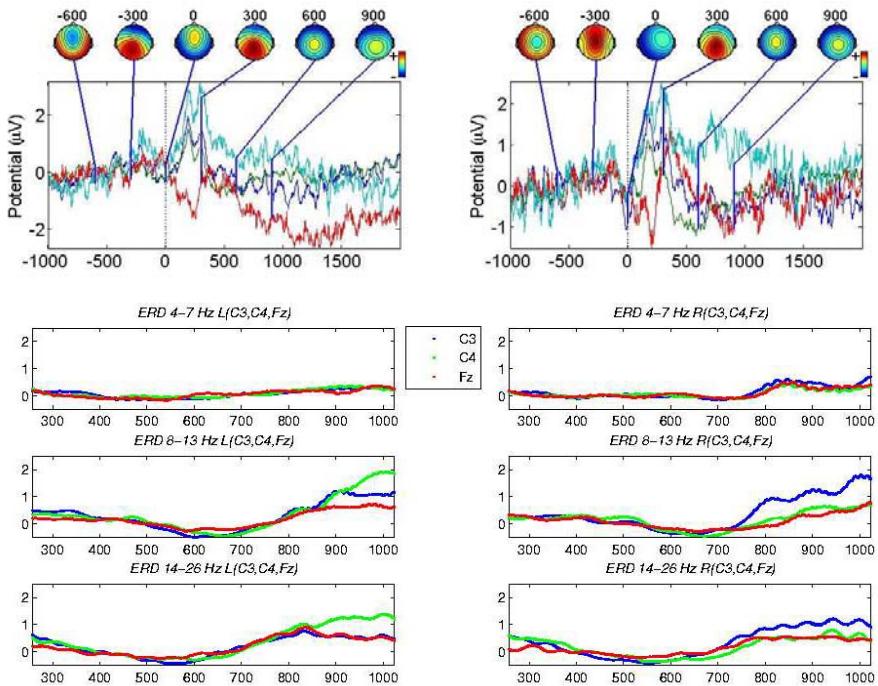


Fig. 4. Averaged EEG data for left and right finger actions (release the hold at 0 sec, correspond to ERD sample number 512)

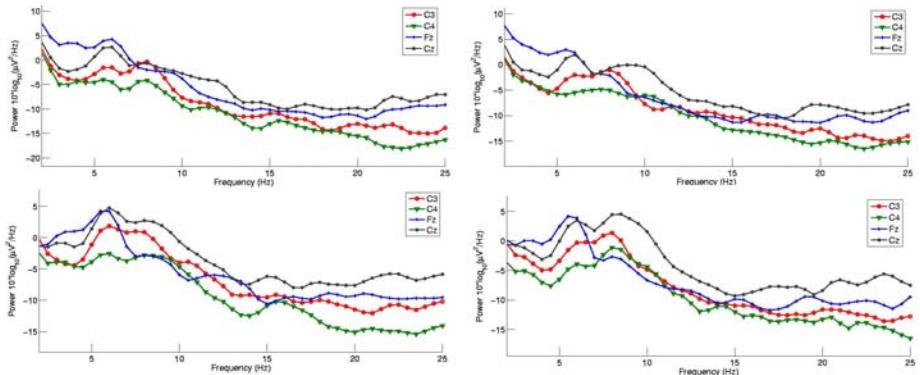


Fig. 5. Averaged EEG spectrum (top-left: Left before press; bottom-left: Left after press; top-right: Right before press; bottom-right: Right after press)

alpha and beta band. This occurs around sample number 256 (i.e., 1 second after pressing) in the alpha band and around sample number 190 (i.e., 0.75 second) in the beta band.

Figure 4 shows both ERP and ERD/ERS just before and after the release of the press action. It is interesting to observe that the ERP amplitude of Fz deflects in the opposite direction from the ERP from C3, C4 and Cz. This opposite deflection is clearly observed for about 300 ms after releasing the hold. There is also a momentary ERD on the alpha and beta bands before transforming to ERS. On the contrary, there is not much activity in the theta band.

5 Discussion and Conclusion

In this report, we investigate spontaneous event-related motor potential. It is believed that CNV before an action is an expression of motor preparation process. This is a complex process and we probably do not have a clear picture of the whole process at the moment. From this study, we suggest the following interpretations from the data:

1. Before press and hold (see figure 2): according to the flow of the experiment, the subject was free to press the button after seeing a cross hair appear on the computer screen. A significant decrease in μ rhythm on Fz could be observed from the averaged FFT spectrum from both left and right hand movements (see figure 5). ERD/ERS also showed an incremental power changes especially after the onset of the press. Contralateral ERD was not clear here. This could be from the fact that the task to be performed was too simple therefore no significant difference between the ERD from C3 and C4. It is interesting to note that a clear surge in ERD for all channels was observed in the theta band just before the onset of the action.
2. Holding the press (see figure 3): ERD/ERS also showed an incremental power changes after the onset of the press in alpha and beta bands. The ERD/ERS of the theta band remained constant in all channels.
3. Releasing the hold (see figure 4): There was a momentary ERD around the onset of the release. This was followed by an ERS. Contrary to the onset of the press action, no surge of power in theta band was observed, The ERD/ERS of the theta band remained constant in all channels.

Acknowledgments. This material is based upon work supported by the Ministry of Science, Technology and Innovation of Malaysia (MOSTI) under Grant No. 01-02-01-SF0071.

References

1. Deecke, L., Scheid, P., Kornhuber, H.H.: Distribution of readiness potential, pre-motion positivity and motor potential of the human cerebral cortex preceding voluntary finger movements. *Journal of Experimental Brain Research* 7(2), 158–168 (1969)
2. Delorme, A., Makeig, S.: EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods* 134(1), 9–21 (2004)

3. Jankelowitz, S.K., Colebatch, J.G.: Movement-related potential associated with self-paced, cued and imaginary arm movements. *Journal of Experimental Brain Research* 147, 98–107 (2002)
4. Leuthold, H., Sommer, W., Ulrich, R.: Preparing for action: Inferences from CNV and LRP. *Journal of Psychophysiology* 18, 77–88 (2004)
5. Millan, J., del, R., Franzé, M., Mourão, J., Cincotti, F., Babiloni, F.: Relevant EEG features for the classification of spontaneous motor-related tasks. *Biological Cybernetics* 86, 89–95 (2002)
6. Pfurtscheller, G., Lopes da Silva, F.H.: Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology* 110(11), 1842–1857 (1999)
7. Riecker, A., Wildgruber, D., Mathiak, K., Grodd, W., Ackermann, H.: Parametric analysis of rate-dependent hemodynamic response functions of cortical and subcortical brain structures during auditorily cued finger tapping: a fMRI study. *NeuroImage* 18, 731–739 (2003)
8. Zani, A., Proverbio, A.M.: Electromagnetic manifestations of mind and brain. In: Zani, A., Proverbio, A.M. (eds.) *The cognitive electrophysiology of mind and brain*. Academic Press, London (2003)

Competition between Synapses Located in Proximal and Distal Dendrites of the Dentate Granule Cell through STDP

Yukihiro Nonaka and Hatsuo Hayashi

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
nonaka-yukihiro@edu.brain.kyutech.ac.jp,
hayashi@brain.kyutech.ac.jp

Abstract. We investigated competition between synapses located proximal and distal dendrites through STDP learning rules using a dentate granule cell model. The proximal dendrite and the distal dendrite were stimulated by a regular pulse train and a random pulse train respectively. When both synapses were subject to asymmetric STDP rules, the distal synapse was not enhanced but the proximal synapse was enhanced through synaptic competition. This competition was caused when the integral of the LTD window was larger than that of the LTP window of the STDP rule in the distal dendrite. In contrast, when the proximal synapse was not subject to the asymmetric STDP rule but a Mexican-hat STDP rule, location of the synapse enhanced by synaptic competition was switched from the proximal synapse to the distal synapse. We also examined the role of inhibitory interneurons in the synaptic competition.

1 Introduction

When afferent fibers conveying different information make synapses on dendrites of target neurons, the network structure is often organized well in the hippocampus and the neocortex. For example, the lateral and the medial perforant pathways conveying spatial and non-spatial information terminate within the outer 1/3 and the middle 1/3 of dendrites of dentate granule cells respectively [1, 2]. On the other hand, most synapses in the hippocampus and the cortex are characterized by spike-timing-dependent plasticity (STDP) [3, 4]. Synapses on the distal dendrites of the dentate granule cells are subject to an asymmetric STDP rule as well [5]. Although the learning rule of synapses on the proximal dendrites of the granule cells has not been elucidated, it would be plausible to assume that those synapses are also subject to a STDP rule, because most proximal synapses are subject to a STDP rule in the hippocampus and the cortex. The shape of the STDP curve depends on dendritic location. For example, in the cortex layer II/III pyramidal cells, the ratio of the integral of the LTD window to that of the LTP window (LTD/LTP) is larger at distal synapses than at proximal synapses [6]. Proximal synapses of the hippocampal CA1 pyramidal cells are subject to a Mexican-hat STDP rule, while distal synapses are

subject to an asymmetric STDP rule [7, 8]. Location of synapses receiving signals of different information is functionally organized on the dendrite, and the STDP learning rule often depends on the distance from the soma. However, little is known about competition between those synapses and the functional role of the competition.

In this study, we investigated synaptic competition between proximal and distal synapses of the dentate granule cell, which were supposed to receive spatial and non-spatial information respectively, using a multicompartmental granule cell model. We also examined the influence of inhibitory interneurons, i.e. the MOPP cell and the basket cell, to the synaptic competition.

2 Methods

A 4-compartmental model of the dentate granule cell was developed by modifying a 47-compartmental model developed by Aradi and Holmes [9]. Equations of the Aradi-Holmes model are as follows:

$$\begin{aligned} C_m \frac{dV_i}{dt} = & g_{Na} \cdot m^3 \cdot h \cdot (E_{Na} - V_i) + g_{fK_{DR}} \cdot n_f^4 \cdot (E_K - V_i) + g_{sK_{DR}} \cdot n_s^4 \cdot (E_K - V_i) + g_{K_A} \cdot k \cdot l \cdot (E_K - V_i) \\ & + g_{TCa} \cdot a^2 \cdot b \cdot (E_{Ca} - V_i) + g_{NCa} \cdot c^2 \cdot d \cdot (E_{Ca} - V_i) + g_{LCa} \cdot e^2 \cdot (E_{Ca} - V_i) \\ & + g_{BK} \cdot r \cdot s^2 \cdot (E_K - V_i) + g_{SK} \cdot q^2 \cdot (E_K - V_i) + r_{i,i+1}(V_{i+1} - V_i) + r_{i,i-1}(V_{i-1} - V_i) \end{aligned} \quad (1)$$

$$\frac{dz}{dt} = \alpha_z - (\alpha_z + \beta_z)z \quad (2)$$

$$\frac{ds}{dt} = \frac{s_\infty - s}{\tau_s} \quad s_\infty = 1/(1 + 4/[Ca^{2+}]) \quad (3)$$

where the subscript i denotes the numbers of the compartments. Although nine ionic currents of the present model were the same as those on the right-hand side of Eq. 1, the number of compartments was reduced and parameters of those ionic channels were modified to reproduce experimental observations [11, 12]. $r_{i,i+1}$ ($r_{i,i-1}$) is the conductance between compartments, i and $i+1$ (i and $i-1$). The variable z in Eq. 2 stands for the gating variables, m , h , n , and so on. $[Ca^{2+}]$ is the intracellular calcium concentration produced by the T-, N-, and L-type calcium currents in Eq. 1 (see [9] for further details). The present model was composed of three dendrite compartments and one soma compartment, as shown in Fig. 1(a). Pulse trains were fed to the compartments, 1 and 2, through the lateral perforant path (LPP) and the medial perforant path (MPP) synapse respectively. The inhibitory interneuron MOPP cell (M) received pulses from LPP and MPP, and then inhibited the compartments, 1 and 2, through synapses mediated by GABA_B (Fig. 1(b)). The basket cell (BC) also received pulses from LPP and MPP, besides the feedback from the granule cell, and then inhibited the compartments, 1 and 2, through synapses mediated by GABA_A.

As neurons in the medial entorhinal cortex, which project axons, i.e. MPPs, to the dentate gyrus, generates a theta rhythm, it has been supposed that firing of neurons in the medial entorhinal cortex is paced by the theta rhythm. On the other hand, neurons in the lateral entorhinal cortex, which project axons, i.e. LPPs, to the dentate gyrus, do not generate a theta rhythm and fire randomly [10]. Therefore, in the present model, a

regular pulse train (interpulse interval = 125 ms, i.e. 8 Hz) was fed to the granule cell through the MPP synapse, and a random pulse train (mean interpulse interval = 10–250 ms, SD = 2.5–62.5 ms, Gaussian distribution) was fed to the granule cell through the LPP synapse. The standard deviation SD was 25% of the mean interpulse interval.

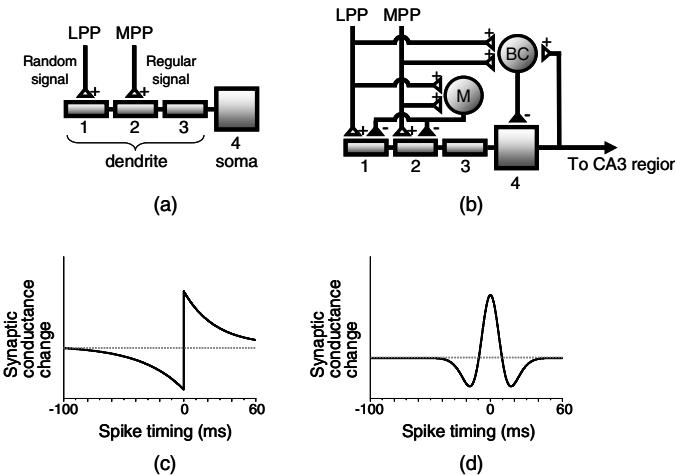


Fig. 1. Dentate granule cell model and STDP learning rules. (a, b) 4-compartmental model of the dentate granule cell. The lateral perforant path (LPP) and medial perforant path (MPP) synapses conveying random and regular pulse trains terminate at the compartments, 1 and 2, respectively. The MOPP cell (M) and the basket cell (BC) are included in (b). (c) Asymmetric STDP learning rule from Lin et al. [5]. The LPP synapse was subject to this type of STDP rule. (d) Mexican-hat STDP learning rule. The MPP synapse was subject to the asymmetric or the Mexican-hat STDP rule.

The asymmetric STDP learning rule that is an approximation of the synaptic conductance as a function of the spike-timing observed in the outer 1/3 of the dentate molecular layer by Lin et al. [5] is shown in Fig. 1(c). When presynaptic spikes precede the postsynaptic spikes within a time window, the synaptic conductance is enhanced. However, presynaptic spikes follow the postsynaptic spikes within a time window, the synaptic conductance is reduced. The ratio of the integral of the LTD window to that of the LTP window is approximately 1.2. The LPP synapse was subject to this type of STDP rule. A Mexican-hat STDP rule is shown in Fig. 1(d). Since the learning rule of the MPP synapse had not been elucidated, the asymmetric STDP rule or the Mexican-hat STDP rule was used for the MPP synapse. The spike-timing was measured as the difference between a backpropagating action potential (BAP) and an EPSP. Positive spike-timing was defined by the order, EPSP → BAP. Initial synaptic conductance was set so that a single EPSP fired no action potential though an integrated EPSP evoked by almost simultaneous MPP and LPP inputs fired an action potential.

3 Result

3.1 Competition between Synapses Located in Proximal and Distal Dendrites

A regular pulse train (8 Hz) and a random pulse train (mean interpulse interval = 100 ms, SD = 25 ms, Gaussian distribution) were fed to the granule cell through the MPP synapse and the LPP synapse respectively, using the granule cell model shown in Fig. 1(a). Both synapses were slightly enhanced at the initial stage (Fig. 2, S1). Then, the MPP synapse was enhanced at a relatively high grow-rate at the second stage (Fig. 2, S2), while the slight enhancement of the LPP synapse was maintained or rather depressed.

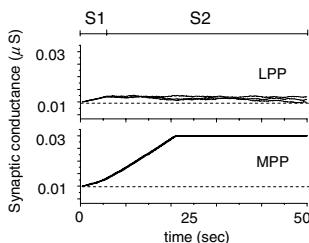


Fig. 2. Synaptic conductances. The LPP and MPP synapses being subject to the asymmetric STDP rule shown in Fig. 1(c) were stimulated by a random pulse train (mean frequency = 10 Hz) and a regular pulse train (8 Hz) respectively. The LPP synapse was not enhanced but the MPP synapse was enhanced finally through competition between the synapses. Three traces obtained using different initial conditions of the granule cell are superimposed. The dotted lines indicate the initial synaptic conductance ($0.01 \mu\text{S}$).

The spike-timings at LPP and MPP synapses were positive at the initial stage, because the granule cell was fired by an integrated EPSP evoked by occasional pairing of MPP and LPP pulses, although those synaptic inputs were uncorrelated. The spike-timing between EPSP and BAP as a function of the timing between the MPP and LPP pulses depends on the dendritic location, and most spike-timings at the MPP synapse were smaller than those at the LPP synapse. Enhancement of the MPP synapse was therefore relatively larger than that of the LPP synapse, and the granule cell was fired only by MPP pulses at the end of the initial stage. Then, the MPP synaptic conductance increased much faster at the second stage, because most spike-timings at the MPP synapse were positive. Spike-timings at the LPP synapse were, however, distributed from negative to positive due to non-correlation between the inputs through MPP and LPP synapses, and enhancement and depression counterbalanced each other. Consequently, the MPP synaptic conductance was enhanced, and the LPP synaptic conductance was maintained near the initial conductance or rather depressed.

When the mean frequency of random pulses through LPP was increased up to 40 Hz, the LPP synapse was not enhanced but the MPP synapse was enhanced as well as the above. When the mean frequency was above 50 Hz, temporally integrated EPSPs evoked at the LPP synapse were able to elicit action potentials, and these led to positive-spike timings. However, as the average interpulse interval was less than 20 ms, other

EPSPs existed within 10 ms after the firing and these EPSPs led to negative spike-timings. Enhancement and depression counterbalanced each other, and consequently the LPP synapse was not enhanced when the mean frequency was above 50 Hz.

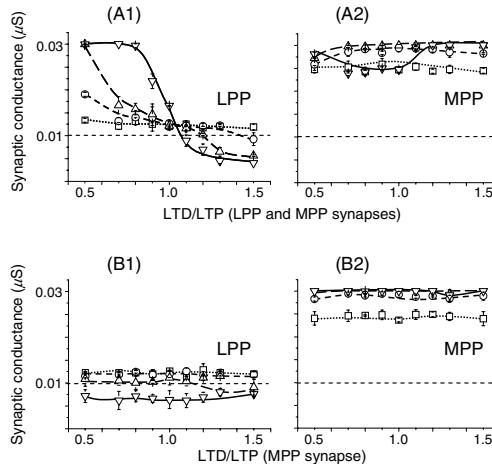


Fig. 3. Synaptic conductance as a function of the ratio LTD/LTP of the asymmetric STDP rule. (A1 and A2) Synaptic conductances of the LPP synapse and the MPP synapse respectively. The ratio LTD/LTP was varied simultaneously at the LPP and MPP synapses. (B1 and B2) The ratio LTD/LTP was set equal to 1.2 at the LPP synapse and varied at the MPP synapse. The LPP synapse and the MPP synapse were stimulated by a random pulse train and a regular pulse train respectively. The mean frequency of random pulse train fed to the LPP synapse was ∇ : 40, \triangle : 20, \circ : 10, or \square : 5 Hz. The frequency of the regular pulse train fed to the LPP synapse was 10 Hz. The dotted lines indicate the initial synaptic conductance (0.01 μ S).

3.2 Dependence of the Competition between Proximal and Distal Synapses on the Ratio of the Integral of the LTD Window to That of the LTP Window of the Asymmetric STDP Rule

In order to explore the conditions on which synaptic competition shown in the previous section occurs, the ratio of the integral of the LTD window to that of the LTP window (LTD/LTP) of the asymmetric STDP rule was varied simultaneously at LPP and MPP synapses (Fig. 3(A1) and (A2)). Enhancement of the MPP synapse caused after the competition did not depend on the ratio LTD/LTP (Fig. 3(A2)), although the LPP synapse was enhanced when the ratio LTD/LTP was less than one and the average frequency of random pulses through LPP was higher (Fig. 3(A1)). This suggests that the ratio LTD/LTP should be larger than one at least at distal synapses in order to cause the synaptic competition shown in the previous section. Actually, the ratio LTD/LTP of the STDP rule observed in distal dendrites by Lin et al. [5] is about 1.2. The ratio LTD/LTP was, therefore, fixed at 1.2 at the LPP synapse and the ratio LTD/LTP was varied at the MPP synapse (Fig. 3(B1) and (B2)). The LPP synapse was not enhanced but the MPP synapse was in the range of the ratio LTD/LTP between 0.5 and 1.5. This suggests that the synaptic competition occurs in

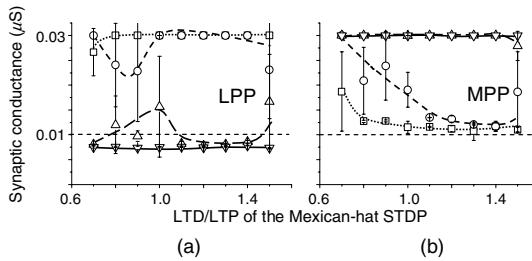


Fig. 4. Synaptic conductance as a function of the ratio LTD/LTP of the Mexican-hat STDP rule. The LPP synapse and the MPP synapse were subject to the asymmetric STDP rule and the Mexican-hat STDP rule respectively. The ratio LTD/LTP of the asymmetric STDP rule was set equal to 1.2. (a) Synaptic conductance of the LPP synapse. (b) Synaptic conductance of the MPP synapse. The LPP synapse was stimulated by a random pulse train; the mean frequency was ∇ : 40, \triangle : 20, \circ : 10, or \square : 5 Hz. The MPP synapse was stimulated by a 10 Hz regular pulse train. The dotted lines indicate the initial synaptic conductance ($0.01 \mu\text{S}$).

the dentate gyrus regardless of the ratio LTD/LTP of the asymmetric STDP rule at proximal dendrites if the ratio LTD/LTP is larger than one at distal dendrites.

It was then examined whether the shape of the STDP rule influenced the synaptic competition. The LPP and the MPP synapses were subject to the asymmetric and the Mexican-hat STDP rules shown in Fig. 1 (c) and (d) respectively. The ratio LTD/LTP of the Mexican-hat STDP rule was varied, while the ratio of the asymmetric STDP rule was set equal to 1.2. Location of the synapse enhanced by the competition was switched from the MPP synapse to the LPP synapse in the frequency range of the LPP input below 10 Hz when the ratio LTD/LTP of the Mexican-hat STDP rule was larger than one (Fig. 4; see symbols, \circ and \square). As the LTP window of the Mexican-hat STDP rule was narrow (about ± 12 ms) and a LTD window existed in the range of positive spike-timing, enhancement of the MPP synapse was relatively smaller than that of the LPP synapse, and finally the LPP synaptic conductance increased enough to fire action potentials. This suggests that the location of the synapse enhanced by the competition can be switched between the proximal and distal dendrites depending on the shape of the STDP rule for the proximal synapse. When the MPP synapse was still subject to the asymmetric STDP rule, the switching did not occur unless the integral of the LTP window was excessively small.

3.3 Role of the Inhibitory Interneurons in Competition between Proximal and Distal Synapses

Competition between proximal and distal synapses was caused when both proximal and distal synapses were subject to asymmetric STDP rules and the integral of the LTD window was larger than that of the LTP window. When the average frequency of random pulses fed to the LPP synapse was higher than 50 Hz, the firing rate of the granule cell increased (Fig. 5(a)) and a number of EPSPs also existed within 10 ms after firing of the granule cell. Therefore, the LPP synapse was not enhanced as mentioned above. However, the LPP synapse was depressed excessively because the ratio LTD/LTP was larger than one, and it is possible that integrated EPSPs cannot

fire the granule cell anymore. This implies that competition between proximal and distal synapses never occurs again.

The amplitude of EPSPs was frequently reduced by IPSPs evoked by feed-forward and feedback inhibition when the MOPP cell and the basket cell were included. The firing rate of the granule cell was also reduced below 5 Hz (Fig. 5(b)). Consequently, the LPP synapse was not depressed excessively by the high-frequency input.

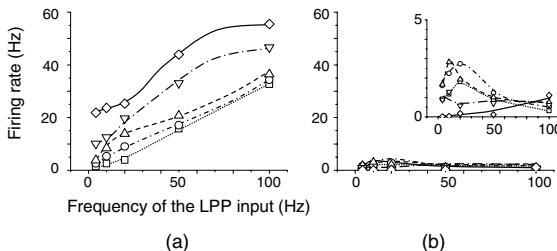


Fig. 5. Firing rate of the granule cell as a function of the mean frequency of the random pulse train fed to the LPP synapse. The standard deviation was 25% of the mean interpulse interval. (a) Without interneuron. The firing rate increased with increase in the mean frequency of the LPP input. (b) With MOPP cell and basket cell. The firing rate was reduced below 5 Hz. The ordinate is magnified in inset. The frequencies of the regular pulse train fed to the MPP synapse were \square : 4, \circ : 10, \triangle : 20, ∇ : 50, and \diamond : 100Hz.

4 Summary and Conclusions

LTP was not induced in the LPP synapse but in the MPP synapse through competition between those synapses being subject to asymmetric STDP rules when LTD/LTP was larger than one at the LPP synapse. It has been reported that the ratio LTD/LTP is about 1.2 in distal dendrites of the dentate granule cell [5]. It is therefore suggested that, in the dentate gyrus, proximal synapses receiving spatial information are enhanced in competition with the distal synapses receiving non-spatial information such as odor, and the distal synapses are not enhanced after the competition. This implies that the pathway of the spatial information is enhanced at synapses in proximal dendrites of dentate granule cells with the aid of non-spatial information, which is projected to distal dendrites of the granule cells. In other words, impressive environment, i.e. odor, color, shape, and so on, might emphasize the necessity for storing correlated place information.

In contrast, when the MPP synapse was subject to a Mexican-hat STDP rule, the MPP synapse was not enhanced but the LPP synapse was enhanced when the frequency of the LPP input was below 10 Hz. This suggests that the location of the synapses enhanced by the competition can be switched between the proximal and distal dendrites depending on the shape of the STDP rule. The shape of the STDP rule is actually able to change [7, 8]. When distal synapses are enhanced, place information stored in the hippocampus would be recalled by non-spatial information alone.

Feed-forward and feedback inhibition by dentate inhibitory interneurons, the MOPP cell and the basket cell, reduced the amplitude of EPSPs and the firing rate of

the granule cell as shown in this paper. In terms of the competition between proximal and distal synapses, the feed-forward and feedback inhibition prevented excessive depression of the LPP synapse induced by high-frequency stimulation. In other words, the inhibition may prevent a malfunction in the dentate gyrus when exposed to upstream convulsive activity, besides blockade of propagation of the convulsive activity to the hippocampus.

References

1. McNaughton, B.L.: Evidence for two physiologically distinct perforant pathways to the fascia dentate. *Brain Res.* 199, 1–19 (1980)
2. Hargreaves, E.L., Rao, G., Lee, I., Knierim, J.J.: Major dissociation between medial and lateral entorhinal input to dorsal hippocampus. *Science* 308, 1792–1794 (2005)
3. 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)
4. Feldman, D.E.: Timing-based LTP and LTD at vertical input to layer II/III pyramidal cells in rat barrel cortex. *Neuron* 27, 45–56 (2000)
5. Lin, Y.W., Yang, H.W., Wang, H.J., Gong, C.L., Chiu, T.H., Min, M.Y.: Spike-timing-dependent plasticity at resting and conditioned lateral perforant path synapses on granule cells in the dentate gyrus: different roles of Nmethyl-D-aspartate and group I metabotropic glutamate receptors. *Eur. J. Neurosci.* 23, 2362–2374 (2006)
6. Froemke, R.C., Poo, M.M., Dan, Y.: Spike-timing-dependent synaptic plasticity depends on dendritic location. *Nature* 434, 221–225 (2005)
7. Tsukada, M., Aihara, T., Kobayashi, Y., Shimazaki, H.: Spatial analysis of spike-timing-dependent LTP and LTD in the CA1 area of hippocampal slices using optical imaging. *Hippocampus* 15, 104–109 (2005)
8. Nishiyama, M., Hong, K., Mikoshiba, K., Poo, M.M., Kato, K.: Calcium stores regulate the polarity and input specificity of synaptic modification. *Nature* 408, 584–588 (2000)
9. Aradi, I., Holmes, W.R.: Role of multiple calcium and calcium-dependent conductances in regulation of hippocampal dentate granule cell excitability. *Neurosci.* 6, 215–235 (1999)
10. Wang, X., Lambert, N.A.: Membrane properties of identified lateral and medial perforant path way projection neurons. *Neurosci.* 117, 485–492 (2003)
11. Spruston, N., Johnston, D.: Perforated patch-clamp analysis of the passive membrane properties of three classes of hippocampal neurons. *J. Neurophysiol.* 67, 508–529 (1992)
12. Staley, K.J., Otis, T.S., Mody, I.: Membrane properties of dentate gyrus granule cells: Comparison of sharp microelectrode and whole-cell recordings. *J. Neurophysiol.* 67, 1346–1358 (1992)

An Analysis of the Autonomic Cardiac Activity by Reducing the Interplay between Sympathetic and Parasympathetic Information

Fausto Lucena¹, D.S. Brito², Allan Kardec Barros², and Noboru Ohnishi¹

¹ Nagoya University, Graduate School of Information Science, Aichi, Nagoya 464-8603, Japan

² UFMA, Laboratory for Biological Information Processing – PIB,
São Luís, MA, Campus do Bacanga S/N 65075-460, Brazil

Abstract. Herein, we make a theoretical effort to characterize the interplay of the main stimuli underlying the cardiac control. Based on the analysis of heartbeat intervals and using neural coding strategies, we investigate the hypothesis that information theoretic principles could be used to give insights to the strategy evolved to control the heart. This encodes the sympathetic and parasympathetic stimuli. As a result of analysis, we illustrate and emphasize the basic sources that might be attributed to control the heart rate based on the interplay of the autonomic tones.

1 Introduction

The mammalian heart works towards maintaining the homeostasis by filling the body with blood. In this scheme, the heart muscles receive an electrical impulse originated in a region known as sinoatrial node that is rich in pacemaker cells. This function represents an evolved accomplishment of nature that is able to adapt the heart rate to fast internal and external changes. In order to fulfill this task, the heart is richly enervated by the autonomous nervous system (ANS) that grants a regulated modulation rhythm supervised by the sympathetic (SNS) and parasympathetic nervous system (SNS). These are able to increase or decrease the heart activity. The question of whether or not it is possible to understand the ANS variations given the sinoatrial node behavior as a mechanism controlling the heart rate, has been at the center of long discussions [9][13].

Several approaches have already tried to explain the autonomic cardiac control using open or closed loop models [1][2][3][4]. However, since there is no accurate method to measure the individual influence of the sympathetic and the parasympathetic tones, it becomes difficult to quantify how accurate those models are. The common idea is to predict the cardiac patterns by describing the erratic behavior of cardiac signals [5]. Unfortunately, efficient models accounting for the cardiac system are intractable, due to the number of variables to be considered. Furthermore, only a few works have shown indication of the basic temporal structure that stresses the cardiac regulatory mechanisms. Therefore we try to circumvent this task by a theoretical approach. We characterized the strategy of the cardiac control by estimating a limited number of features that might act as the main representative forms, i.e. the basic structure of control for the cardiac heartbeat intervals. This alternative method might be accomplished by analyzing the

relationship the SNS and the PNS activity on behalf of the ANS in function of the SA node discharges. The analysis is carried out through understanding how the SNS and the PNS are statistically connected. Thus, the interplay among them would be used as the primary assumption to give insights about the strategy used by the cardiac control to maintain the homeostasis, i.e., how the SA node codifies or “sees” the ANS regulation.

2 Autonomic Cardiac Control and Heartbeat Intervals

The representation of the cardiac cycle in the electrocardiogram (ECG) illustrates the overlapping of waveforms (P, Q, R, S, and T). These are the electrical representations of the heart activity starting from the SA node region composed of pacemaker cells. It is acceptable to represent this spike discharge based on the P-wave of the ECG as representing the beginning of the cardiac cycle. Although, it is computationally complex to determine the P-wave time position. An alternative solution is to mimic such representation with the R-wave, which is the most noticeable wave in the ECG. Thus, the heartbeat intervals obtained from the temporal differences among consecutive maximum peak of R-waves are a representation of autonomic cardiac control (Fig. 1(a)). Figure 1(c) illustrates the spectral analysis for short-time heartbeat intervals (Fig. 1(b)) and shows two main frequency peaks. Those waveforms are assumed to be connected with the autonomic cardiac control discharges, i.e., SNS and PNS.

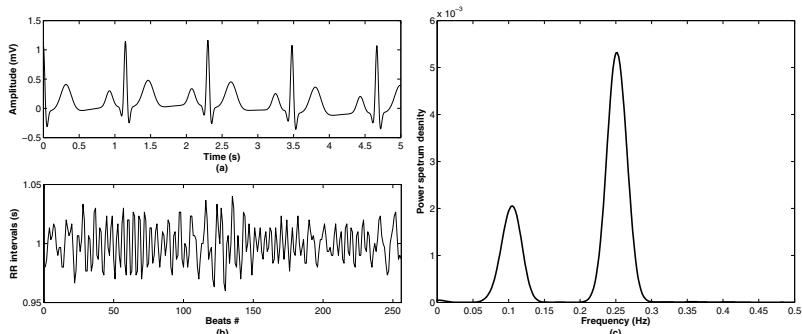


Fig. 1. Analysis of continuous synthetic electrocardiogram(~ 4 min) with an average of 60 ± 1 bpm and noiselessness. (a) Five seconds of the ECG. (b) heartbeat intervals. (c) Power spectrum density of the heartbeat intervals obtained from Welch's method.

Based on clinical experiments, it was suggested that delimited frequency bands would provide a standard basis to analyze the variations in the spectrum [9]. Thus, low frequency bands between $0.03 - 0.15$ Hz are connected with SNS and coupled with PNS, and high frequency bands between $0.15 - 0.4$ Hz for PNS [12]. The advantage is that each frequency band is related to a physiological process. However, there are no strong statistical evidences of those boundaries being in fact the “golden standard”. The SNS and PNS are working in a fashion of equilibrium, where the former is related to “fight or flight” actions of the body, such as threatening situations causing an increase of

the heart rate. On the other hand, the latter neurons work in relaxing responses slowing down the heart rate. The overload of one might cause a cardiac disease (e.g., cardiac heart failure) or autonomic dysfunction (e.g., stress). Several works stressed that it is possible to use this information to assist clinical trials.

3 Estimating Patterns from the Sinoatrial Node Activity

We assume that n observed heartbeat intervals from a dataset - composed of the all possible activity of the autonomic cardiac control \vec{x} - can be seen as a linear combination of i cardiac patterns modeled by spanning the sinoatrial node activity in $\mathbf{A} = [\vec{a}_1 \dots \vec{a}_n]$, given by,

$$\vec{x} = \sum_i^n \vec{a}_i \vec{s}_i = \mathbf{A} \vec{s}, \quad (1)$$

where \vec{s} are underlying factors deriving from the cardiovascular system.

General Approach. To recover \vec{s} only from the observation of \vec{x} , we need to find an alternative representation where a matrix of filters \mathbf{W} will result in $\mathbf{u} = \mathbf{W} \vec{x}$. Thus, $\mathbf{u} = \mathbf{W} \mathbf{A} \vec{s} = \mathbf{P} \vec{s}' = \mathbf{I} \vec{s}''$, where \mathbf{P} is a permutation matrix. It yields to $\mathbf{A} = \mathbf{W}^{-1}$, where the rows of \mathbf{W} are termed filters. This procedure is solved by inferring the statistical structure of the cardiovascular system in relation to the estimation of \mathbf{W} [3].

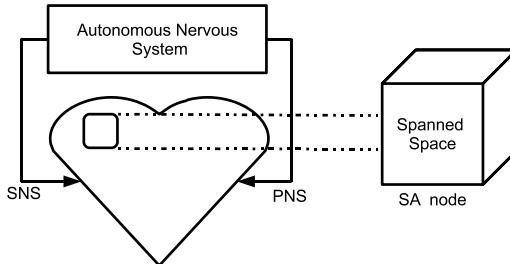


Fig. 2. An illustrative example of the autonomic control of the heart. SNS – Sympathetic nervous system. PNS – Parasympathetic nervous system. SA node – Sinoatrial node.

Inference. If \vec{s} is mainly connected to regulatory mechanisms that increase (SNS) or decrease (PNS) the heart rate. One could argue that both autonomic tones stimulate the heart based on different pathways. Despite this fact, they cause antagonistic behaviors that characterize an interplay [11]. This suggests that they may have an interdependency among their stimuli due to the autonomic cardiac muscle's control that is trying to maintain the homeostasis. Based on these points, it may be sufficient to assume that a decorrelation process (pairwise correlation) would be suitable to study the autonomic stimuli arriving to the heart. Thus, \mathbf{W} were estimated using principal component analysis (PCA) by diagonalizing the matrix $E\{\vec{x}\vec{x}^T\}$.

Database. The mutually orthogonal PCA filters were obtained using two datasets: normal sinus rhythm (NSR) and cardiac heart failure (CHF), as shown in Fig. 3. We used

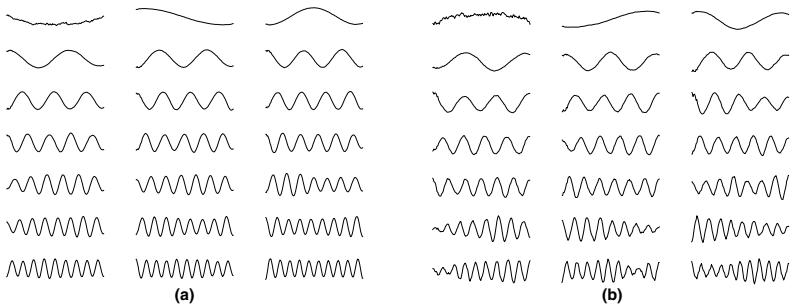


Fig. 3. The first 21 emerged PCA-derived filters. (a) Normal sinus rhythm (NSR). (b) Cardiac heart failure (CHF).

24 hours of heartbeat series (54 NSR and 29 CHF). They were preprocessed with an autoregressive method to eliminate the ectopic or the undetected beats contained in those heartbeat series [7]. Ectopic beats are connected with sporadic contractions of the cardiac muscle and the undetected beats to the lack of precision of the R-wave detector. In both cases, those beats are not connected with the autonomic control of the heart. It means that they are not originated from the sinoatrial node. No supplementary preprocessing was performed over the heartbeat intervals. The database used is freely available [6]. Each heartbeat series was divided into continuous and non-overlapping segments of 128 beats.

4 Analysis of the Emerged Sinoatrial Node Activity

Figure 4 represents the time and frequency plane response for the derived 128 PCA filters. Each overlapped contour plot forming the tilling pattern of the Fig. 4(a-b) corresponds to the energy core for each filter based on the analytical part of the Hilbert transform and Wigner-Ville transform. In other words, let $g(t)$ be a PCA filter. The analytical signal representing the complex function in time is described by $G(g) = g(t) + i\hat{g}(t) = A(t)e^{i\phi(t)}$, where $\hat{g}(t)$ is understood as the Hilbert transform of $g(t)$ and $A(t) = \sqrt{g^2(t) + \hat{g}^2(t)}$ is the instantaneous amplitude. The advantage of applying analytical signal approaches is the guarantee of straightforward positive representations of the frequency signatures.

The PCA base-filters show segregation in two frequency bands. These are ranging from 0.0 to 0.15 Hz and 0.15 to 0.5 Hz for NSR intervals. As expected, filters obtained by PCA are mostly sinusoidal waveforms, where the frequency response is widely localized in time given a tuned frequency localization. For definition, Fourier-like basis represent a signal as a linear superposition of sinusoidal functions and are finned localized in frequency, as shown in Fig. 4. Fourier transforms sacrifices time resolution to find the best frequency localization. By visual acuity, one might notice that the loss of the sinusoidal pattern occurs after 0.15 Hz for NSR as represented in Fig. 4(a) like a tilling area. For CHF (Fig. 4(b)) there is a vague resemblance to the pattern found for NSR after 0.75 Hz. It is not clear whether the PCA basis functions are predicting the

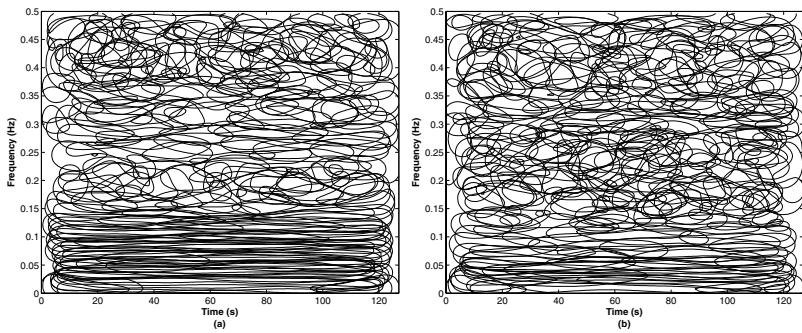


Fig. 4. Time and Frequency plane analysis. (a) Normal sinus rhythm (NSR). (b) Cardiac heart failure (CHF).

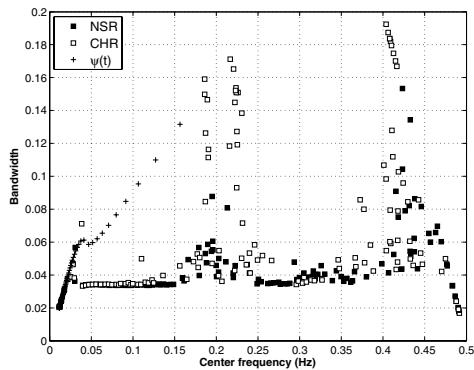


Fig. 5. Comparing the basis population based on filter characteristics

real form of sinoatrial node activity or this phenomena is connected with the stationary character for each window size. We can argue that if the statistical structure of the ensemble used is not being modeled by the emerged PCA basis, why do the NSR and CHF show disparities for the time and frequency population response with identical window size analysis (Fig. 4). Despite the fact that such filters are mostly Fourier-like filters, the division resembles the analysis of short-time heartbeat intervals, these similar patterns are illustrated in NSR and CHF power spectrum analysis. In fact, CHF heartbeat intervals are connected with a suppression of high-frequency energy, when compared to NSR. Figure 5 shows the filter population characteristics, i. e. center frequency and bandwidth of each filter. For instance, Fourier filters are presented like a constant bandwidth along the center frequency.

Filters are used by engineers to retrieve what they want to analyze. The challenge in modeling them is to define parameters to reduce the loss of the object of study, based on the trade-off between time and frequency. Ivanov *et al* [10] heuristically introduced a representation-based filter that uncovered a robust structure in the heartbeat intervals by scaling its behavior. The method filters the heartbeat intervals with the first derivative

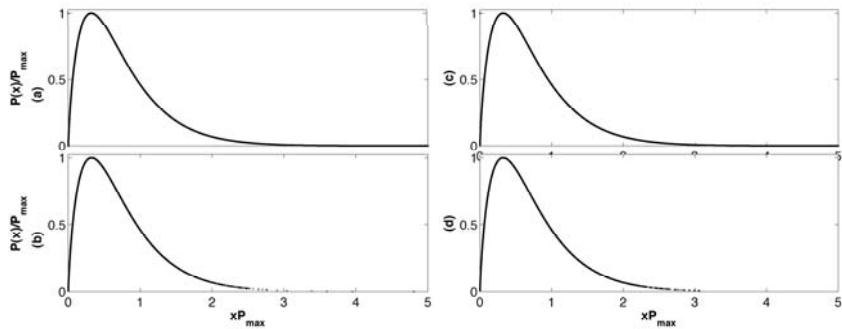


Fig. 6. Probability Gamma distribution. (a) Normal sinus rhythm scaling behavior for frequencies until 0.15 Hz. (b) Normal sinus rhythm scaling behavior for frequencies between 0.15 and 0.5 Hz. (c) Cardiac heart failure scaling behavior for frequencies until 0.15 Hz. (d) Cardiac heart failure scaling behavior for frequencies between 0.15 and 0.5 Hz.

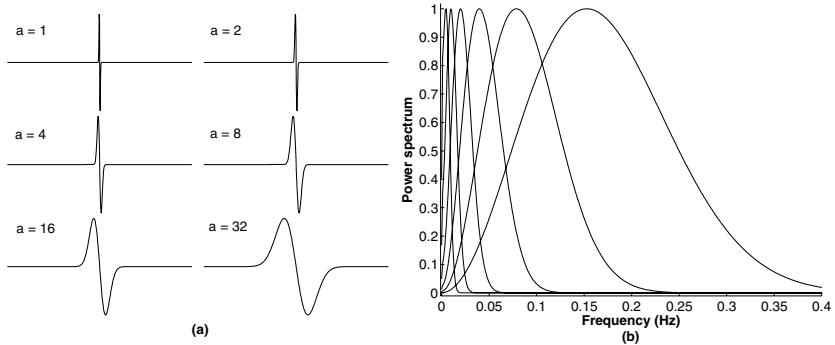


Fig. 7. Time and frequency analysis of the first derivative of the Gaussian function. (a) The time representation of Ivanov's heuristic filter under different scales. (b) Scale analysis in frequency domain for scales varying from 1 to 32, where the bandwidth is becoming narrow for higher scales. It shows that only when $a=32$ can the filter capture some information above 0.2 Hz.

($n = 1$) of a Gaussian function, $\psi(t)^{(n)} = \frac{d^{(n)}}{dt^{(n)}} e^{-0.5t^2/a}$, and scales its behavior with a Gamma distribution of probability [10], where a is the scale. In this case, the functions describing the $\psi(t)$ for different scales are illustrated by a linear increase of bandwidth in function of the center frequency. But, what are the parameters of this filter? Can they have some connections with the filters derived from non-supervised methods? In fact, the first derivative of the Gaussian function, when compared to PCA basis-like filters, presents a partial match for very-low frequency bands illustrated in Fig. 5. Using similar strategy, our results resemble Ivanov's experiments. By filtering heartbeat intervals with PCA basis as filter coefficients can be used to scale the behavior of those intervals, as illustrated in Fig. 6.

The strategy here is based on the similarities between PCA- and Gaussian-derived filters. The former is characterizing a similar scaling behavior. Whereas the filter

applicability of the ones derived from Gaussian function are limited by the scale size, a , which varies from $a = 1$ to $a = 32$. In order to attain basis of comparison, we derived Gaussian filters varying from $a = 1$ until $a = 32$ with the step size of 0.01 Hz, which are depicted in Fig. 7 for the main values of a . One reason for the discontinuity observed in Fig. 5 between 0.15 – 0.25 Hz and 0.4 – 0.5 Hz is explained by the influence caused by the synchrony of the respiratory oscillations. On the other hand, influences above 0.4 Hz are accepted to be vagal nervous control. It is interesting to see that the population code characteristics resemble the energy concentrations on the healthy short-time power intervals, when comparing Fig. 1(c) with Fig. 5.

5 Conclusion

There are several unconventional methods to analyze neural variability, which seem to be applicable to the quasi-oscillatory rhythm of the heart [13]. So far, theoretical information principles have been slowly applied to exploit the underlying structure of the muscle activity during the movement. Still, it is not a common alternative to analyze cardiac muscle variability. In this work, we used an old fashion method to extract information from the heartbeat intervals. The idea of PCA is primarily limited to represent the information contained in the power spectrum (at least for stationary signals), which makes this method an interesting tool to analyze the heartbeat intervals. The reasoning is that the ANS is characterized by the information contained in the power spectrum analysis of heartbeat intervals. The novelty of this paper was to analyze the autonomic interactions that cause the contractions of the cardiac muscle instead of trying to mimic the heartbeat intervals. Since we assumed that the cardiac muscle is efficiently modulated by the activity of several pacemaker cells concentrated in the sinoatrial node, its behavior seems to be carrying a natural strategy to enhance the cardiac information of the heart rate.

Our results suggest the existence of a central command regulating sympathetic and parasympathetic information to control the demand of the body, when the assumption of decorrelation is used. This command may be characterized by Fourier-like basic pattern generators or filters, which would give an explanation of why the spectral response of heartbeat intervals reflects sympathetic and parasympathetic variations.

References

1. 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 Biomed. Eng.* 26, 293–307 (1998)
2. 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)
3. 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)
4. Wiklund, U., Akay, M., Niklasson, U.: Short-term analysis of heart-rate variability by adapted wavelet transforms. *IEEE Eng. Med. and Biol.* 16(5), 113–118 (1997)

5. Pola, S., Macerata, A., Emdin, M., Marchesi, C.: Estimation of the power spectral density in nonstationary cardiovascular time series: Assessing the role of the time-frequency representations (tfr). *IEEE Trans. Biomed. Eng.* 43(1), 46–59 (1996)
6. 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), e215–e220 (2000)
7. Wessel, N., Voss, A.: Renormalised Entropy: A New Method of Non-Linear Dynamics for the Analysis of Heart Rate Variability. *Comp. Card.* 93, 1043–1065 (1994)
8. Bell, A.J., Sejnowski, T.J.: The “Independent Components” of Natural Scenes are Edge Filters. *Vision Res.* 33, 3327–3338 (1997)
9. Akselrod, A., Gordon, A., Ubel, F.A., Shannon, D.C., Barger, A.C., Cohen, R.J.: Power Spectrum Analysis of Heart Rate Fluctuation: A Quantitative Quantitative Probe of Beat-to-Beat Cardiovascular Control. *Science* 213, 220–222 (1981)
10. Ivanov, P.C., Rosemblyum, M.G., Peng, C.-K., Mietus, J., Havlin, S., Eugene, S.H., Goldberger, A.L.: Scaling behaviour of heartbeat intervals obtained by wavelet-based time-series analysis. *Nature* 383, 323–327 (1996)
11. Armour, A.J.: Cardiac neuronal hierarchy in health and disease. *Am. J. Physiol. Regul. Integr. Comp. Physiol.* 271, R262–R271 (2004)
12. Task Force of the ESC and the NASPE. Heart rate variability: standards of measurement, physiological interpretation and clinical use. *Circulation* 93, 1043–1065 (1996)
13. Berntson, G.G., Cacioppo, J.T.: Heart Rate Variability: A Neuroscientific Perspective for Further Studies. *Card Electrophysiol Review* 3, 279–282 (1999)

On Similarity Measures for Spike Trains

Justin Dauwels^{1,2}, François Vialatte¹, Theophane Weber²,
and Andrzej Cichocki¹

¹ RIKEN Brain Science Institute, Saitama, Japan

² Massachusetts Institute of Technology, Cambridge, MA

{jdauwels, theo_w}@mit.edu,

{fvialatte, cia}@brain.riken.jp

Abstract. A variety of (dis)similarity measures for one-dimensional point processes (e.g., spike trains) are investigated, including the Victor-Purpura distance metric, the van Rossum distance metric, the Schreiber *et al.* similarity measure, the Hunter-Milton similarity measure, the event synchronization proposed by Quiroga, and the stochastic event synchrony measures (SES) recently proposed by Dauwels *et al.*

By analyzing surrogate data, it is demonstrated that most measures are not able to distinguish timing precision and event reliability, i.e., they depend on both aspects of synchrony. There are two exceptions: with appropriate choice of parameters, event synchronization quantifies event reliability, independently of timing precision; the two SES parameters quantify both timing precision and event reliability separately. Before one can apply the (dis)similarity measures (with the exception of SES), one needs to determine potential lags between the point processes. On the other hand, SES deals with lags in a natural and direct way, and therefore, the SES similarity measures are robust to lags.

As an illustration, neuronal spike data generated by the Morris-Lecar neuron model is considered.

1 Introduction

In the last years, the problem of detecting correlations between neural signals has attracted quite some attention in the neuroscience community (e.g., [14]). Several studies have related neural synchrony to attention and cognition (e.g., [2]); recently, it has been demonstrated that patterns of neural synchronization flexibly trigger patterns of neural interactions [18]. Moreover, it has frequently been reported that abnormalities in neural synchrony lie at the heart of a variety of brain disorders such as Alzheimer's and Parkinson's disease (e.g., [15]).

In this paper, we focus on the synchrony of *pairs of spike trains*. We consider two different aspects of synchrony: timing precision and reliability. Those concepts can be understood from the following analogy; when you wait for a train in the station, the train may come at the station or it may not come at all, for example, it may be out of service due to some mechanical problem. If the train comes, it may or may not be on time. The former uncertainty is related to reliability, whereas the latter is related to precision.

We will compare and assess a variety of classical and recently proposed spiking synchrony measures, including the Victor-Purpura distance metric [17], the van Rossum distance metric [16], the Schreiber *et al.* similarity measure [12], the Hunter-Milton similarity measure [7], the event synchronization measure proposed in [10], and the stochastic event synchrony measures (SES) proposed by Dauwels *et al.* [3].

We assess those measures by means of surrogate data, which allows us to investigate the statistical properties of those measures; such study, albeit important, does not seem to have been carried out before. Next we apply those measures to quantify the spiking synchrony of Morris-Lecar neurons. (This paper summarizes some of the results, we refer to [4] for a longer manuscript.)

This paper is organized as follows. In the next section, we review the (dis)similarity measures for one-dimensional point processes considered in this study. In Section 3 we investigate the robustness and reliability of those (dis)similarity measures by means of surrogate data. In Section 4 we apply those measures to quantify the firing reliability of Morris-Lecar type I and type II neurons. We offer some concluding remarks in Section 5.

2 Review of Similarity Measures

In this section, we review the (dis)similarity measures considered in this paper. All measures are applied to pairs of point processes x and x' . For the sake of definiteness, we will consider point processes in time, e.g., spike trains.

2.1 Victor-Purpura Spike Train Metric

The distance metric D_V of [17] defines the distance between two point processes as the minimum cost of transforming one point process into the other. This transformation is carried out by combining three basic operations: event insertion, event deletion, and event movement. The cost of deleting or inserting of an event is set to one, whereas the cost of moving an event in time is proportional to the time shift. The inverse of the proportionality constant C_V , denoted by $\tau_V = 1/C_V$, defines the time scale of distance metric D_V . If and only if the point processes x and x' are identical, the distance metric $D_V = 0$.

2.2 Van Rossum Similarity Measure

In the approach of [16], the two point processes are converted into continuous time series. In particular, each event of x is convolved with an exponential function $\exp(t - x_k/\tau_R)$ (with $t > x_k$), resulting in the time series $s(t)$. Likewise each event of x' is convolved with this exponential function, leading to the time series $s'(t)$. From the time series $s(t)$ and $s'(t)$, the van Rossum distance measure [16] is computed as:

$$D_R = \frac{1}{\tau_R} \int_t [s(t) - s'(t)]^2 dt. \quad (1)$$

Note that $D_R(\sigma_S) = 0$ if and only if x and x' are identical. The time scale of this distance measure is determined by the time constant τ_R .

2.3 Schreiber *et al.* Similarity Measure

Also in the approach proposed in [6] and [12], the two point processes x and x' are first convolved with a filter, resulting in time series $s(t)$ and $s'(t)$. The filter may for example be exponential [6] or Gaussian [12], and it has a certain width τ_S . Next the pairwise correlation between the time series $s(t)$ and $s'(t)$ is computed:

$$S_S = \frac{\int_t s(t)s'(t) dt}{\sqrt{\int_t s^2(t) dt} \sqrt{\int_t s'^2(t) dt}}. \quad (2)$$

It is noteworthy that the width τ_S of the filter defines the time scale of interaction between the two point processes. We also wish to point out that if and only if x and x' are identical, we have $S_S = 1$.

2.4 Hunter-Milton Similarity Measure

An alternative similarity measure was proposed in [7]. For each event x_k , one identifies the nearest event $x'_{k'(k)}$ in the point process x' . The degree of coincidence between those two events is determined as $d(x_k) = \exp(-|x_k - x'_{k'(k)}|/\tau_H)$. Along the same lines, one identifies for each $x'_{k'}$ the nearest event $x_{k(k')}$ in the point process x , and determines the degree of coincidence $d(x'_{k'})$. The similarity S_H between x and x' is then computed as:

$$S_H = \frac{\frac{1}{N} \sum_{k=1}^N d(x_k) + \frac{1}{N'} \sum_{k'=1}^{N'} d(x'_{k'})}{2}. \quad (3)$$

The parameter τ_H sets the time scale for event coincidence. If x and x' are identical, we have $S_H = 1$.

2.5 Event Synchronization

Event synchronization [10] defines similarity in terms of coincident events. Two events are considered to be coincident if their timing offset is smaller than a maximum lag τ_Q . This lag can be extracted automatically from the point processes x and x' :

$$\tau_Q(k, k') = \min(x_{k+1} - x_k, x_k - x_{k-1}, x'_{k'+1} - x'_{k'}, x'_{k'} - x'_{k'-1})/2. \quad (4)$$

One computes the number of times an event appears in x shortly after an event appears in x' :

$$d(x|x') = \sum_{k=1}^N \sum_{k'=1}^{N'} J_{kk'}, \quad (5)$$

where

$$J_{kk'} = \begin{cases} 1 & \text{if } 0 < x_k - x'_{k'} \leq \tau_Q \\ 1/2 & \text{if } x_k = x'_{k'} \\ 0 & \text{else.} \end{cases} \quad (6)$$

Similarly one can define $d(x'|x)$, and eventually, event synchronization is determined as:

$$S_Q = \frac{d(x|x') + d(x'|x)}{\sqrt{NN'}}. \quad (7)$$

If and only if all events in x and x' are coincident, we have $S_Q = 1$.

2.6 Stochastic Event Synchrony

Stochastic event synchrony (SES) [3] considers two point processes x and x' as synchronous if they are identical apart from: (i) a time shift δ_t ; (ii) small deviations in the event occurrence times; (iii) a few event insertions and/or deletions.

At the heart of SES lies a statistical model $p(x, x', \delta_t, s_t)$ that describes how the two point processes x and x' may be generated, as illustrated in Fig. 1. One first generates a point process v of length ℓ , next one makes two identical copies of v and shifts those over $-\delta_t/2$ and $\delta_t/2$ respectively; the events of the resulting point process are randomly shifted (with variance $s_t/2$), and some of those events are deleted, independently with probability p_d (indicated by minus sign), resulting eventually in x and x' . As a result of those deletions (or equivalently, insertions), some events in x are non-coincident (marked in red), i.e., they can not be associated to an event in x' and vice versa.

More specifically, SES is defined as the triplet (δ_t, s_t, ρ) , where ρ is the percentage of non-coincident events. We will denote the standard deviation of the (event) timing jitter by σ_t , and hence $s_t = \sigma_t^2$. The SES parameters are computed by conducting statistical inference in the model $p(x, x', \delta_t, s_t)$, more precisely, by coordinate descent and dynamic programming [3].

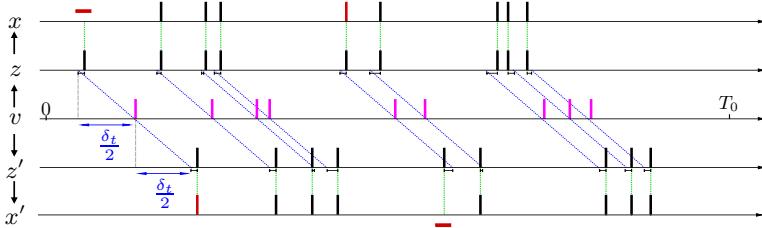


Fig. 1. Stochastic event synchrony: procedure relating x to x'

2.7 Discussion

It is noteworthy that the above mentioned approaches do not discretize the time axis; therefore they avoid the tricky issue of choosing the bin width.

Most of the above measures depend on a parameter that defines the time scale of the interaction between the point processes. Event synchronization, however, adapts its time scale automatically, the user does not need to specify it. The

same holds for SES: the time scale is determined by the parameter s_t , which is computed by the inference algorithm, and does not need to be specified a priori.

There might be a delay between the two point processes x and x' . Before the above mentioned measures (except SES) can be applied, one first needs to estimate potential delays, and shift the point processes accordingly. On the other hand, SES directly handles delays, and it does not require a separate procedure to estimate delays.

3 Analysis of Surrogate Data

In order to benchmark the different measures, we apply them to surrogate data. We randomly generated 10'000 pairs of one-dimensional point processes (x, x') according to the procedure depicted in Fig. 1. We considered several values of the parameters ℓ , p_d , δ_t and s_t (σ_t). More specifically, the length ℓ was chosen as $\ell = \ell_0/(1 - p_d)$, where $\ell_0 \in \mathbb{N}_0$ is a constant. With this choice, the expected length of x and x' is ℓ_0 , independently of p_d . We considered the values $\ell_0 = 100$, $p_d = 0, 0.1, \dots, 0.4$, $\delta_t = 0\text{ms}, 25\text{ms}, 50\text{ms}$, and $\sigma_t = 10\text{ms}, 30\text{ms}$, and 50ms .

The constant C_V of the Victor-Purpura metric was set to 0.001ms^{-1} , and the time constants τ_R , and τ_S , τ_H , and τ_Q were set of 20ms . Those values seemed to yield the most reasonable results. In order to assess the (dis)similarity measures, we compute for each above mentioned parameter setting and for each measure S the expectation $E\{S\}$ and normalized standard deviation $\bar{\sigma}\{S\} = \sigma\{S\}/E\{S\}$. Those statistics are computed by averaging over 10'000 pairs of point processes (x, x') . Some of the results are summarized in Fig. 2. In particular, that figure contains results for SES, the Victor-Purpura metric, and the Schreiber *et al.* measure; the other measures of Section 2 lead to similar results as the Schreiber *et al.* measure. Note that Fig. 2(a) to Fig. 2(d) show curves for $\delta_t = 0\text{ms}, 25\text{ms}, 50\text{ms}$; the curves are practically coincident, except in Fig. 2(b). For the sake of clarity, the other figures only contain curves for $\delta_t = 0\text{ms}$, the curves for different δ_t are not coincident.

From this study of surrogate data, we can conclude the following:

- The measures considered in this paper are reliable in the sense that their statistical fluctuations are relatively small; their normalized standard deviation is typically below 30%, and often even below 20%.
- Most measures depend on both p_d and s_t , and therefore, they are *not* able to separate the two key aspects of synchrony, i.e., timing precision and event reliability. There are two exceptions: the distance metric D_V grows with p_d independently of s_t (cf. Fig. 2(e)). The same holds for the SES parameter ρ (cf. Fig. 2(c)); both D_V and ρ are measures of event reliability. Note that ρ is robust to lags δ_t , in contrast to D_V . The SES parameter s_t is largely independent of p_d (cf. Fig. 2(a)), it is a robust measure for timing dispersion.
- SES is insensitive to lags: the curves in Fig. 2(a) and 2(c) for different values of δ_t are coincident. The other measures strongly depend on δ_t (see Fig. 3(a)), the similarity (dissimilarity) substantially decreases (increases) with δ_t ; therefore, one needs to estimate potential lags before they can be applied.

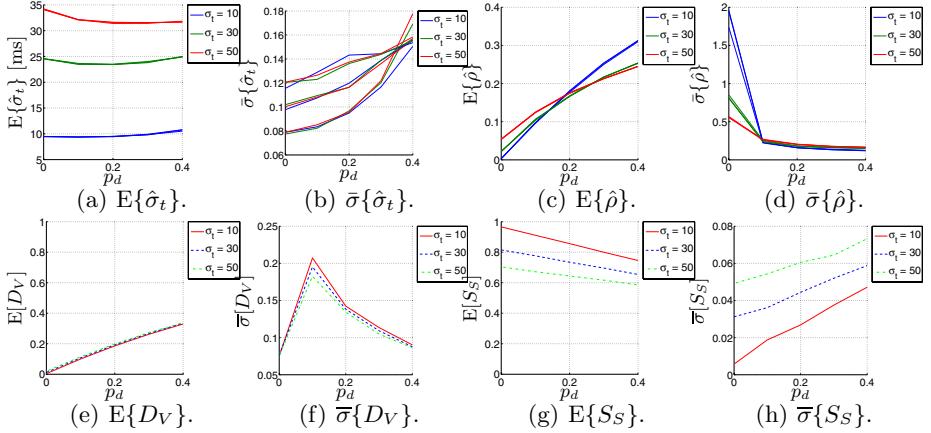


Fig. 2. Results for SES, Victor-Purpura metric D_V and Schreiber *et al.* measure S_S

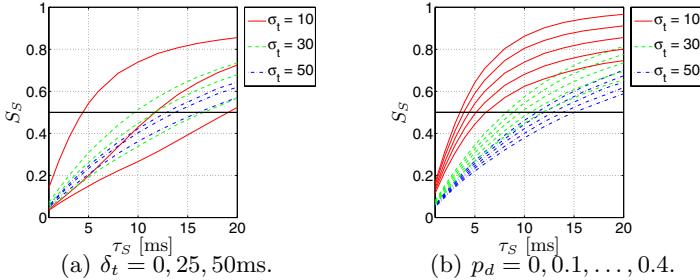


Fig. 3. Sensitivity of the Schreiber *et al.* measure S_S to δ_t and p_d

- There exists a classical procedure to estimate the timing dispersion based on the Schreiber *et al.* measure S_S (see, e.g., [14]). One computes S_S for a range of values of τ_S . The value of τ_S at which $S_S = 0.5$ is considered as an estimate σ_S of the timing dispersion. It is important to realize, however, that since the Schreiber *et al.* measure S_S significantly depends on p_d , also the resulting estimates of timing dispersion will significantly depend on p_d (see Fig. 3(b)). In contrast, the estimate \hat{s}_t of the SES parameter s_t does not suffer from those shortcomings (see Fig. 2(a)).

4 Firing Reliability of a Neuron

In this section, we use the (dis)similarity measures to quantify the firing reliability of neurons. We consider the Morris-Lecar neuron model [9], which exhibits properties of type I and II neurons [5]. The spiking behavior differs in both neuron types, as illustrated in Fig. 4.

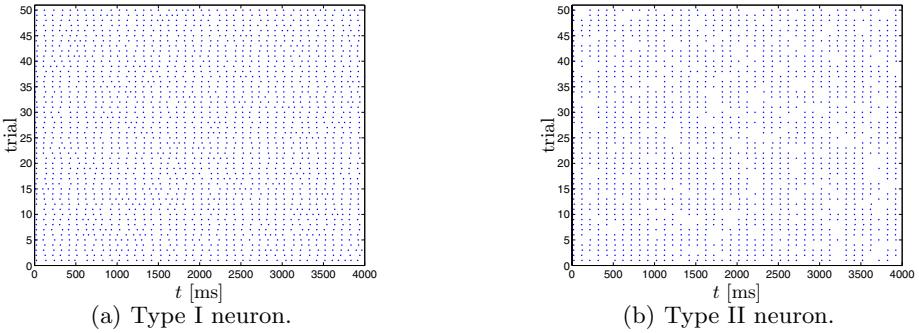


Fig. 4. Raster plots of spike trains from type I (top) and type II (bottom) neurons

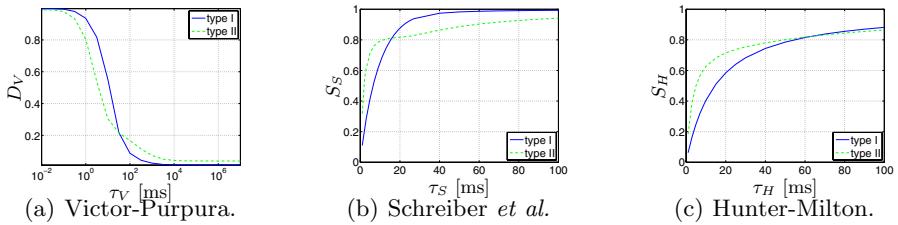


Fig. 5. (Dis)similarity of spike trains of type I/II Morris-Lecar neurons

In our experiments, the input current of the Morris-Lecar neuron consists of a baseline, a sinusoidal component, and zero-mean additive white Gaussian noise. The sinusoidal component forces the neuron to spikes regularly, however, the precise timing varies from trial to trial due to the Gaussian noise (see Fig. 4). Our objective is to investigate how the noise affects the spike timing and the tendency to drop spikes. We are especially interested in how the effect of noise differs in both neuron types. In type II neurons, the timing jitter is small, but spikes tend to drop out. In type I neurons, on the other hand, fewer spikes drop out, but the dispersion of spike times is larger. In other words, type II neurons prefer to stay coherent or to be silent, on the other hand, type I neurons follow the middle course between those two extremes [11].

We computed the similarity measures for each pair of 50 trials. Next we averaged those parameters over all pairs; since there are 50 trials, we have 1225 such pairs in total. A similar approach was followed in [12, 7].

SES results in the estimates $(s_t, \rho) = ((15.2\text{ms})^2, 0.029)$ and $(s_t, \rho) = ((2.7\text{ms})^2, 0.27)$ for type I and type II neurons respectively. This agrees with our intuition: since in type II neurons spikes tend to drop out, ρ should be large. On the other hand, since the timing dispersion of the spikes in type I is large, we expect s_t to be large in those neurons.

The results for the other measures are summarized in Fig. 5; it can be seen that the similarity measures S_S and S_H are larger for type II neurons than for

type I neurons for small time constants τ_S and τ_H , whereas for large time constants, the opposite holds. This can be explained as follows: since the timing dispersion in type I neurons is fairly large, many spikes of type I neurons will be treated as non-coincident (non-overlapping) for small τ_S and τ_H . On the other hand, for large time constants, most spikes of type I neurons will be considered as coincident (overlapping). In contrast, type II neurons have high timing precision, and therefore, the similarity measures S_S and S_H grow quickly with the time constants τ_S and τ_H . However, the measures converge to relatively small values: due to the large number of drop-outs in spike trains of type II neurons, a substantial amount of spikes are treated as non-coincident; therefore, as the time constants grow, the similarity measures S_S and S_H attain smaller values than in type I neurons.

The results of the Victor-Purpura distance metric D_V and the van Rossum distance metric D_R (not shown here) can be understood along the same lines.

As we pointed out earlier, SES adjusts its time scale automatically. The same holds for event synchronization [10]: one may adapt the time constant τ_Q according to [4]. With this adaption rule for τ_Q , we obtained $S_Q = 0.96$ for type I neurons and $S_Q = 0.83$ for type II neurons. This can be understood as follows: since for the data at hand, the adaptive time constant τ_Q is typically about 50ms or larger, the value of S_Q is the lowest in type II neurons due to the frequent drop-outs.

This analysis underlines an important issue: most classical measures depend on a time constant, and in many practical situations, it is not obvious how to choose the “optimal” value of those time constants. Indeed, Fig. 5 suggests that one should compute the measures for a *range* of values of the time constants. As a result, one obtains not just one *single* measure of similarity, but a similarity *function* $S(\tau)$. Such function may not always be easy to interpret, compare, or manipulate in practice. As we pointed out earlier, event synchronization and SES do not suffer from this shortcoming, since they automatically determine the appropriate time scale.

5 Conclusions

We compared various classical and recently proposed measures for spike synchrony. We have shown that most measures are not able to distinguish timing precision and event reliability, with two notable exceptions: with appropriate choice of parameters, event synchronization quantifies event reliability, independently of timing precision; the two SES parameters quantify both timing precision and event reliability separately. Moreover, all measures require the separate procedures to estimate potential lags between the point processes, except SES, which deals with lags in a natural and direct way, and consequently, the SES similarity measures are robust to lags.

References

1. Abeles, M., Bergman, H., Margalit, E., Vaadia, E.: Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. *J. Neurophysiol.* 70(4), 1629–1638 (1993)

2. Buzsáki, G.: Rhythms of the brain. Oxford University Press, Oxford (2006)
3. Dauwels, J., Vialatte, F., Rutkowski, T., Cichocki, A.: Measuring neural synchrony by message passing. In: Advances in Neural Information Processing Systems 20 (NIPS 20) (2007) (in press)
4. Dauwels, J., Vialatte, F., Weber, T., Cichocki, A.: Quantifying statistical interdependence by message passing on graphs. Part I: One-dimensional point processes. *Neural Computation* (2008) (in press)
5. Gutkin, B.S., Ermentrout, G.B.: Dynamics of membrane excitability determine interspike interval variability: a link between spike generation mechanisms and cortical spike train statistics. *Neural Computation* 10, 1047–1065 (1998)
6. Haas, J.S., White, J.A.: Frequency selectivity of layer II stellate cells in the medial entorhinal cortex. *J. Neurophysiology* 88, 2422–2429 (2002)
7. Hunter, J.D., Milton, G.: Amplitude and frequency dependence of spike timing: implications for dynamic regulation. *J. Neurophysiology* 90, 387–394 (2003)
8. Kreuz, T., Haas, J.S., Morellie, A., Abarbanel, H.D.I., Politia, A.: Measuring spike train synchrony. *Journal of Neuroscience Methods* 165(1), 151–161 (2007)
9. Morris, C., Lecar, H.: Voltage oscillations in the barnacle giant muscle fiber. *Bioophys. J.* 35, 193–213 (1981)
10. Quiroga, R.Q., Kreuz, T., Grassberger, P.: Event synchronization: a simple and fast method to measure synchronicity and time delay patterns. *Physical Review E* 66 (2002)
11. Robinson, H.P.C.: Computational Neuroscience: A Comprehensive Approach. In: Feng, J. (ed.) The biophysical basis of firing variability in cortical neurons. Mathematical Biology & Medicine Series. Chapman & Hall/CRC, Boca Raton (2003)
12. Schreiber, S., Fellous, J.M., Whitmer, J.H., Tiesinga, P.H.E., Sejnowski, T.J.: A new correlation-based measure of spike timing reliability. *Neurocomputing* 52, 925–931 (2003)
13. Tiesinga, P., Sejnowski, T.J.: Rapid temporal modulation of synchrony by competition in cortical interneuron networks. *Neural Computation* 16, 251–275 (2004)
14. Tiesinga, P., Fellous, J.-M., Sejnowski, T.J.: Regulation of spike timing in visual cortical circuits. *Nature Reviews Neuroscience* 9, 97–107 (2008)
15. Uhlhaas, P., Singer, W.: Neural synchrony in brain disorders: relevance for cognitive dysfunctions and pathophysiology. *Neuron* 52, 155–168 (2006)
16. van Rossum, M.C.W.: A novel spike distance. *Neural Computation* 13, 751–763 (2001)
17. Victor, J.D., Purpura, K.P.: Metric-space analysis of spike trains: theory, algorithms, and application. *Network: Comput. Neural Systems* 8(17), 127–164 (1997)
18. Womelsdorf, T., Schöffelen, J.M., Oostenveld, R., Singer, W., Desimone, R., Engel, A.K., Fries, P.: Modulation of neuronal interactions through neuronal synchronization. *Science* 316, 1609–1612

Relationship between an Input Sequence and Asymmetric Connections Formed by Theta Phase Precession and STDP

Naoyuki Sato and Yoko Yamaguchi

RIKEN Brain Science Institute, Wako-shi, Saitama 351-0198, Japan

satona@brain.riken.jp

<http://www.dei.brain.riken.jp/~satona/>

Abstract. Neural dynamics of the "theta phase precession" in the hippocampus are known to have a computational advantage with respect to memory encoding. Computational studies have shown that a combination of theta phase precession and spike-timing-dependent plasticity (STDP) can serve as recurrent networks in various methods of memory storage. Conversely, the proposed dynamics of neurons and synapses appear too complicated to give any clear perspective on the network formation in the case of a large number of neurons (>1000). In this paper, we theoretically analyzed the evolution of synaptic weights under a given input sequence. We present our results as a simple equation demonstrating that the magnitude of the slow component of an input sequence giving successive coactivation results in asymmetric connection weights. Further comparison with computer experiments confirms the predictability of network formation.

1 Introduction

It is widely known that the hippocampus serves to maintain environmental memory. The rodent hippocampus is believed to accomplish this by making a cognitive map of the environment [1], while the human hippocampus is believed to be further associated with the episodic memory that is characterized by the individual history of daily life. The episodic memory is experimentally modeled by object-place associative memory [2] which is more complex than simple spatial memories. Recently the rodent hippocampus has also found to demonstrate object-place memory [3], thus the hippocampus is generally considered to have the ability to store complex information.

The local field potential (LFP) theta in the rat hippocampus has been known to appear during running in an environment [4]. O'Keefe and Recce [5] found an interesting relationship between LFP and place cell firings where the phase of the firing to LFP theta gradually increases as the rat passes through the place field. Multi-unit recording findings further demonstrate that the individual place cells show different phase precession patterns and the sequence of place cell activation is represented in spike sequences in each LFP theta cycle. As a

result, the phase precession is suggested to represent "temporal compression" of the input sequence [6]. Since the time scale of the phase difference of two place cell firings in neighboring place fields agrees within a time window of spike time dependent plasticity (STDP) [7], the phase precession pattern is expected to contribute to synaptic plasticity in the hippocampus [6].

The computational ability of theta phase precession in the memory encoding has been evaluated in terms of the sequence memory [8] [9] [10] [11], the spatio-temporal pattern [12], the cognitive map [13], the goal-directed navigation [14] and the object-place memory [15]. These studies demonstrate that theta phase precession could contribute memory formation in various input modalities where the asymmetric connections represent temporal order [8] [9] [10], direction of goal [14] and object-place associations [15]. However, the computational complexity of the system including differential equations is large and, consequently, simulations that adequately describe hippocampal episodic memory are difficult to apply.

In this paper, we theoretically evaluate the associative network formed by theta phase precession as well as the recall of the resultant network. These analyses would contribute to the generalization of the functional role of the asymmetric connection formed by theta phase precession.

2 Mathematical analysis

2.1 Neural Network Model of Theta Phase Precession

Yamaguchi [10] proposed that the input sequence is translated into a phase precession pattern at the entorhinal cortex and the pattern is stored as a CA3 recurrent connection according to STDP. The same dynamics have been applied to temporal order memory [11], spatio-temporal pattern memory [12], cognitive map [13] and object-place memory [15]. The model consists of the input layer, the entorhinal layer and the CA3 layer, where each unit is topologically connected one-by-one. The units in the entorhinal and the CA3 layers are described by phase oscillators. The activation of i -th unit at time t , $p_i(t)$, is given by

$$p_i(t) = \begin{cases} \cos \phi_i(t) & \text{if } \phi_i \bmod 2\pi < \pi \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where ϕ_i is phase state of i -th unit. Dynamics of ϕ_i are given by

$$\dot{\phi}_i = \omega_i - \{C_{\text{exc}}I_i + C_{\text{theta}} \cos(\omega_0 t) - C_0\} \sin \phi_i \quad (2)$$

with

$$\tau_\omega \dot{\omega}_i = \omega_i^{\min} - (\omega_i^{\max} - \omega_i^{\min})I_i - \omega_i \quad (3)$$

where ω_i is native frequency of i -th unit, ω_0 is the theta frequency, I_i is an input of i -th unit representing a binary value (0 or 1) and C_{exc} , C_{theta} , C_0 , τ_ω , ω_i^{\min} and ω_i^{\max} are constants. The phase locking between the unit activation and LFP theta, $\cos(\omega_0 t)$, results in a constant value in relation to the native frequency,

ω_i . During $I_i = 1$, ω_i gradually increases, thus the phase of the unit activation also gradually increases and produces a phase precession pattern as illustrated in Fig.1a.

The phase precession pattern at the entorhinal layer is transmitted to the CA3 layer. The connection weights between the i -th and j -th units, a_{ij} , changes according to the phase precession pattern, $\{\rho_i\}$, as given by,

$$\tau_a a_{ij} = C_w \{ \rho_i(t) \rho_j(t - T_w) - C_r \rho_i(t - T_w) \rho_j \} - (\rho_i(t) + \rho_j(t)) a_{ij} \quad (4)$$

where τ_a , C_w , T_w and C_r are constants. In the following section the network formed by theta phase precession is analyzed.

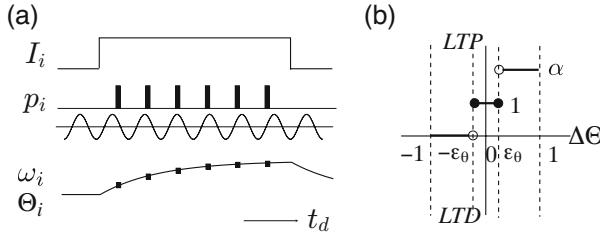


Fig. 1. Memory encoding by theta phase precession. (a) Theta phase precession is generated by phase locking between LFP theta and unit oscillation where the native frequency increases during the receipt of large inputs. (b) A modified Hebb rule with asymmetric time window.

2.2 Memory Encoding

The above neural network model was devised using continuous dynamics, whereas in this section the phase precession is modeled discretely. Here, $t_d = nT_\theta$ is defined by count of theta oscillations (Fig.1a). The phase of the i -th unit activation, $\Theta_i(t_d)$, is proportional to the native frequency, ω_i in eq.(2). ω_i is proportional to a low-pass filtered input sequence, I_i^{LPF} , of which the transfer function is given by $H(z) = \frac{z}{z - e^{-\tau_\omega/T_\theta}}$. Thus, Θ_i is simply given by

$$\Theta_i = I_i^{\text{LPF}} \quad (5)$$

The CA3 connection weight between the i -th and j -th units, $a_{ij}(t_d)$, changes according to STDP. In eq.(4), the resultant increase of connection weight is given by a convolution between a delta function of STDP and half-waves of cosine-shape activation (in eq.(1)). The efficient weight increase could be simplified as in (Fig. 1b),

$$F(\Delta\Theta) = \begin{cases} 0 & \text{if } \Delta\Theta < -\epsilon_\theta \\ 1 & \text{if } -\epsilon_\theta \leq \Delta\Theta \leq \epsilon_\theta \\ \alpha & \text{if } \Delta\Theta > \epsilon_\theta. \end{cases} \quad (6)$$

where ϵ_θ denotes a threshold of the phase difference inducing asymmetric synaptic weight. The evolution of connection weights, a_{ij} , is given by

$$\begin{aligned}\tau_A \dot{a}_{ij} &= I_i I_j F(\Theta_i - \Theta_j) - \beta(I_i + I_j) \\ &= I_i I_j F(I_j^{\text{LFP}} - I_i^{\text{LFP}}) - \beta(I_i + I_j).\end{aligned}\quad (7)$$

where β is a constant ($\ll 1$) associated with τ_a and C_w in eq (4).

The resultant connection weight after one trial encoding (the duration of which is T), A_{ij} , is given by

$$\tau_A A_{ij} = \int_0^T I_i I_j F(I_j^{\text{LFP}} - I_i^{\text{LFP}}) - \beta(I_i + I_j) dt. \quad (8)$$

When $\tau_A \gg 0$, the above equation is transformed into a function of the parameter of the input sequence as follows.

$$\frac{\tau_A}{T} A_{ij} = P_{ij}^S + \alpha P_{ij}^A - \beta(P_i + P_j) \quad (9)$$

with

$$\begin{aligned}P_{ij}^S &= P(I_i = 1, I_j = 1, |I_j^{\text{LFP}} - I_i^{\text{LFP}}| \leq \epsilon_\theta) \\ P_{ij}^A &= P(I_i = 1, I_j = 1, I_j^{\text{LFP}} - I_i^{\text{LFP}} > \epsilon_\theta) \\ P_i &= P(I_i = 1) \\ P_j &= P(I_j = 1)\end{aligned}$$

where the P_{ij}^S denotes a coactivation probability of i -th and j -th units during $|I_j^{\text{LFP}} - I_i^{\text{LFP}}| < \epsilon_\theta$, the P_{ij}^A denotes coactivation probability of i -th and j -th units during $I_j^{\text{LFP}} - I_i^{\text{LFP}} > \epsilon_\theta$, and P_i and P_j are i -th and j -th unit activation probabilities, respectively. The connection weight does not simply reflect the coactivation of the i -th and j -th units, $P(I_i = 1, I_j = 1)$, but rather it further depends on the slow component of the input sequence represented by $I_j^{\text{LFP}} - I_i^{\text{LFP}}$. This is considered to be a major effect of theta phase precession on the synaptic plasticity. If there is no theta phase precession, only symmetric connections are formed by the Hebb rule as follows,

$$\frac{\tau_A}{T} A_{ij} = P_{ij} - \beta(P_i + P_j). \quad (10)$$

3 Computational Experiment

The above equation is tested by using a computational experiment where the original model (in section 2.1) and the simplified equation (in section 2.2) are compared in terms of memory encoding.

3.1 Memory Encoding of Random Sequence in Two Units

The connection weights formed in a pair of two units crucial for the resultant recurrent networks are evaluated. Figure 2a shows the result of the input pattern

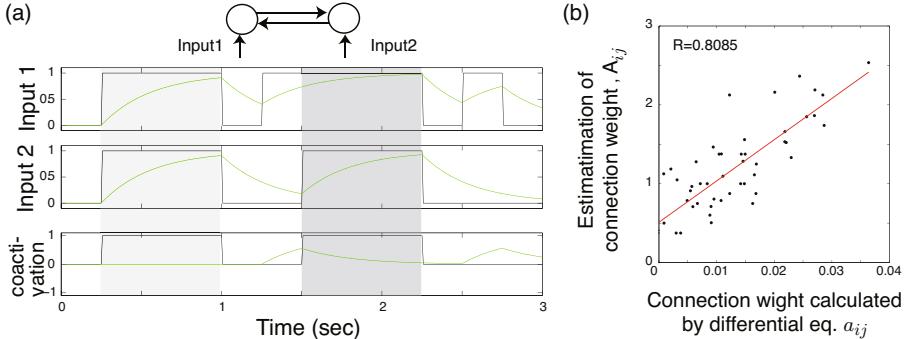


Fig. 2. Memory encoding of random sequence. (a) Input pattern, low-pass input pattern and coactivation of the input and difference of low-passed between units 1 and 2. Periods shown in light gray satisfies the condition, $I_i = 1$ and $I_j = 1$ and $|I_j^{\text{LFP}} - I_i^{\text{LFP}}| < \epsilon_\theta$. Periods shown in light satisfies the condition, $I_i = 1$ and $I_j = 1$ and $|I_i^{\text{LFP}} - I_j^{\text{LFP}}| > \epsilon_\theta$. (b) Correlation of connection weights calculated by the neural network and by eq.(9). $R=0.8085$, $N=50$.

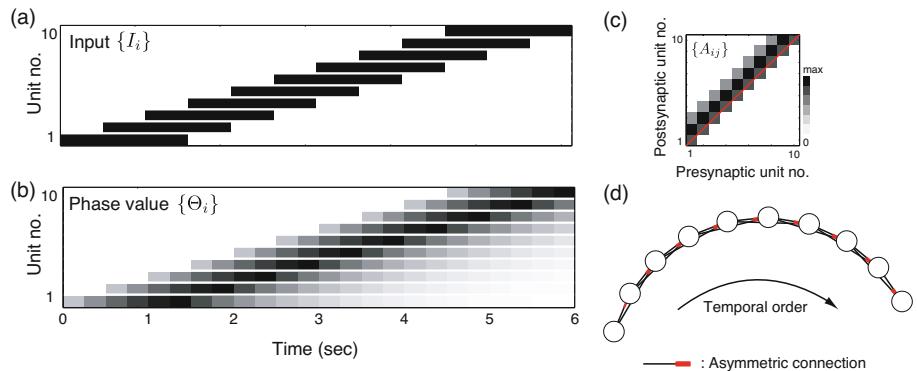


Fig. 3. Memory encoding of a gradually changing input sequence. (a, b) Input sequence and low-passed input sequence evaluating phase value of phase precession. (c) Evaluated connection matrix after single trial learning. The diagonal part indicates the self-connections, with the left and right sides indicating anterograde and retrograde connections, respectively. (d) An equivalent graph of the connection matrix. Each node represents the unit and each link with arrowhead represents the directional connection weight between units. Asymmetric connections are found to be associated with the temporal order of the sequential input.

and low-pass filtered pattern. By using eq.(9), the connection weights are evaluated by these coactivation probabilities. A random input sequence is used where the value of the input changes at $t = nT_\theta/4$, ($n = 1, 2, 3, \dots, 20$), and the average activation probability is 0.5. Using a random input sequence, the connection weights calculated by the neural network are compared with those calculated by the input coactivation probability (Fig. 2b). High correlation between the two

are found ($R = 0.8085, N = 50$). This demonstrates that the connection weights formed by theta phase precession are characterized by the differences of low-pass input patterns sampled by coactivation of these inputs as described in eq.(9).

3.2 Memory Encoding of Temporal Sequence in Multiple Units

In this section, eq. (9) is evaluated by using typical input sequences encoding the environment. Figure 3 shows the results of the evaluated weights of a gradually changing input sequence of the place cell activations. Similarly to the previous report by using explicitly phase precession pattern [10] [11], the asymmetric connection associated with the temporal order of the sequence is successfully obtained. Figure 4 shows the result of the evaluated weights of object–place associations encoded by saccadic visual input sequence. The hierarchical structure of the network is successfully formed by asymmetric connections between units of larger and smaller spatial scales [15]. These results demonstrate that eq.(9) can properly evaluate the connection weights formed by phase precession.

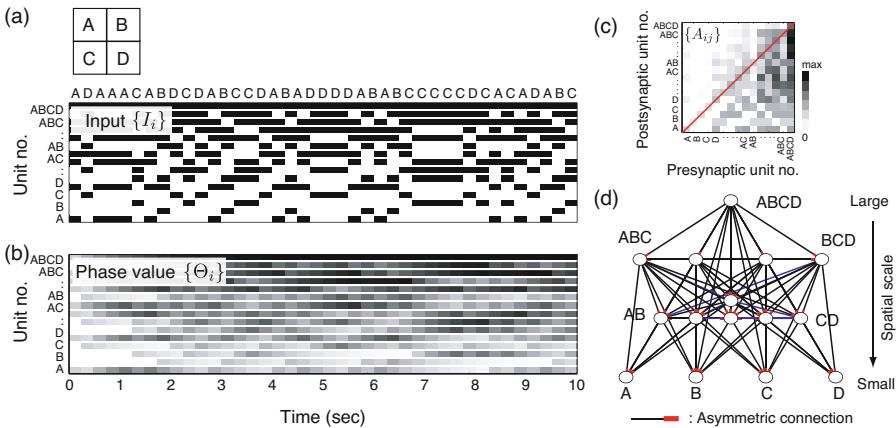
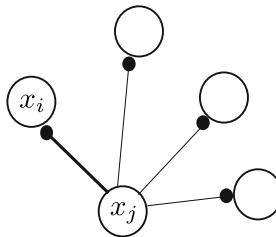


Fig. 4. Memory encoding of object–place associations. (a) Visual environment and visual input sequence representing a random saccadic eye movement, where object input (A, B, C and D) is defined by object feature at the fixation point and spatial units are broadly activated during fixating neighbor objects. (b) Evaluated phase value in phase precession pattern. (c) Evaluated connection matrix. (d) An equivalent graph of the connection matrix. Asymmetric connections are found units between larger and smaller spatial scales. This structure is a hierarchical cognitive map of object–place associations as shown in previous report by using explicit phase precession pattern [15].

4 Discussion

We theoretically evaluated the network formed by theta phase precession and the STDP learning rule. It is known that the phase precession can work as a “temporal compression” of the input sequence [6], while our current result shows



$$P(I_i = 1, I_j = 1, |I_j^{\text{LFP}} - I_i^{\text{LFP}}| \leq \epsilon_\theta) + \alpha P(I_i = 1, I_j = 1, I_j^{\text{LFP}} - I_i^{\text{LFP}} > \epsilon_\theta)$$

Fig. 5. The network formed by theta phase precession and STDP is characterized by a low-pass input sequence sampled by unit coactivation

that a low-passed input sequence sampled by the units' coactivation is encoded by theta phase precession as described by eq.(9) (Fig.4). This slow component of the input sequence has not been emphasized in the previous computational studies of theta phase precession. Nonetheless, these components have been found to be essential for memory encoding in the hippocampus.

Importantly, the result of the current analysis (eq.(9)) does not include any differential equations. Therefore quantitative analysis becomes possible to elucidate the complex memory structures. For example, the memory network of the object-place associations [15] and object relocation [16] have yet to be analyzed. The network of object-place memory has been demonstrated to store the complex information as a hierarchical structure with asymmetric connections. The network is expected to be extended to other memories, such as the semantic network and the language memory. Moreover the current analysis also supports the calculations required by a memory network sufficiently large and complex to model episodic memory in humans.

The Modified Hebb rule for slow input components has been theoretically discussed in the visual [17] and reward-related systems [18]. Theta phase coding outside of the hippocampus is still under discussion [19], while the neural mechanism of theta phase precession can produce a neural encoding of the slow component of the input pattern.

Acknowledgments. This work was supported by Grant-in-Aid for Scientific Research (No. 19700283).

References

1. O'Keefe, J., Nadel, L.: The hippocampus as a cognitive map. Clarendon Press, Oxford (1978)
2. Gaffan, D.: Scene-specific memory for objects: a model of episodic memory impairment in monkeys with fornix transection. *J. Cogn. Neurosci.* 6, 305–320 (1994)
3. Eacott, M.J., Norman, G.: Integrated memory for object, place, and context in rats: a possible model of episodic-like memory? *J. Neurosci.* 24, 1948–1953 (2004)

4. Vanderwolf, C.H.: Hippocampal electrical activity and voluntary movement in the rat. *Electroencephalogr. Clin. Neurophysiol.* 26(4), 407–418 (1969)
5. O’Keefe, J., Recce, M.L.: Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus* 3, 317–330 (1993)
6. Skaggs, W.E., McNaughton, B.L., Wilson, M.A., Barnes, C.A.: Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus* 6, 149–172 (1996)
7. 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)
8. Jensen, O., Lisman, J.E.: Hippocampal CA3 region predicts memory sequences: accounting for the phase precession of place cells. *Learning and Memory* 3, 279–287 (1996)
9. Levy, W.B.: A sequence predicting CA3 is a flexible associator that learns and uses context to solve hippocampal-like tasks. *Hippocampus* 6(6), 579–590 (1996)
10. Yamaguchi, Y.: A theory of hippocampal memory based on theta phase precession. *Biol. Cybern.* 89, 1–9 (2003)
11. Sato, N., Yamaguchi, Y.: Memory encoding by theta phase precession in the hippocampal network. *Neural Comput.* 15(10), 2379–2397 (2003)
12. Wu, Z., Yamaguchi, Y.: Input-dependent learning rule for the memory of spatiotemporal sequences in hippocampal network with theta phase precession. *Biol. Cybern.* 90(2), 113–124 (2004)
13. Wagatsuma, H., Yamaguchi, Y.: Cognitive map formation through sequence encoding by theta phase precession. *Neural Comput.* 16(12), 2665–2697 (2004)
14. Burgess, N., Recce, M., O’Keefe, J.: A model of hippocampal function. *Neural Netw.* 7, 1065–1081 (1994)
15. Sato, N., Yamaguchi, Y.: On-line formation of a hierarchical cognitive map for object-place association by theta phase coding. *Hippocampus* 15, 963–978 (2005)
16. Sato, N., Yamaguchi, Y.: Memory Encoding of Object Relocation in a Hierarchical Associative Network with Theta Phase Coding. In: IEEE International Joint Conference on Neural Networks, Budapest, pp. 637–642 (2004)
17. Földiak, P.: Learning invariance from transformation sequences. *Neural Comput.* 3, 194–200 (1991)
18. Izhikevich, E.M.: Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral Cortex* 17(10), 2443–2452 (2007)
19. Jones, M.W., Wilson, M.A.: Phase precession of medial prefrontal cortical activity relative to the hippocampal theta rhythm. *Hippocampus* 15, 867–873 (2005)

Analysis of Microelectrographic Neuronal Background in Deep Brain Nuclei in Parkinson Disease

Hsiao-Lung Chan¹, Ming-An Lin¹, Tony Wu², Pei-Kuang Chao¹,
Shih-Tseng Lee³, and Peng-Chuan Chen¹

¹ Department of Electrical Engineering, Chang Gung University

² Department of Neurology, Chang Gung Memorial Hospital

³ Department of Neurosurgery, Chang Gung Memorial Hospital

259 Wen-Hua First Road, Gui-Shan, 333, Taoyuan, Taiwan

chanhl@mail.cgu.edu.tw

Abstract. This paper proposes that spectral characteristics of background neuronal potentials can be effective parameters to classifying and identifying neural activities from subthalamic nucleus (STN) and substantia nigra (SNr). The spike-free background signals were obtained from inter-spike microelectrode recording signals. The averaged periodogram was then used to compute the power spectral density of the background signals. Three spectral parameters were extracted and used as discriminant features for artificial neural networks. The commonly used neuronal firing patterns were also extracted from the detected neuronal spikes and used as discriminant features. Our results showed that the classification performance based on background parameters was similar or better than using neuronal firing patterns. This implied that neuronal background can be useful as an aid in targeting STN as well as neuronal firing patterns, saving from spike identification as single- or multi-neuron discharges.

1 Introduction

Subthalamic nucleus (STN) is frequently the target to study and to treat Parkinson's disease [1, 2]. Placing a microelectrode to record neural activities in STN provides information for localization during deep brain stimulation (DBS) neurosurgery. DBS has been approved by Food and Drug Administration (FDA) since 1998[3]. The surgery implants a stimulator to deep brain nuclei, usually STN, to alleviate Parkinson's symptoms, such as tremor and rigidity. For localizing the target, imaging modalities such as X-ray computed tomography and magnetic resonance imaging provide pre-operated, anatomical landmarks [4,5,6]. Neurologists and neurosurgeons can identify whether the electrode reaches the target by hearing, visualizing the recorded signals along electrode trajectories through the intraoperative microelectrode recordings (MER) [7,8]. An important step to determine the STN target is to distinguish the MER signals of STN from its nearby areas, e.g. substantia nigra (SNr) which is a little ventral and medial to STN.

Two kinds of neural information can be retrieved from MER signals. One is the action potentials of neuron cells nearby the microelectrode, producing spike waves.

Neuronal firing patterns, including firing rate (defined as the number of neural spikes within a period) [9] and bursting (defined as clusters of high-frequency spikes released by a neuron [10]) are commonly used to identify deep brain nuclei. Another is the background waves (not including spike waves), aggregation of action potentials of neurons a little distant from the microelectrode. The neuronal firing patterns have some limitations: the computation requires a sufficient spike number to stabilize the measures, thereby needing a longer recording time; the detected spikes may be from a single neuron or multiple neurons so that applying ad-hoc spike classification (an arduous work) helps purifying the quantification. In contrast background wave analysis is less affected by the possible uncertainties caused by nonstationary neuronal discharges and ambiguous spike-classification situations.

In literatures, spectral analysis of inter-spike neuronal background [11] and the spike-removed neuronal signal by replacing spikes with zeros [12] demonstrated increased high-frequency power in the STN. The high-level neuronal background could be attributed to large cells in the STN and hyperactive STN neurons in Parkinson's disease [12]. Nevertheless, the spectral characteristics of neuronal background were less addressed. The replacement of neuronal spikes with zeros would yield a biased spectral estimation of neuronal background due to incomplete signal and the introduced nonstationary components (zero gaps). The biased estimation is in particular serious in the neuronal signal with high spiking rate.

In the present study the spectral characteristics of spike-free signals form STN and SNr in patients with Parkinson's disease were investigated as well as neuronal firing patterns. The power spectral density of the spike-free neuronal signal was then calculated from the inter-spike neuronal background signals. The quantified spectral indexes as well as firing patterns were used to build artificial neural networks for distinguishing MER signals of STN form SNr.

2 Method

The neuronal data used in this study was acquired from 9 patients during DBS implantation surgery in Chang Gung Memorial Hospital. With assistance of imagery localization systems, trials (10s for each) of MERs under spontaneous neuronal discharges were collected in 24,000 Hz. Due to the patients' individual difference, e.g. head size, the depth of STN from the scalp was found varied between 15 and 20 cm. During surgery, the elevation angle of the microelectrode probe from the inter-commissural line was around 50 to 75° and the angle between the probe and the midline was about 8 to 18° toward right or left. 39 trials inside STN and 28 trials inside SNr were collected according to a neurologist expert.

2.1 Spike Detection

Neuronal spikes in each MER were detected by an adaptive threshold that uses proportional feedback to estimate the root-mean-square (RMS) value of near spike-free background noises by keeping the duty cycle (defined as the portion of data with a larger magnitude than the estimated RMS value in a 100-ms running window) at 15.85% [13]. The detection threshold was set to 3 times the estimated background RMS. After a spike was located, searching for larger spikes in following 1 ms was continued to prevent small, connected spikes from being detected as separate. The

power spectral density of the spike-free neuronal signal was then calculated from the inter-spike neuronal background signals.

2.2 Spectral Analysis of Neuronal Background

Owing to the stochastic nature of background signals, there would be variances between the estimated power spectral density and true power spectral density. In order to reduce these variances, the averaged periodogram was employed. Each inter-spike section was divided into contiguous nonoverlapping 1707-ms data segments ($N=4096$ points). The power spectral density of each data segment was computed by the periodogram defined as

$$\hat{S}_{xx}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi fn} \right|^2$$

The spectra of all data segments in each trial were then averaged

$$\hat{S}_{av}(f) = \frac{1}{M} \sum_{i=1}^M \hat{S}_{xx}^{(i)}(f)$$

Fig. 1 shows the averaged periodograms of neuronal backgrounds form one STN and one SNr. The spectra were then characterized by three parameters:

- (1) High-frequency power rate (HFPR) defined as the power over 3000 Hz divided by total spectral power.
- (2) High- to low-frequency power ratio (HLFPR) defined as the power from 3000 to 5000 Hz divided by the power below 2000 Hz.
- (3) Spectral entropy (SE) defined as

$$SE = - \sum_f p(f) \cdot \ln(p(f))$$

where $p(f)$ is the probability density function at frequency f , defined as the power spectral density normalized by total spectral power. If the neuronal background distributes over a narrower frequency band, the estimated SE is lower; vice versa.

2.3 Neuronal Firing Patterns

For comparison, five neuronal firing patterns were computed:

- (1) Firing rate (FR) defined as the total number of detected spikes in a trial divided by trial duration (10 s).
- (2) Burst index (BI) defined as the number of inter-spike-intervals (ISI) < 10 ms divided by the number of ISI > 10 ms [10].
- (3) Pause index (PI) defined as the number of ISI > 50 ms divided by the number of ISI < 50 ms [10].
- (4) Spike strength (SS) defined as the integration of squared spike amplitude for each trial [14].
- (5) Spike number (SN) defined as average spike quantity within bursts [15] where a burst was identified by two criteria: the minimum number of spikes to form a burst is 5; the maximum ISI of the adjacent spikes in a burst is 50 ms.

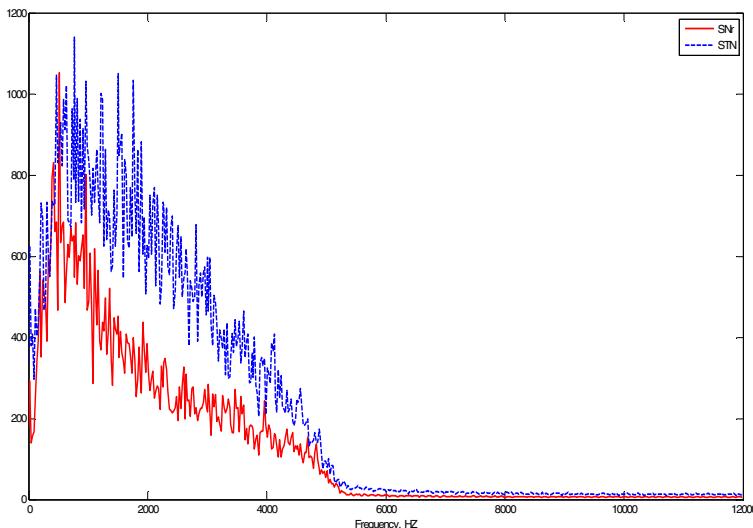


Fig. 1. Power spectra of neuron background of one STN (blue color) and one SNr (red color) trial. The STN signal has a spread distributed spectrum compared to SNr.

2.4 Artificial Neural Networks

Backpropagation neural network with inputs of spike-free background parameters or neuronal firing patterns, 10 or 20 or 40 hidden neurons and an output node for STN and SNr were used to build the classifier for STN and SNr signals. To train and test the neural networks, 39 STN and 28 SNr trials were randomly assigned to training and testing groups. The training group contained 20 trials for STN and 14 trials for SNr and the other trials were used as the testing group. Two dependent variables used for evaluation were sensitivity and specificity. The sensitivity was defined as the number of being correctly identified as STN divided by the number of STN trials in the testing set. The specificity was defined as the number of being correctly identified as SNr divided by the number of SNr trials in the testing set. The above random data partition, network training and testing procedures repeated for 50 times to avoid chance effects.

3 Results

Table 1 shows the sensitivity and specificity of the artificial neural networks in distinguishing STN and SNr signals using background parameters including HFPR, HLFPR, and SE. Table 2 shows the results using neuronal firing patterns including FR, BI, PI, SS, and SN. The classification performance using background parameters was a little better than using firing patterns. Using 10 hidden neurons had better performance than using 20 or 40 hidden neurons.

Table 1. Sensitivity and specificity for distinguishing STN and SNr based on background parameters (HFPR, HLFPR, and SE)

Number of hidden neurons	Sensitivity	Specificity
10	81.6%	68.1%
20	79.6%	72.7%
40	79.3%	70.0%

Table 2. Sensitivity and specificity for distinguishing STN and SNr based on neuronal firing patterns (FR, BI, PI, SS, and SN)

Number of hidden neurons	Sensitivity	Specificity
10	71.7%	65.3%
20	71.1%	66.0%
40	73.1%	66.6%

4 Discussion

MERs from STN and SNr show valuable information which may be useful to assist neuron targeting. Our results showed the neuronal backgrounds led to similar or better classification rate than the commonly used neuronal firing patterns. This implied that neuronal background can be useful as an aid in targeting STN as well as neuronal firing patterns. In future work, trials outside STN and SNr areas should be included in the classifier so that the MERs from sparse neuron areas can be identified.

Acknowledgments. The authors would like to express sincere appreciation to the grant support from the Ministry of Economic Affairs in Taiwan, under contract 95-EC-17-A-19-S1-035 and from National Science Council in Taiwan, under contract NSC 96-2221-E-182-001-MY3.

References

1. Baufreton, J., Zhu, Z.-T., Garret, M., Bioulac, B., Johnson, S.W., Taupignon, A.I.: Dopamine Receptors Set the Pattern of Activity Generated in Subthalamic Neurons. *FASEB J.* 19, 1771–1777 (2005)
2. Magarinos-Ascone, C.M., Figueiras-Mendez, R., Riva-Meana, C., Cordoba-Fernandez, A.: Subthalamic Neuron Activity Related to Tremor and Movement in Parkinson's Disease. *Eur. J. Neurosci.* 12, 2597–2607 (2000)
3. Deuschl, G., Volkmann, J., Krack, P.: Deep Brain Stimulation for Movement Disorders. *Mov. Disord.* 1, S1 (2002)

4. Lee, J.D., Huang, C.H., Lee, S.T.: Improving Stereotactic Surgery Using 3-D Reconstruction. *IEEE Eng. Med. Biol.*, 109–116 (November/December 2002)
5. Miocinovic, S., Zhang, J., Xu, W., Russo, G.S., Vitek, J.L., McIntyre, C.C.: Stereotactic Neurosurgical Planning, Recording, and Visualization for Deep Brain Stimulation in Non-Human Primates. *J. Neurosci. Meth.* 162, 32–41 (2007)
6. Pollo, C., Vingerhoets, F., Pralong, E., Ghika, J., Maeder, P., Meuli, R., Thiran, J.P., Villemure, J.G.: Localization of Electrodes in the Subthalamic Nucleus on Magnetic Resonance Imaging. *J. Neurosurg.* 106, 36–44 (2007)
7. Magnin, M., Jetzer, U., Morel, A., Jeanmonod, D.: Microelectrode Recording and Macrostimulation in Thalamic and Subthalamic MRI Guided Stereotactic Surgery. *Neurophysiol. Clin.* 31, 230–238 (2001)
8. Benazzouz, A., Breit, S., Koudsie, A., Pollak, P., Krack, P., Benabid, A.L.: Intraoperative Microrecordings of the Subthalamic Nucleus in Parkinson's Disease. *Mov. Disord.* 17, S145–S149 (2002)
9. Sterio, D., Zonenshayn, M., Mogilner, A.Y., Rezai, A.R., Kiprovschi, K., Kelly, P.J., Beric, A.: Neurophysiological Refinement of Subthalamic Nucleus Targeting. *Neurosurg.* 50, 58–69 (2002)
10. Favre, J., Taha, J.M., Baumann, T., Burchiel, K.J.: Computer Analysis of the Tonic, Phasic, and Kinesthetic Activity of Pallidal Discharges in Parkinson Patients. *Surg. Neurol.* 51, 665–673 (1999)
11. Merello, M., Tenca, E., Cerquetti, D.: Neuronal Activity of the Zona Incerta in Parkinson's Disease Patients. *Mov. Disord.* 21, 937–943 (2006)
12. Novak, P., Danniluk, S., Ellias, S.A., Nazzaro, J.M.: Detection of the Subthalamic Nucleus in Microelectrographic Recordings in Parkinson Disease Using the High-Frequency (>500 Hz) Neuronal Background. *J. Neurosurg.* 106, 175–179 (2007)
13. Harrison, R.R.: A Low-Power Integrated Circuit for Adaptive Detection of Action Potentials in Noisy Signals. In: Proc. 25th Int. IEEE EMBS Conf., pp. 3325–3328 (2003)
14. Mulloney, B.: A Method to Measure the Strength of Multi-Unit Bursts of Action Potentials. *J. Neurosci. Meth.* 146, 98–105 (2005)
15. Turnbull, L., Dian, E., Gross, G.: The String Method of Burst Identification in Neuronal Spike Trains. *J. Neurosci. Meth.* 145, 23–35 (2005)

Gating Echo State Neural Networks for Time Series Forecasting

Štefan Babinec¹ and Jiří Pospíchal²

¹ Department of Mathematics, Fac. of Chemical and Food Technologies
Slovak University of Technology, 812 37 Bratislava, Slovakia
Tel.:/Fax: +421 2 52495177
stefan.babinec@stuba.sk

² Institute of Applied Informatics, Fac. of Informatics and Information Technologies
Slovak University of Technology, 842 16 Bratislava, Slovakia
Tel.: +421 2 60291548; Fax: +421 2 65420587
pospichal@fiiit.stuba.sk

Abstract. "Echo State" neural networks, which are a special case of recurrent neural networks, are studied from the viewpoint of their learning ability, with a goal to achieve their greater predictive ability. In this paper we study the influence of the memory length on predictive abilities of Echo State neural networks. The conclusion is that Echo State neural networks with fixed memory length can have troubles with adaptation of its intrinsic dynamics to dynamics of the prediction task. Therefore, we have tried to create complex prediction system as a combination of the local expert Echo State neural networks with different memory length and one special gating Echo State neural network. This approach was tested in laser fluctuations prediction. The prediction error achieved by this approach was substantially smaller in comparison with prediction error achieved by standard Echo State neural networks.

1 Introduction

From the point of information transfer during processing, neural networks can be divided into two types: feed-forward neural networks and recurrent neural networks [3]. Unlike the feed forward networks, recurrent neural networks contain at least one cyclical path, where the same input information repeatedly influences the activity of the neurons in a cyclical path. The advantage of such networks is their close correspondence to biological neural networks, but there are many theoretical and practical difficulties connected with their adaptation and implementation. The common problem of all such networks is the lack of an effective supervised training algorithm. This problem was overcome with Echo State neural networks [2]. A very fast algorithm is used in these networks consisting of a calculation of one pseudo-inverse matrix, which is a standard numerical task.

In this paper we studied the influence of the memory length on predictive abilities of these special neural networks. We have found out that Echo State

neural networks (ESN) with fixed memory length can have troubles with adaptation of its intrinsic dynamics to dynamics of the prediction task. Thus we tried to create complex prediction system as a combination of the local expert Echo State neural networks with different memory length and one special gating ESN. The number increase of the Echo State neural networks in the whole prediction system does not mean any substantial increase in computational demands due to very fast training algorithm. The advantage of this approach is that we get higher flexibility and better quality of prediction.

Connection between "liquid state" computing, related to echo states, and backpropagation was mentioned previously in [7][1]. In our previous work [8][9][10] we explored a possibility to improve "one-step" learning by evolutionary approaches and Anti-Oja's learning. Mixtures of local expert neural networks and it's application in time series prediction can be found in [14][5].

2 Echo State Neural Network

Echo State neural networks are atypical in architecture and training within the framework of recurrent neural networks (RNN). This new approach leads to a fast, simple and constructive supervised learning algorithm for the RNN. The basic idea of Echo State neural networks is an application of a huge reservoir, as a source of dynamic behavior of a neural network, from which neural activities are combined into the required output.

2.1 Description of the Neural Network

Neural network consists of (K) input, (N) hidden and (L) output neurons. The state of the neurons in the input layer at time (n) is characterized by the vector $\mathbf{u}_n = (u_1(n), u_2(n), \dots, u_K(n))$, in the output layer by the vector $\mathbf{y}_n = (y_1(n), y_2(n), \dots, y_L(n))$ and in the hidden layer by the vector $\mathbf{x}_n = (x_1(n), x_2(n), \dots, x_N(n))$. The values of all the synaptic weights will be stored in matrices. An input weight matrix shall be created: $\mathbf{W}^{in} = (w_{ij}^{in})$ with dimensions $N \times K$, a weight matrix between hidden neurons: $\mathbf{W} = (w_{ij})$ with dimensions $N \times N$, a matrix of output weights: $\mathbf{W}^{out} = (w_{ij}^{out})$ with dimensions $L \times (K + N + L)$, and a matrix of weights from the output back to the reservoir: $\mathbf{W}^{back} = (w_{ij}^{back})$ with dimensions $N \times L$. It is notable that in this type of network both direct input-output weights, as well as the weights between output neurons are allowed. The largest part of the ESN, the hidden layer, is called dynamic reservoir DR. The structure and the topology of Echo State networks can be adjusted according to their current task. It is not necessary, for example, to use sigmoid output neurons, back weights from the output layer to the reservoir may or may not exist, and even the input neurons may not be used. In this application, sigmoid output neurons were not used, however, back weights from the output layer to the reservoir were used. No loops were used for output neurons. We can find detailed description of the learning algorithm in [2][6].

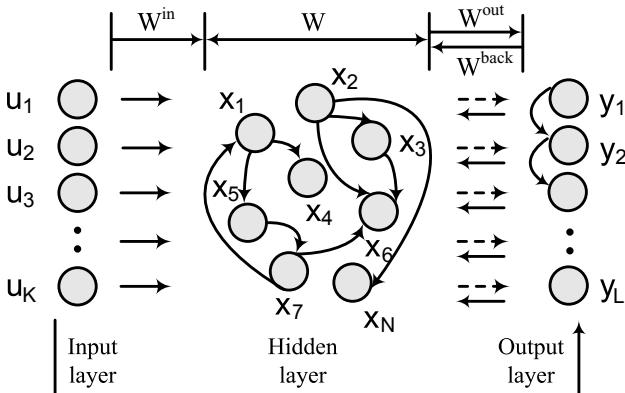


Fig. 1. The typical architecture of Echo State neural networks. The only weights which are trained in this architecture, are the weights from the hidden to the output layer (dashed arrows).

3 Motivation and Method

One of the most important parameters influencing the prediction ability of Echo State neural networks is spectral radius (α) of synaptic weights matrix W . The appropriate choice of the spectral radius has crucial importance for the eventual success of the ESN training. This is because α is closely connected to the intrinsic timescale of the dynamics of the dynamic reservoir. Small value of α means that we have a fast DR, large value of α (close to 1) means that we have a slow DR. The intrinsic timescale of the task should match the DR timescale. Standard settings of α lies in a range between 0.6 and 0.99 (based on empirical observations). Proper setting of this parameter is crucial for the prediction quality of the resulting ESN. It should be small for the fast training data dynamics and large for the slow training data dynamics. Typically, α needs to be hand-tuned by trying out several settings. The DR timescale depends exponentially on $1 - \alpha$, so e.g. settings of $\alpha = 0.99, 0.98, 0.97$ will yield an exponential speedup of DR timescale, not a linear one [2]. From the other point of view α is influencing the memory of the ESN. If the value of α is less than 1, we can use the response of DR in more complex ESN architecture. On the other hand, the neural network has the tendency to gradually forget information from the previous time steps. In other words, ESN has memory with exponential decrease of information.

With the change of α parameter we are also changing the number of the previous inputs, which will have the influence on the current state of the DR. We have made the following simple experiments. Different Echo State neural networks were trained and used for the time series prediction. The only difference among individual ESN networks was in the α parameter value. In the end, every ESN network had different prediction accuracy of the whole testing set. But what is more important, not every sample from the testing set was

predicted most accurately by the winning ESN. In other words, Echo State neural networks with different memory length differed in the prediction accuracy for the individual time steps of the whole testing set. Thus our goal was to create complex prediction system as a combination of the local expert Echo State neural networks with different α parameter and one special gating ESN. Local expert Echo State neural networks were used for the testing set prediction, but the task of the gating ESN was different. This neural network should determine which local expert ESN has the best prediction accuracy for the given time step of the testing set. The number increase of the Echo State neural networks in the whole prediction system does not mean any substantial increase in computational demands, because the whole ESN training process is only computation of one pseudo inverse matrix. We can see the scheme of the whole prediction system in the Figure 2.

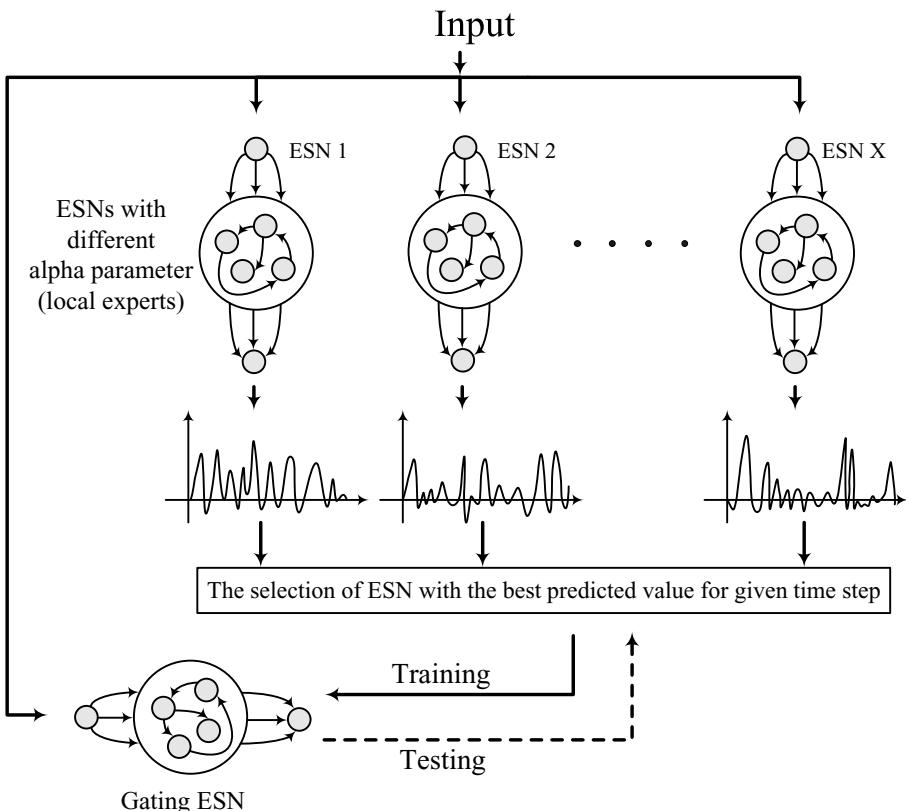


Fig. 2. The scheme of the whole prediction system

4 Experiments

We have used standard benchmarking data set in this paper. The testing set was composed of a time sequence of 1000 samples of laser fluctuations data, and the quality of prediction was measured by an error of prediction in the next 100 steps. A mean absolute percentage error (MAPE) was used to measure the quality of prediction of this testing set, where test values P_i^{real} and predicted values P_i^{calc} are used, and N is the number of points of the testing set (the length of the predicted time series):

$$MAPE = \frac{\sum_{i=1}^N \left| \frac{P_i^{real} - P_i^{calc}}{P_i^{real}} \right|}{N} \times 100. \quad (1)$$

The whole learning process was divided into three phases. The first phase was aimed at finding the best parameters of local expert ESN networks for quality prediction results from the training set. The results for training set are in Table 1. In this table, DR represents the dynamic reservoir; α is spectral radius of synaptic weights matrix W . These DR and α values were chosen in accordance with the proposal used by Jaeger (see [2]). Experiments were carried out in the following way. For each value of DR and the parameter α , the values of synaptic weights in DR were randomly generated 50 times and for each initialization of weights, the error for the testing set was calculated. Further, an average error of all 50 trials is presented in the column *Average MAPE*. Also, the smallest achieved error was recorded in the *Best MAPE* column in the same table. A clear correlation between Best and Average value columns is apparent from Table 1. When a better *Average MAPE* was achieved, there is also a better *Best MAPE*. The best results were achieved with a DR consisting of 250 neurons and for the parameter α equal 0.8.

The second phase of the experiments was aimed at finding the best parameters of the gating Echo State neural network. First, we had to create the training set for this special neural network. Therefore we had trained 40 ESN networks with different α parameter values. The rest of the parameter values and initialization of the synaptic weights of these neural networks were chosen based on

Table 1. Results of representative experiments in the first phase: quality of the laser prediction for different parameter values

Index	DR	α	Average MAPE	The best MAPE
1	200	0.8	33.86 %	31.24 %
2	250	0.8	34.23 %	29.52 %
3	250	0.7	36.29 %	31.34 %
4	300	0.8	35.44 %	30.89 %
5	300	0.7	35.94 %	32.85 %
6	350	0.8	38.22 %	34.64 %

the best results from the first phase of the experiments. The α parameter value of individual ESN ranged between 0.59 and 0.99 with sampling step 0.1. We have used the individual ESN networks after the training process for the self prediction of the training set. Afterwards, we were able to select the ESN networks with the best prediction accuracy for the individual time steps of the training set. The architecture of the gating ESN was composed of one input and one output neuron. The value of the time series (training set) at time n represented the input into the gating ESN and the desired output was sequence number of the ESN network with the best prediction accuracy for the following timestep (time $n + 1$). Thus the gating ESN had the classification task. The best results were achieved with a DR consisting of 300 neurons and for the parameter α equal 0.76.

The main experiments were carried out in the third phase with already trained local expert ESN networks and the gating ESN. All 40 local expert ESN networks were used for the prediction of the next 100 values, which were not a part of the training set (these values were used for the prediction quality measurements). After this process, we have created one time series as an average of all 40 time series predicted by local expert ESN networks. This time series served as an input

Table 2. Comparison of two different approaches

Approach	MAPE
Standard Echo State neural network	29.52 %
Gating of Echo State neural networks	16.62 %

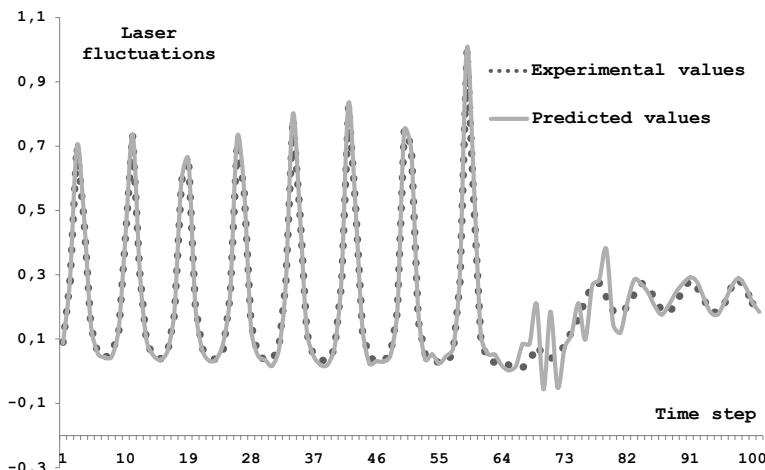


Fig. 3. Testing data: 100 records of laser fluctuations and 100 values predicted by standard Echo State neural network (MAPE 29.52 %)

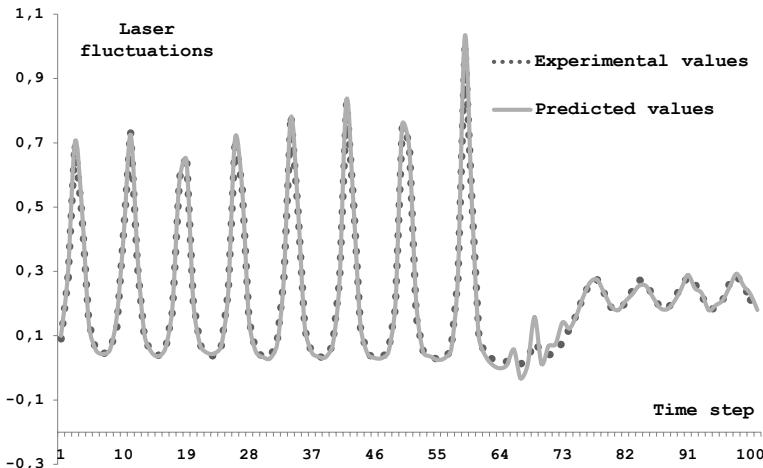


Fig. 4. Testing data: 100 records of laser fluctuations and 100 values predicted by gating of Echo State neural networks (MAPE 16.62 %)

into the gating ESN. Gating ESN was consequently used for the determination of the local expert ESN with the best prediction accuracy for individual time steps of the testing set.

In the following Table 2 we can see the comparison of best achieved errors on testing data sets for standard Echo State neural network and our new approach – gating of Echo State neural networks. We can see graphical representation of these two approaches in Figures 3 and 4. It is clear from this table and figures that the gating of Echo State neural networks can considerably increase the quality of prediction in comparison with classic Echo State neural network.

5 Conclusions

Echo State neural networks are relatively new in the domain of neural networks. Their advantage is a closer connection with biological models inherent to recurrent neural networks and in their usage of the reservoir of dynamic behavior without adjusting the weights within the hidden layer. Echo State neural networks have a substantial advantage over other types of recurrent networks in their "one-step" learning ability.

However, there is no incremental approach further improving performance of an Echo state neural network. We decided to improve the performance by using a set of Echo state neural networks, each trained with different value of alpha parameter. This parameter controls the length of time, during which the old inputs can influence the output of the Echo state neural network, metaphorically "the length of the Echo". Gating Echo state neural network was trained to decide, in which part of the dynamic phase the modeled system is. In each

time step of the test, this gating network selected from the set the best Echo state neural network trained with suitable alpha parameter. The answer of this selected network was used as a response in the current time step.

We have chosen laser fluctuations as a testing data. Our aim was to find out if this approach is able to increase prediction quality in comparison with original Echo State neural networks. From the results shown in the paper, it is clear, that this aim has been accomplished. Gating of Echo State neural networks can increase the quality of the network's prediction.

Acknowledgement. This work was supported by Scientific Grant Agency Vega of Slovak Republic under grants 1/4053/07, 1/0804/08 and 1/0848/08.

References

1. Jacobs, R., Jordan, M., Nowlan, S., Hinton, G.: Adaptive mixtures of local experts. *Neural Computation* 3, 79–87 (1991)
2. Jaeger, H.: The Echo State Approach to Analysing and Training Recurrent Neural Networks. German National Research Center for Information Technology, GMD report 148 (2001)
3. Haykin, S.: Neural networks - A comprehensive foundation. Macmillan Publishing, New York (1994)
4. Jordan, M., Jacobs, R.: Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* 6, 181–214 (1994)
5. Carvalho, A., Tanner, M.: Mixtures-of-experts of autoregressive time series: asymptotic normality and model specification. *IEEE Transactions on Neural Networks* 16(1), 39–56 (2005)
6. Jaeger, H.: Short Term Memory in Echo State Networks. German National Research Center for Information Technology, GMD report 152 (2002)
7. Natschläger, T., Maass, W., Markram, H.: The "liquid computer": A novel strategy for real-time computing on time series. Special Issue on Foundations of Information Processing of *TELEMATIK* 8(1), 39–43 (2002)
8. Babinec, S., Pospíchal, J.: Optimization in Echo state neural networks by Metropolis algorithm. In: Matousek, R., Osmera, P. (eds.) *Proceedings of the 10th International Conference on Soft Computing, Mendel 2004*, pp. 155–160. VUT Brno Publishing (2004)
9. Babinec, S., Pospíchal, J.: Two approaches to optimize echo state neural networks. In: Matousek, R., Osmera, P. (eds.) *Proceedings of the 11th International Conference on Soft Computing, Mendel 2005*, pp. 39–44. VUT Brno Publishing (2005)
10. Babinec, S., Pospíchal, J.: Improving the Prediction Accuracy of Echo State Neural Networks by Anti-Oja's Learning. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) *ICANN 2007. LNCS*, vol. 4668, pp. 19–28. Springer, Heidelberg (2007)
11. Goldenholz, D.: Liquid computing: A real effect. Technical report, Boston University Department of Biomedical Engineering (2002)

A Novel Artificial Model of Spiral Ganglion Cell and Its Spike-Based Encoding Function

Hiroyuki Torikai and Toru Nishigami

Graduate School of Engineering Science, Osaka University, Japan
torikai@sys.es.osaka-u.ac.jp, tohru@hopf.sys.es.osaka-u.ac.jp

Abstract. In the mammalian inner ear, the inner hair cell transforms a sound-induced mechanical vibration into an electric potential. The spiral ganglion cell encodes the electric potential into a spike-train which is transmitted to the central nervous system. In this paper we present a novel artificial electrical circuit model of the spiral ganglion cell. We derive a return map which can analytically describe dynamics of the model. Using the map, we can derive theorems that guarantee that the presented model can realize some of important properties of the biological spiral ganglion cell. The theorems are confirmed numerically.

1 Introduction

In the mammalian inner ear, the inner hair cell transforms a sound-induced mechanical vibration of the basilar membrane into a receptor potential. A set of *spiral ganglion* (SG) cells encodes the receptor potential into a set of spike-trains which is transmitted to the central nervous system. Physiological experiments have revealed various properties of the SG cell, e.g., properties about responses to the receptor potential and about spontaneous activities [1]. In this paper we focus on the following important properties about the encoding function: (P1) density of the spike timings mimics waveform of the receptor potential, and (P2) the spike-trains should not synchronize to each other in order to realize a high encoding resolution. In this paper, first, we present a novel electrical circuit model of the SG cell. We then show that the SG cell model can satisfy the properties P1 and P2 by a numerical simulation. Second, we derive a return map (called phase map) which can analytically describe dynamics of the SG cell model without any approximation. Using the phase map, we can derive two theorems that can guarantee that the SG cell model can satisfy P1 and P2. In addition, the theorems can guarantee robustness of the encoding function against a parameter mismatch. Significances and novelties of this paper are many, including the following points. (a) Analysis of encoding functions of a spiking neuron model is an important research topic from both fundamental and application viewpoints. For example, several simple spiking neuron models have been synthesized and their encoding functions have been clarified as summarized in Table 1 [2]-[8]. This paper firstly presents the integrate-and-fire type neuron model (i.e., the SG cell model) that can satisfy the encoding function based on P1 and P2. (b) The presented SG cell model will be a prototype to synthesize a modified model which can mimic not only P1 and P2 but also other properties of the biological SG cell. Such a biologically plausible SG cell model will be applicable for a biologically plausible cochlea implant [9]. (c) Such a biologically plausible cochlea implant should include a huge number of artificial SG cells like the mammalian inner ear. The presented SG cell model is suitable for this purpose since it has a simple circuit structure.

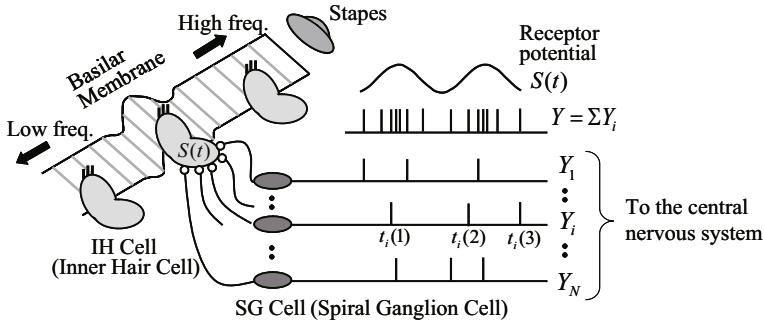


Fig. 1. Some of basic mechanisms of the mammalian inner ear. This figure is a modified version of a figure in the representative text book on the ear mechanisms [1].

2 Basic Inner Ear Mechanism and Problem Setting

In this section we review some of basic mechanisms of the mammalian inner ear [1] and explain purpose of this paper. Fig.1 shows a schematic sketch of the inner ear. A sound input induces mechanical vibrations of the stapes and the basilar membrane. The basilar membrane works like a Fourier transformer, i.e., a higher frequency component of the sound input induces vibration of a membrane's partition that is nearer to the stapes side. Let a partition of the membrane be vibrated with an angle frequency ω . The vibration is input to an *inner hair* (IH) cell which is attached to the membrane. The IH cell transforms the mechanical vibration into an electrical potential $S(t)$ which is referred to as a *receptor potential*. The receptor potential $S(t)$ can be approximated by $S(t) = P_0 + K \cos(\omega t)$ for a certain range of the angle frequency ω of the mechanical vibration, where P_0 and K are determined by amplitude of the vibration. One IH cell is connected by a set of N *spiral ganglion* (SG) cells, where $N \simeq 20$ in the case of humankind. The receptor potential $S(t)$ is commonly input to each SG cell which outputs a spike-train $Y_i(t)$, where $i \in \{1, 2, \dots, N\}$ is a cell index. It is suggested that the receptor potential $S(t)$ is encoded into the sum $Y(t)$ of the spike-trains:

$$Y(t) = \sum_{i=1}^N Y_i(t).$$

In order to investigate such an encoding function, various physiological experiments have been done and several properties of the SG cell have been revealed [1]. Among such properties, in this paper we focus on the following important ones.

- P1.** As shown in Fig.1 let $t_i(n)$ be the n -th spike position of $Y_i(t)$. Also let $\Theta_i(n) = t_i(n) \pmod T$ be the corresponding spike phase, where $T = 2\pi/\omega$ is period of the input $S(t)$ to the SG cell. Physiological experiments show that density (histogram) of the spike phase $\Theta_i(n)$ mimics waveform of the input $S(t)$ for $0 \leq t < T$ [1].
- P2.** The spike-trains $\{Y_1, \dots, Y_N\}$ should not synchronized to each other so that the set of N SG cells can have a high resolution of the encoding function. If all the spike-trains are synchronized, the set of N SG cells can only have the same encoding resolution as the single SG cell.

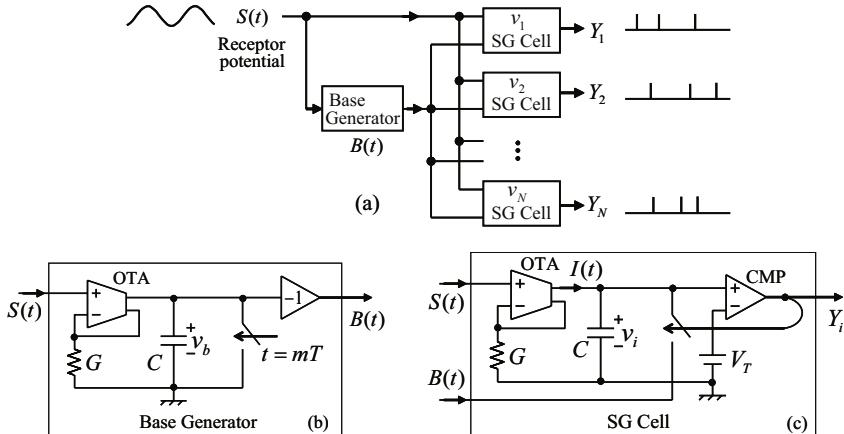


Fig. 2. (a) Electrical circuit model of *spiral ganglion* (SG) cells. (b) Base generator. OTA represents operational transconductance amplifier. The switch is closed instantaneously at $t = mT$, $m = 0, 1, 2, \dots$. (c) SG cell. CMP represents comparator.

In the following sections we present a novel electrical circuit model of the SG cell and show that it can satisfy the properties P1 and P2 both numerically and theoretically.

3 Electrical Circuit Model of the SG Cell and Its Encoding Function

We present a novel electrical circuit model of the set of SG cells in Fig.2. The model is just called *SG cell* when there is no confusion with the biological one. A receptor potential $S(t) = S_0 + K \cos(\omega t)$ is commonly input to the N SG cells as shown in Fig.2(a), where $S_0 > 0$ and $K > 0$. The SG cell in Fig.2(c) has the operational transconductance amplifier (OTA) which generates the current $I(t) = GS(t)$ flowing into the capacitor C . The capacitor voltage v_i corresponds to a membrane potential, and the SG cell has the common firing threshold V_T . If $v_i < V_T$, the comparator (CMP) outputs $Y_i = -E$, where E is the power-supply voltage. If $v_i = V_T$, the comparator generates a firing spike $Y_i = E$. The spike $Y_i = E$ closes the switch instantaneously and then the voltage v_i is reset to $B(t)$ which is referred to as a *reset base*. Each SG cell has the common reset base $B(t)$ which is generated by the base generator in Fig.2(b). Using the following dimensionless variables and parameters

$$\begin{aligned} \tau &= \frac{1}{T}t, \quad x_i = \frac{1}{V_0}v_i, \quad s_0 = \frac{T}{RCV_0}S_0, \quad k = \frac{T}{RCV_0}K, \quad a = \frac{1}{V_0}V_T, \\ s(\tau) &= \frac{T}{RCV_0}S(T\tau), \quad b(\tau) = \frac{T}{RCV_0}B(T\tau), \quad y_i(\tau) = \frac{1}{2E}(Y_i + E), \end{aligned}$$

dynamics of the SG cell can be described by the following dimensionless Equations (1), (2), (3) and (4), where V_0 is some positive voltage for scaling. Each SG cell accepts the common input

$$s(\tau) = s_0 + k \cos(2\pi\tau) \quad (1)$$

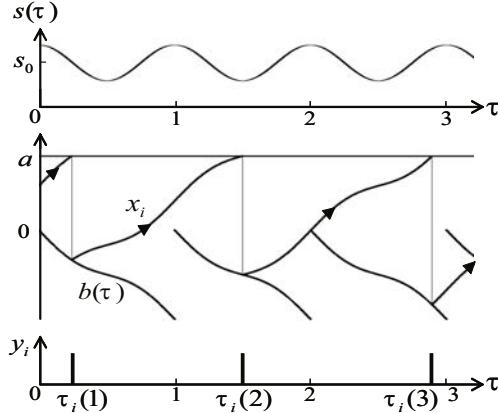


Fig. 3. Basic dynamics of the SG cell

where $s_0 > k \geq 0$ is assumed for simplicity. An example of $s(\tau)$ is shown in Fig.3. The dynamics of the membrane potential x_i is described by

$$\begin{cases} \dot{x}_i = s(\tau) & \text{for } x_i < a, \\ x_i(\tau^+) = b(\tau) & \text{if } x_i(\tau) = a, \end{cases} \quad (2)$$

where $\dot{x} = dx/d\tau$, $\tau^+ = \lim_{\epsilon \rightarrow 0} \tau + \epsilon$ and $\epsilon > 0$. A typical waveform of the membrane potential x_i is shown in Fig.3. When the potential x_i reaches the firing threshold a , the potential x_i is reset to the reset base $b(\tau)$ which is described by

$$b(\tau) = -\sigma(\tau(\bmod 1)), \quad \sigma(\tau) = \int^\tau s(\xi)d\xi = s_0\tau + \frac{k}{2\pi} \sin(2\pi\tau). \quad (3)$$

A reset base $b(\tau)$ is shown in Fig.3. Repeating the resettings of the membrane potential x_i , the SG cell outputs a spike-train $y_i(\tau)$ described by

$$y_i(\tau^+) = \begin{cases} 0 & \text{for } x_i < a, \\ 1 & \text{if } x_i(\tau) = a. \end{cases} \quad (4)$$

Remark 1 (Reset base and encoding function): The SG cell described by Equations (2) and (4) can realize various spike-based encoding functions by adjusting shape of the reset base $b(\tau)$ as summarized in Table 1. In this paper we propose the reset base $b(\tau)$ in Equation (3) and its generator in Fig.2(b) in order for the SG cell to satisfy the properties P1 and P2. We emphasize that only one base generator is needed for the set of N SG cells to satisfy the properties P1 and P2.

Now let us confirm whether the SG cell satisfies the properties P1 and P2 by using a numerical simulation. (Theoretical results are given in the next section.) Fig.4 shows a simulation result for $N = 3$. Fig.4(a) shows the input $s(t)$ and the spike-trains (y_1, y_2, y_3) . Let $\tau_i(n)$ be the n -th spike position of $y_i(\tau)$, and let

$$\theta_i(n) = \tau_i(n) \pmod{1}$$

Table 1. Shapes of the reset base $b(t)$ and typical phenomena

Shape of reset base $b(\tau)$	Typical phenomena of the SG cell described by Eqs. (2) and (4)	Typical functions and applications
Constant	Quasi-periodic spike-train $y_i(\tau)$	Rate coding [2], Temporal coding [3]
Periodic sawtooth wave	Chaotic spike-train $y_i(\tau)$, Various unstable periodic spike-trains	Temporal coding [4][5]
Sinusoidal wave	Resonance of ISI [6], Period-doubling bifurcation [8]	Input parameter identification [6][7]
Equation (3) in this paper	Chaotic spike-train $y_i(\tau)$	Encoding of sound signal based on P1 and P2

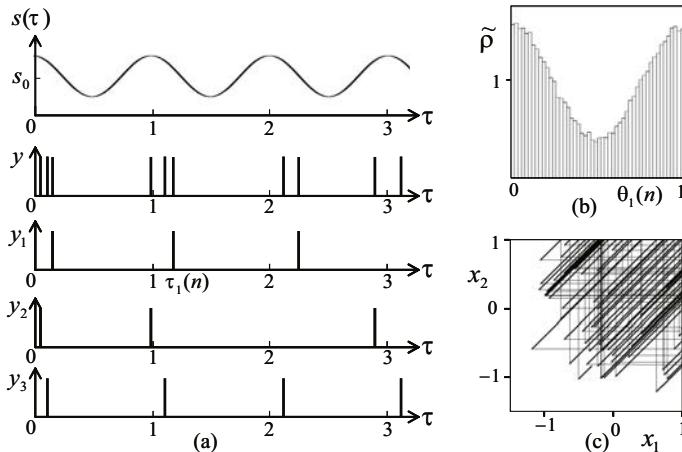


Fig. 4. (a) Spike-trains of the SG cells. $N = 3$ and $(s_0, k, a) = (1.2, 0.7, 1)$. (b) Histogram $\tilde{\rho}$ of the spike phase $\theta_1(n) = \tau_1(n) \pmod{1}$. Similar histograms are obtained for $\theta_2(n)$ and $\theta_3(n)$. Hence the property P1 is satisfied. (c) Lissajous figure on the (x_1, x_2) -plane. Similar figures are obtained for the (x_1, x_3) - and the (x_2, x_3) -planes. Hence the property P2 is satisfied.

be the corresponding spike phase. Fig.4(b) shows histogram $\tilde{\rho}$ of the spike phase $\theta_1(n)$, where area of the histogram is normalized to 1. It can be seen that the histogram $\tilde{\rho}$ mimics waveform of the input $s(\tau)$ for $0 \leq \tau < 1$. Similar results are obtained for $\theta_2(n)$ and $\theta_3(n)$. Hence the property P1 is satisfied. Fig.4(c) shows a Lissajous figure on the (x_1, x_2) -plane. It can be seen that the 1st SG cell and the 2nd SG cell are not synchronized¹. Similar results are obtained for Lissajous figures on the (x_1, x_3) - and the (x_2, x_3) -planes. Hence the SG cells are not synchronized to each other and the property P2 is satisfied.

4 Theoretical Analysis

In this section we theoretically show that the presented SG cells can satisfy the properties P1 and P2. First we derive the following spike position map f of the spike position $\tau_i(n)$ from $\mathbf{R}^+ = [0, \infty)$ into itself:

¹ If they synchronize, the Lissajous figure becomes the diagonal line $x_1 = x_2$.

$$\tau_i(n+1) = f(\tau_i(n)) = \sigma^{-1} (2\sigma(\tau_i(n)) + a - s_0 \text{Int}(\tau_i(n))), \quad (5)$$

where $\text{Int}(\tau)$ givens the integer part of τ , and $n = 1, 2, \dots$. An example of f is shown in Fig.5. In order to analyze dynamics of the spike phase $\theta_i(n)$, we define the following spike phase map F from an interval $[0, 1)$ into itself:

$$\theta_i(n+1) = F(\theta_i(n)) \equiv f(\theta_i(n)) \pmod{1}. \quad (6)$$

Substituting Equation (5) into Equation (6), we obtain

$$F(\theta_i(n)) \equiv \sigma^{-1}(2\sigma(\theta_i(n)) + a \pmod{s_0}). \quad (7)$$

Relation between the spike position map f and the spike phase map F can be seen in Fig.5. The spike phase map F is characterized by the offset s_0 and the amplitude k of the input $s(\tau)$, and the firing threshold a of the SG cell. Some examples of the spike phase map F are shown in Fig.6. Let the density function of the trajectory of $\theta_i(n)$ be denoted by ρ , and let it be called a *phase density*. Now we can give the following.

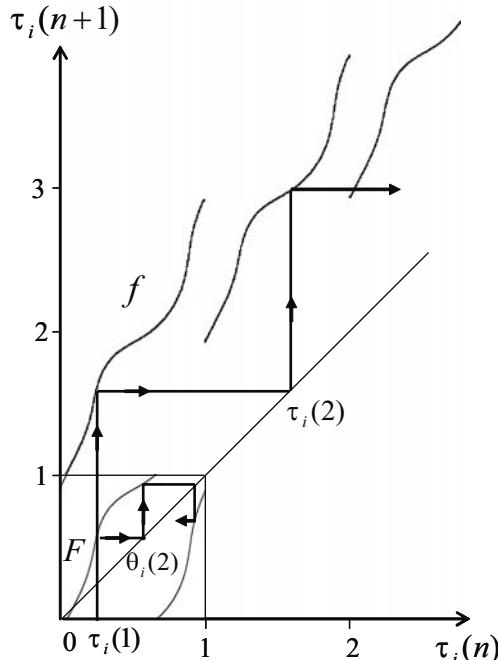


Fig. 5. Spike position map f and spike phase map F . $(s_0, k, a) = (1.2, 0.7, 1)$.

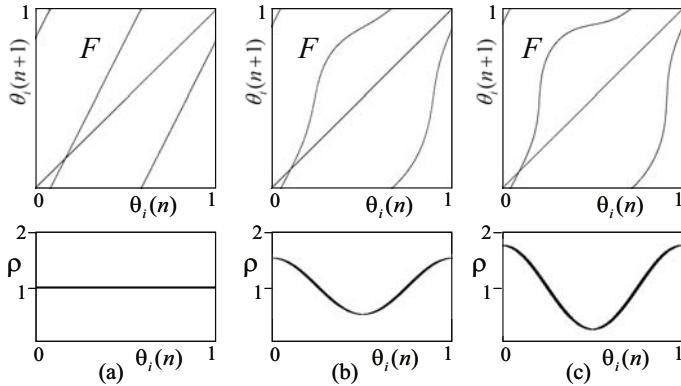


Fig. 6. Spike phase map F and phase density ρ . $(s_0, a) = (1.2, 1)$. (a) $k = 0$. (b) $k = 0.7$. (c) $k = 1$.

Theorem 1: For all $s_0 > k \geq 0$, $a > 0$ and for almost all² $0 \leq \theta_i(1) < 1$, the spike phase map F has the phase density

$$\rho(\theta_i) = \frac{1}{s_0} s(\theta_i). \quad (8)$$

A proof can be given by using the phase map F and will be given elsewhere. Some densities ρ are shown in Fig.6. We can confirm Theorem 1 by comparing the density ρ in Fig.6(b) and the corresponding histogram $\tilde{\rho}$ in Fig.4(b). Theorem 1 says that the density ρ (histogram $\tilde{\rho}$) of the spike phase $\theta_i(n)$ can mimic waveform of the input $s(\tau)$ for $0 \leq \tau < 1$. Hence the SG cell satisfies the property P1. In order to give another theorem, let us introduce the Lyapunov exponent λ_F for the spike phase map F :

$$\lambda_F = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \log_e \frac{dF(\theta_i(n))}{d\theta_i(n)}.$$

A negative λ_F implies that the trajectory $(\theta_i(1), \theta_i(2), \dots)$ converges to a stable periodic attractor. A positive λ_F implies that the trajectory $(\theta_i(1), \theta_i(2), \dots)$ is chaotic.³ Now we can give the following theorem.

Theorem 2: For all $s_0 > k \geq 0$, $a > 0$ and $0 \leq \theta_i(1) < 1$, the Lyapunov exponent λ_F of the spike phase map F is given by

$$\lambda_F = \log_e 2. \quad (9)$$

A proof can be given by using the phase map F and will be given elsewhere. Theorem 2 says that the spike phase $\theta_i(n)$ is chaotic and has sensitivity to the initial state $\theta_i(1)$.

² The term "for almost all $\theta_i(1) \in I$ " means "all the initial states $\theta_i(1) \in I$ except for a subset of I having measure zero" [10]. From a real circuit viewpoint, the probability to be able to put the initial state $\theta_i(1)$ in such an exceptional subset (i.e., the measure zero subset) is zero.

³ There exist various definitions of chaos [10][11]. A positive Lyapunov exponent is one of the most convenient criteria for chaos generation.

Hence the membrane potentials x_i and x_j ($i \neq j$) can not synchronize to each other as shown in Fig.4(c), and the SG cell satisfies the property P2.

Remark 2 (Robustness of encoding): The phase map F is characterized by the input-originated parameters (s_0, k) (i.e., offset and amplitude of the input $s(\tau)$) and the system-originated parameter a (i.e., firing threshold of the SG cell). Theorem 1 says that the input-originated parameters (s_0, k) can be encoded into the phase density ρ . We note that both theorems hold true for any value of the firing threshold a . This means that the encoding function based on P1 and P2 is robust against parameter mismatch of the firing threshold a of each SG cell. The value of the firing threshold a affects to firing rate which we will investigate in a future work.

5 Conclusions

We have presented the circuit model of the SG cell. Using the phase map, we have derived the theorems that can guarantee that the SG cell can realize the encoding function based on the properties P1 and P2. The theorems also guarantee that the encoding function is robust against the parameter mismatch in an implementation circuit. Future problems include the followings: (a) analysis of robustness of the encoding function against various parameter mismatches, (b) development of the SG cell to satisfy other properties of the biological SG cell, and (c) synthesis of an implementation circuit of the SG which is suitable for a future cochlea implant. The authors would like to thank Professor Toshimitsu Ushio of Osaka University for valuable discussions.

References

1. Geisler, C.D.: From sound to synapse: physiology of the mammalian ear. Oxford University Press, Oxford (1998)
2. Izhikevich, E.M.: Dynamical systems in neuroscience. MIT Press, Cambridge (2006)
3. Wei, D., Harris, J.: Signal reconstruction from spiking neuron models. In: Proc. of IEEE/ISCAS, vol. V, pp. 353–356 (2004)
4. Hamanaka, H., Torikai, H., Saito, T.: Quantized spiking neuron with A/D conversion functions. IEEE Trans. CAS-II 53(10), 1049–1053 (2006)
5. Torikai, H., Tanaka, A., Saito, T.: Artificial Spiking Neurons and Analog-to-Digital-to-Analog Conversion. IEICE Trans. Fundamentals 2008 (to appear)
6. Torikai, H., Saito, T.: Resonance phenomenon of interspike intervals from an Integrate-and-Fire Model with two periodic inputs. IEEE Trans. CAS-I 48(10), 1198–1204 (2001)
7. Nishigami, T., Torikai, H.: Basic analysis of a leaky spiking oscillator with two periodic inputs. In: Proc. NOLTA (2008) (to appear)
8. Perez, R., Glass, L.: Bistability, period doubling bifurcations and chaos in a periodically forced oscillator. Phys. Lett. 90A(9), 441–443 (1982)
9. Martignoli, S., van der Vyver, J.-J., Kern, A., Uwate, Y., Stoop, R.: Analog electronic cochlea with mammalian hearing characteristics. Applied Physics Letters 91, 064108 (2007)
10. Devaney, R.L.: An Introduction to Chaotic Dynamical Systems, 2nd edn. Addison Wesley, Reading (1989)
11. Lasota, A., Mackey, M.C.: Chaos, Fractals, and Noise: stochastic aspects of dynamics, 2nd edn. Springer, Berlin (1994)

Evolution of Neural Organization in a Hydra-Like Animat

Ben Jones¹, Yaochu Jin², Xin Yao¹, and Bernhard Sendhoff²

¹ School of Computer Science, University of Birmingham, UK

² The Honda Research Institute, Offenbach, Germany

Abstract. The role of efficient information processing in organizing nervous systems is investigated. For this purpose, we have developed a computational model termed the *Hydramat Simulation Environment*, so named since it simulates certain structural aspects of fresh water hydra. We compare the evolution of neural organization in architectures that remain static throughout their lifetimes and neural architectures that are perturbed by small random amounts. We find that (a) efficient information processing directly contributes to the structural organization of a model nervous system and (b) lifetime architectural perturbations can facilitate novel architectural features.

Keywords: Neural organization, hydra, animat.

1 Introduction

In this paper we explore the influence that energy conservation has on neural organization and the additional effect that *noise* has on this process. In order to do this, we have modeled a radially symmetric organism which loosely resembles the freshwater *Hydra*, which we term ‘hydramat’. We focus on how the nervous system should configure itself around a fixed radially-symmetric body-plan.

Our hypothesis is threefold: the neural architecture of a radially symmetric agent will arrange itself such that (i) the agent is afforded maximal functional benefit and (ii) the agent is afforded the ability to conserve energy and (iii) structural innovations are benefited from lifetime architectural perturbations.

The simulated agent proposed in this work is based on observations of the nervous system of the genus *Hydra*, since crucially, they are phylogenetically the first to have a nervous system that resembles major principles of nervous systems in later organisms [8][10]. In doing so we make a very high-level abstraction yet we point out the differences as we describe them. Our actual simulation environment to be outlined in Section 2 is heavily inspired by Albert, who devised a simple model for the hydra and its nervous system [1]. From a systems biology approach, we have been further inspired by the work of Niv et al, who looked at the foraging behavior of honey-bees [9] and Florian, who employed biologically inspired neural networks for agent control [4]. We provide the simulation results in Section 3 and conclude the paper in Section 4.

2 The Hydramat Simulation Environment

The hydramat is modeled on a tube, and has a nervous system consisting of the following types of cells (also see Fig. 1).

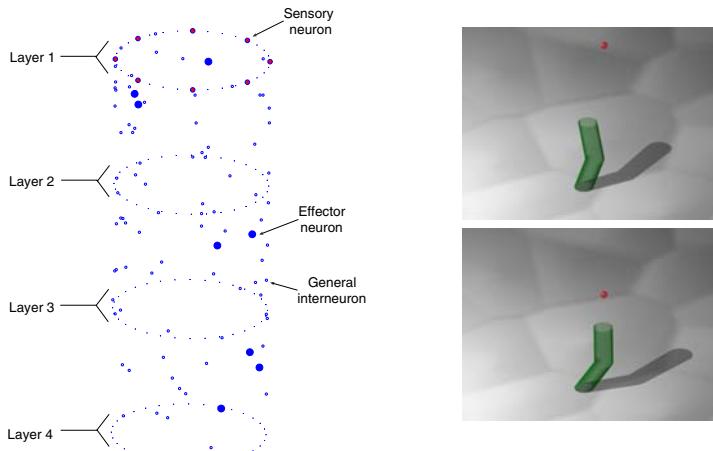


Fig. 1. *Left:* A visualization of the hydramat showing a neural cell distribution; *right:* Two snapshots of the simulation environment. The sphere represents a food particle.

Sensory neurons. These are computational neuron units that remain fixed at the top of the hydramat tube. They detect falling food particles. When a piece of food is dropped, the sensory cell that the piece of food is closest to is the one that ‘spikes’, providing input to the rest of the spiking neural network. The other sensory cells remain dormant.

Effector neurons. Each effector neuron, of which there are always 8, is used to ‘wobble’ the animat in one of eight directions, so that a food particle can be ‘caught’, see Fig. 2. Note that a food particle is deemed caught when it comes to within a small distance threshold of the top of the hydramat. The effector cell that ends up firing the most, is the one that brings about movement whilst the others remain dormant. A firing rate is therefore observed within a short time-frame (10 updates to the simulation environment) to decide the behavior.

General interneurons. Additional neuron units residing within the ‘skin’ of the hydramat. The Euclidean locations of both the interneurons and the effector neurons are evolved throughout a process of simulated evolution and this is the crucial way in which we evolve the architecture of the nervous system. These localities are constrained to reside within the skin of the hydramat, since this is the case in Hydra.

2.1 Movement Dynamics

The hydramat ‘wobbles’ to catch a piece of food by moving its third layer, see Figs. 1 and 2. The other layers passively align themselves to this moving layer. The amount by which a contracting layer moves due to a spiking effector neuron is predefined by a movement magnitude (m), which we always set to 0.8 (see equations in Fig. 2). The other layers iteratively align themselves in proportion to the amount by which layer 3

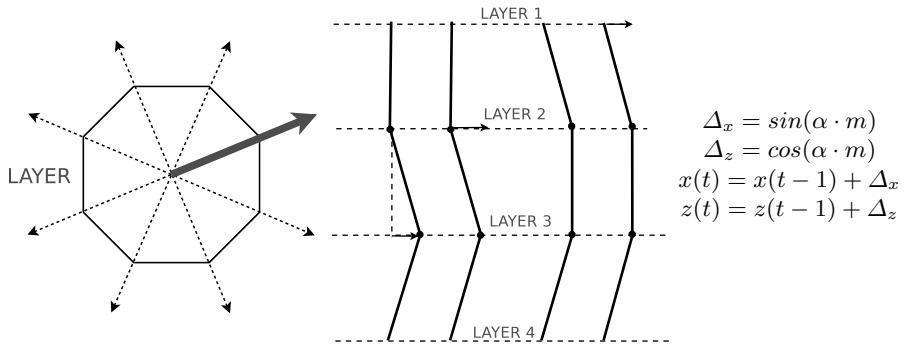


Fig. 2. The movement mechanism of a hydramat layer. A chosen direction of movement has been highlighted for a layer (left-hand diagram). The way in which the hydramat aligns itself following a movement is also depicted (right-hand diagram). The value m is a predefined constant and defines movement magnitude. The value α represents the angle of the directional movement shown in the left-hand diagram.

changed position during such a movement update, (Eqs. 1, 2 and 3). In this respect the passively updating layers observe very simple spring-like dynamics:

$$f_a = d c_a, \quad (1)$$

$$f_b = -d c_b, \quad (2)$$

$$F = f_a + f_b. \quad (3)$$

Note that we employ, f_b , as a kind of relaxation ‘force’, causing the tube to resume to a resting upright state. The parameters, c_a and c_b are predefined constants set to 0.2 and 0.01 respectively.

2.2 Hydramat Nervous System

An integrate and fire spiking neural network. We use the Neuro Simulation Toolkit NEST [5] to build a simple integrate and fire (iaf) model for the hydramat nervous system. A change in membrane potential of a neuron is given by the following differential equation,

$$\tau_i \frac{du_i}{dt} = -u_i + RI_i(t). \quad (4)$$

where I_i is input current and R is resistance. The multiplicative effect is typically equivalent to the accumulated strength of all incoming presynaptic activity i.e. an accumulation of weight values representing the connection strengths multiplied by functions of when presynaptic neurons fired prior to the current ‘time step’ ($\sum_j^N w_{jik}(t - F_j)$). The value τ_i , is the membrane time constant of the neuron.

Neuron positioning and synaptic efficacy. In our model, the distance between a pair of neurons (i and j) determines the connection strength of the connection between them, if such a connection exists. The formula $w_{ji} = \frac{\xi}{d_{ji}}$ is used to derive this value, where d_{ji} is the Euclidean distance between neurons i and j and the value $\xi = 16$ has been set empirically to ensure that w_{ji} has the potential to be significantly large. The maximum value of d_{ji} is given by the bounds of the hydramat's geometrical properties. In our experiments, the length of the cylindrical hydramat and its diameter were always set to 6.0 and ~ 1.6 respectively. Therefore the theoretical ranges of d_{ji} and w_{ji} were $[0, \sim 6.23]$ and $[\sim 0.64, > \xi]$ respectively; the smaller the distance between a pair of neurons, the larger the weight value between them, if a connection exists between them.

Neuron positioning and connectivity. A second aspect of our nervous system model, is that the connectivity between any pair of neurons is determined using a Cauchy probability distribution of the form

$$P_{ji} = \frac{1}{\pi \cdot \lambda_i \cdot \left[\left(\frac{d_{ji}}{\lambda_i} \right)^2 + 1 \right]}. \quad (5)$$

The value of λ_i governs the width and height of the distribution. If the value is small, the shape will be tall and narrow. If it is large, the shape will be short and wide. Each particular neuron-type pair has its own value. Since in the model, there are three neuron types (sensory (S), effector (E) and interneuron (I)), we have six types of connectivity. Given this, the connectivity relation between two neurons is not symmetric. A connection from S to E does not equate to a connection from E to S, i.e., $SE \neq ES$. We actually employ 6+1 λ scalars: one for each of the different connectivities, and a further one since we allow interneurons to connect to other interneurons. Except for the interneurons, a particular type of neuron is never connected to a neuron of the same type.

Finally note that we include an artificial constraint that prevents connections bisecting the hydramat cylinder. This is to ensure that connections remain within the ‘skin’ of the hydramat like it is in real hydra. If a pair of neurons are angled more than 20° away from each other, the connection probability is set to zero.

Lifetime architectural perturbations. In order to assess whether there is an advantage in perturbing the neuronal architecture throughout the lifetime of the hydramat, the neurons are made to move stochastically by small random amounts. We are motivated to do so because in Hydra, the neural cells undergo constant movements (e.g. [B]). At each update of the simulation, a neuron’s position changes with a probability of 0.05, by an amount drawn from a normal distribution having an expectation of zero. For the neuron’s y coordinate gene, the variance of this distribution is set to 0.1 and for the neuron’s angle, 10.0. If during such a movement, the neuron moves outside of the hydramat’s bounds, it is replaced with a neuron occupying the position of the original neuron before any lifetime movements were undertaken.

2.3 Measuring Efficiency

In the hydramat neural model, we consider a single spike as a single unit of information transmission, which is associated with an energy cost, e_j , of the spiking neuron,

$$e_j = \sum_{i=1}^C \frac{d_{ij}}{d_{max}} S_{loss}, \quad (6)$$

where d_{ij} is the distance between two neurons and S_{loss} is the maximum possible energy that will be lost (0.1) when the connection length is d_{max} . Energy is also accumulated whenever the hydramat catches a food particle. In our simulations, this value was set to $F_{gain} = 40.0$. Note, we do not presently take energy loss from movement into consideration; we want to exclusively and concisely focus on energy efficiency within the neural system only, before we look at its significance elsewhere.

2.4 Evolutionary Process

We employ a simple evolutionary algorithm to evolve the neuron positions and connectivity parameters. Typically we have 8 sensory neurons, 8 effector neurons and 84 interneurons. So we have 92 neuron positions to evolve (since the sensory neurons are fixed) and the 7 lambda connectivity parameters. We employ binary tournament selection, discrete recombination and a Gaussian mutation with an adaptive standard deviation (σ). The selection scheme randomly selects pairs of individuals and then selects the fitter of the two until a new population is established; discrete recombination exchanges gene values between two offspring candidates selected randomly from the offspring population. The adaptation process relies on the setting of two strategy parameters, $\tau_0 = 1.0/\sqrt{2.0 * D}$ and $\tau_1 = 1.0/\sqrt{2.0 * \sqrt{D}}$ where D represents the dimensionality of the vector being evolved. These values have been shown to be optimal in a process of self-adaptation [2]. The σ values are adapted as follows,

$$\sigma_i \leftarrow \sigma_i * \exp(N(0, \tau_0) + N_i(0, \tau_1)). \quad (7)$$

Both genes representing the spatial positioning vectors and those representing the connectivity parameters are subsequently normally mutated by applying the corresponding σ value.

2.5 Experiment Overview

We have devised a very simple ‘food catching’ task to explore our hypothesis. In Fig. 3 a birds-eye perspective of the environment is depicted. At the center of the environment resides the hydramat, whose task is to catch each of the eight pieces of food depicted as filled circles (one at a time). The performance (fitness) measure is the amount of energy that can be retained. Each hydramat population member starts with 200 units of energy and units are gained whenever a piece of food is caught and lost whenever a neuron spikes, if energy efficiency is considered. In order to ensure that the spiking neural network truly evolves to react to the food stimuli, the order in which the food pieces are dropped is randomly shuffled at the turn of each population member.

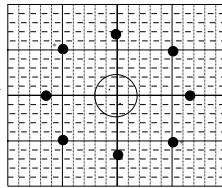


Fig. 3. The 8-food-task environment from a birds-eye perspective. Each of the 8 pieces of food, represented as an outer filled circle, is equidistant from the center of the hydramat, located at the center of the diagram.

3 Summary of Results

Four sets of experiments have been performed to investigate the influence of energy efficiency and lifetime architectural perturbations on neural organization. Fig. 4 shows the energy conserved in the four experimental setups. We can see that evolution progresses steadily in all experiments and when energy consumption due to spikes is not taken into account, energy conserved from food caught increases more rapidly. Fig. 5 presents the relationship between energy conserved and connectivity density. From the figure, we can conclude that minimizing energy loss due to spikes effectively reduces the density of neural connectivity, which, however, becomes harder when lifetime architectural perturbations exist. The optimized neural architectures in the four experiments are illustrated in Fig. 6. From these results, we can make the observation that effector neurons (enlarged filled circles) tend to aggregate around the sensory neurons (small filled circles). This suggests that functionally significant neurons have a tendency to become proximal to each other, which might be a functionally useful feature. Moreover, significance tests based on 30 runs when energy efficiency is considered indicate that lifetime perturbations increased the likelihood that effector neurons would move towards the top of the hydramat, to a significance of 0.005.

Two main findings seem to have emerged from our simulation results. First, energy has a direct impact on neural organization. Second, lifetime neural perturbations can facilitate the emergence of biologically plausible structures.

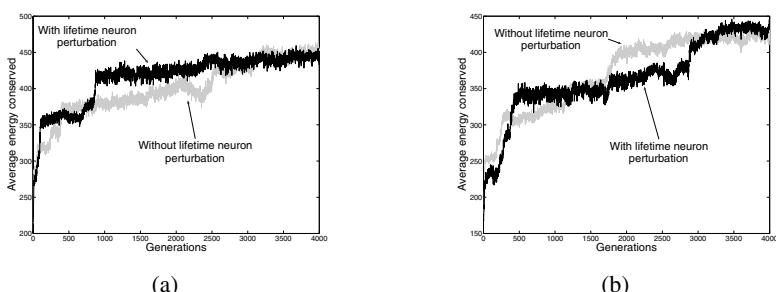


Fig. 4. Energy conserved as evolution proceeds (a) without energy efficiency, and (b) with energy efficiency taken into account

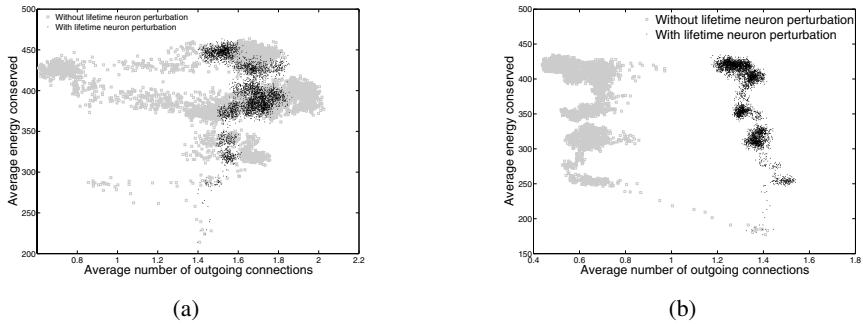


Fig. 5. Energy conserved versus the number of outgoing connections (averaged over the whole population) when energy loss due to spikes is (a) not considered and (b) considered

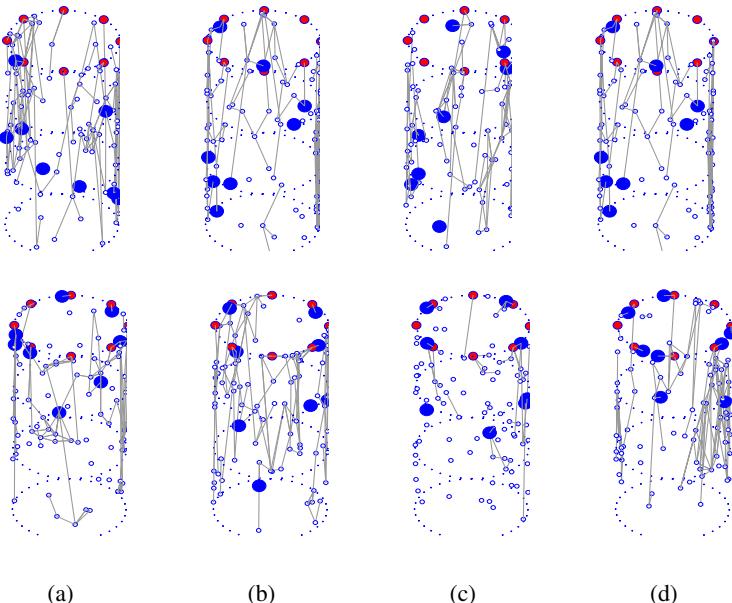


Fig. 6. Initial (*upper*) and optimized (*lower*) neural architectures of the four experiments indicate that effector neurons (larger filled circles) tend to move toward the sensory neurons (smaller filled circles). (a) energy efficiency not considered, no architectural perturbations; (b) energy efficiency not considered, with architectural perturbations; (c) energy efficiency considered, no architectural perturbations; and (d) energy efficiency considered, with architectural perturbations.

4 Conclusions

The contributions of this work are as follows. First, we suggest a novel framework that incorporates energy efficiency in organizing nervous systems. This framework allows us to take the first steps in understanding evolutionary transitions of primitive nervous systems, which provides interesting future research direction. Second, our computational

results demonstrate how efficiency of information processing and minimization of energy consumption has emerged in evolution through the interactions with the environment. Finally, our results show how random perturbations can facilitate the emergence of novel nervous architectures.

It seems sensible to expect that the same holds for the biological radially symmetric organisms of the genus *hydra*, i.e., that the formation of ring structures around the hypostomal region, [7], is a result of selection pressure towards minimalistic structures which coupled with their body morphology leads to similar structures. A further view is that the noisy effect of lifetime neural perturbations might increase robustness of the information processing system.

There are three other extensions of the model presented here. Firstly, we can incorporate simulations of gene regulatory networks and let the neural architectures grow instead of directly encoding spatial information. Secondly, we can allow morphological changes of the hydramat and see how nervous systems depend on and change with body shape; we have begun to approach this in [6]. Thirdly, with this newer model, we can also consider how energy is lost due to the actual movements of the animat.

Acknowledgements

The authors are grateful to J. Eppler and M.-O. Gewaltig for their assistance in using NEST. We also thank U. Körner for her support in understanding the more detailed spike-based processes in *Hydra*. The first author is grateful to Honda Research Institute Europe for a PhD studentship that makes this work possible.

References

- Albert, J.: Computational modeling of an early evolutionary stage of the nervous system. *BioSystems* 54(1–2), 77–90 (1999)
- Bäck, T., Schwefel, H.-P.: An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation* 1(1), 1–23 (1993)
- Bode, H.R.: Continuous conversion of neuron phenotype in *hydra*. *Trends in Genetics* 8(8), 279–284 (1992)
- Florian, V.R.: Biologically inspired neural networks for the control of embodied agents. Technical report, The Center for Cognitive and Neural Studies, Cluj-Napoca, Romania (2003)
- Gewaltig, M.-O., Diesmann, M.: NEST. Scholarpedia (2007)
- Jones, B., Jin, Y., Sendhoff, B., Yao, X.: Evolving functional symmetry in a three dimensional model of an elongated organism. In: *Proceedings, ALife XI*, Winchester, UK, pp. 305–312 (2008)
- Koizumi, O.: Developmental neurobiology of *hydra*, a model animal of cnidarians. *Canadian Journal of Zoology* 80(10), 1678–1689 (2002)
- Meinhardt, H.: The radial-symmetric *hydra* and the evolution of the bilateral body plan: an old body become a young brain. *BioEssays* 24, 181–191 (2002)
- Niv, Y., Joel, D., Meilijson, I., Ruppin, E.: Evolution of reinforcement learning in uncertain environments: a simple explanation for complex foraging behaviors. *Adaptive Behavior* 10(1), 5–24 (2002)
- Swanson, L.W.: The Simplest Nervous Systems. In: *Brain Architecture: Understanding the Basic Plan*. Oxford University Press, New York (2003)

Improved Sparse Bump Modeling for Electrophysiological Data

François-Benoit Vialatte¹, Justin Dauwels^{1,2}, Jordi Solé-Casals³,
Monique Maurice¹, and Andrzej Cichocki¹

¹ Riken BSI, Wako-Shi, Japan
fvialatte@brain.riken.jp

² Massachusetts Institute of Technology, Cambridge, MA, USA

³ University of Vic, Vic, Spain

Abstract. Bump modeling is a method used to extract oscillatory bursts in electrophysiological signals, who are most likely to be representative of local synchronies. In this paper we present an improved sparse bump modeling method. The improvements are done in the adaptation method by optimizing the parameters according to the order of their derivatives; and in the window matching method by changing the selection of the initial function. Experimental results, comparing previous method *vs* the improved version, show that the obtained model fits better the signal, hence the result will be much more precise and useful.

1 Introduction

Brain rhythms are traditionally described by different oscillatory regimes including theta (4-10 Hz), beta (15-40 Hz) and gamma (40-90 Hz) rhythms. The structural organization and associated functional role of these electroencephalographic (EEG) oscillations are still far from being completely understood. Oscillatory activity can be separated in background and burst pattern activities. The background EEG is constituted by regular waves, whereas bursts are transient and with higher amplitudes. These bursts are organized activities, most likely to be representative of local synchronies. They should consequently play a specific functional role, distinct from background EEG activity. EEG event related synchronization (ERS), and similarly event related desynchronization (ERD), can be interpreted as the reorganization of the spontaneous brain oscillations in response to the stimulus [1][2].

Once extracted, the activities can be used as features for classification (they can for instance feed neural network classifiers, see *e.g.* [7]) and signal analysis. However, extracting these activities with a reliable and consistent method is not a simple task. Therefore sparse bump modeling, a 2D extension of the 1D bump modeling described in [10], was developed. It was first applied to invasive EEG (local field potentials), recorded from rats olfactory bulb during a go-no go olfactory memory task [16][15]. Afterwards, it was used to investigate several aspects of brain oscillatory dynamics: scalp EEG data from patients with early stage of

Alzheimer's disease (AD) [13][12][15][3]; simultaneous time-frequency-space representation using a sonification approach [11]; a synchrony model of time-frequency bursts events (stochastic event synchrony, [4][5][6]; modeling oscillations of steady state visual event potential epochs [14].

Bump modeling adapts parametric functions, within selected windows, to the wavelet time-frequency representation of a given signal. The present paper seeks to improve the matching algorithm of bump modeling, by the introduction of improved methods of adaptation and window selection.

2 Methods

2.1 Bump Modeling Procedure

Bump modeling follows the following four steps (rationales for this procedure, proofs and technical details are explained in [15]):

1. the signal is first wavelet transformed into time-frequency maps.
2. the time-frequency map is z-score normalized.
3. the map is described by a set of time-frequency window (which could be seen as a low complexity ‘matching’ step if compared against the matching-pursuit method [8]).
4. parametric functions are adapted within these windows, in decreasing ordered of energy.

2.2 Wavelets

We scale complex Morlet wavelet ϑ to compute time-frequency wavelet representations of the signal \mathbf{X} of dimension T :

$$C_x(t, s) = \int \mathbf{X}(\tau) \vartheta^* \left(\frac{\tau - t}{s} \right) d\tau \quad (1)$$

where s , the scaling factor, controls the central frequency f of the mother wavelet. The modulus of this time-scale representation can therefore be used as a positive time-frequency spectrogram, which we will note \mathbf{C}_x , a time-frequency matrix of dimension $T \times F$.

2.3 Z-Score

The time-frequency spectrogram is normalized depending on a reference signal. The reference signal \mathbf{R} of dimension U is wavelet transformed into a spectrogram \mathbf{C}_r of dimension $U \times F$. The average amplitudes $\mathbf{M}_f = [\mu_1, \mu_2, \dots, \mu_F]$ and standard deviations of amplitude $\mathbf{S}_f = [\sigma_1, \sigma_2, \dots, \sigma_F]$ are computed from \mathbf{C}_r for each of the F frequencies. The z-scored map \mathbf{Z}_x is obtained through normalization using these values:

$$\mathbf{Z}_x(f, t) = \frac{\mathbf{C}_x(f, t) - \mu_f}{\sigma_f} \quad (2)$$

2.4 Z-Score Offset

We will refer to the positive z-score values as ERS-like components: if the signal is recorded during a stimulation, these oscillatory peaks are the most likely constituent of ERS. If the signal is recorded in pre-stimulus period or in rest condition, these oscillations are representative of organized oscillatory bursts. Z-score returns values in \mathbb{R} , but bump modeling only accepts values in \mathbb{R}^+ as inputs. In order to model ERS-like oscillations, we reject the negative components of the map with a threshold, the z-score offset ϕ . The thresholded map \mathbf{Z}_x^ϕ is obtained with:

$$\mathbf{Z}_x^\phi = \frac{1}{2} [(\mathbf{Z}_x + \phi) + |(\mathbf{Z}_x + \phi)|] \quad (3)$$

Usual values of ϕ are in the [0-3] range.

We will refer to the negative z-score values as ERD-like components: if the signal is recorded during a stimulation, these oscillatory peaks are the most likely constituent of ERD. If the signal is recorded in pre-stimulus period or in rest condition, these oscillations are representative of a local disorganization of oscillatory activity. These negative z-score values in \mathbb{R}^- can be extracted into a threshold map \mathbf{Z}_x^- :

$$\mathbf{Z}_x^- = -\frac{1}{2} [|(\mathbf{Z}_x)| - (\mathbf{Z}_x)] \quad (4)$$

2.5 Windowing

The map \mathbf{Z}_x is described by a set of windows $\omega(s, \tau)$ with $s \in \mathbf{S}$, $\tau \in \mathbf{T}$ the position on the general time-frequency map respectively in scale and step; each ω has the dimensions $H \times W$ (height and width) determined depending on the time-frequency resolution at the window's central frequency (see [13]).

2.6 Parametric Functions

We use half ellipsoid functions to model the normalized time-frequency map. The half ellipsoid outskirts are defined as:

$$\Psi(A, h, w, f, t, y, x) = 1 - \frac{(y - f)^2}{h^2} + \frac{(x - t)^2}{w^2} \quad (5)$$

The half ellipsoid bump is obtained with:

$$\xi(A, h, w, f, t, y, x) = \begin{cases} 0 & \text{if } \Psi < \lambda \\ A \cdot \sqrt{\Psi} & \text{else} \end{cases} \quad (6)$$

Hence the adaptation error to be minimized will be:

$$E(A, h, w, f, t, y, x) = \sum_{x=1}^W \sum_{y=1}^H \|\omega_{y,x}(s, \tau) - \xi(A, h, w, f, t, y, x)\|^2 \quad (7)$$

2.7 Improved Adaptation

Derivatives of half ellipsoid bumps are always null when $\Psi(A, h, w, f, t, y, x) < \lambda$. For $\Psi(A, h, w, f, t, y, x) \geq \lambda$, they take the following values:

$$\frac{dE}{dA} = 2\sqrt{\Psi} \left(A\sqrt{\Psi} - \omega_{y,x}(s, \tau) \right) = -2\sqrt{E}\sqrt{\Psi} \quad (8)$$

$$\frac{dE}{dh} = \frac{2(y, f)^2 A \sqrt{\Psi}}{h^3} \left(A\sqrt{\Psi} - \omega_{y,x}(s, \tau) \right) = -2\sqrt{E} \frac{(y-f)^2 A}{h^3 \sqrt{\Psi}} \quad (9)$$

$$\frac{dE}{dw} = \frac{2(x-t)^2 A \sqrt{\Psi}}{w^3} \left(A\sqrt{\Psi} - \omega_{y,x}(s, \tau) \right) = -2\sqrt{E} \frac{(x-t)^2 A}{w^3 \sqrt{\Psi}} \quad (10)$$

$$\frac{dE}{df} = \frac{2(y-f) A \sqrt{\Psi}}{h^2} \left(A\sqrt{\Psi} - \omega_{y,x}(s, \tau) \right) = -2\sqrt{E} \frac{(y-f) A}{h^2 \sqrt{\Psi}} \quad (11)$$

$$\frac{dE}{dt} = \frac{2(x-t) A \sqrt{\Psi}}{l^2} \left(A\sqrt{\Psi} - \omega_{y,x}(s, \tau) \right) = -2\sqrt{E} \frac{(x-t) A}{l^2 \sqrt{\Psi}} \quad (12)$$

Improved Adapation can be obtained by optimizing the parameters stepwise, with a priority depending on the order of their derivatives. The comparison of the above derivatives is obvious: the term $-2\sqrt{E\Psi}$ being always common to all these derivatives, and the other terms being defined in \mathbb{R}^+ , they will all have the same sign. The slope of the adaptation will then be dependant on the multiplicands m applied to $-2E\sqrt{\Psi}$:

1. $m = 1$ for dE/dA ;
2. m is a positive value in $[0 - A]$ in numerator divided by a variable of order 3 for dE/dh and dE/dw ;
3. m is a positive value in $[0 - A]$ in numerator divided by a variable of order 2 for dE/df and dE/dt .

This would probably be working correctly with properly normalized parameters, however we are here adapting parameters of different ranges: $A \in [0 - 1]$, while h and $f \in [1 - H/2]$ and w and $t \in [1 - W/2]$ with H and W usually $>> 1$. Therefore, the multiplicands corresponding to the three above case will be of the order $m \in O(1)$ in case (1), and $m \in O(x^{-3})$ in case (2) and $m \in O(x^{-2})$ in case (3). Practically speaking, it means that the parameters adaptation should be performed stepwise (first h and w , then f and t , and finally A). In the previous implementation of bump modeling [15], we optimized all these parameters simultaneously (first using iterations of first order gradient descent, followed by iterations of the BFGS [9] algorithm). We improve the quality and speed of convergence by performing the following stepwise estimation of these parameters:

1. Update h and w until both their derivatives are below a threshold t_Ψ .
2. Update f and t until both their derivative are below a threshold t_{pos} . If at anytime dE/dh or dE/dw becomes above t_Ψ , go back to 1.

3. Update only A , until its derivative is below a threshold t_A . If at anytime dE/dh or dE/dw becomes above t_Ψ , go back to 1. If at anytime dE/df or dE/dt becomes above t_{pos} , go back to 2.

The adaptation is still performed using the BFGS [9] algorithm. We used $t_\Psi = 10^{-5}$.

2.8 Improved Window Matching

The first step of bump modeling consist in the selection of a window within the boundaries of which the function is estimated. In the previous version of bump modeling [15], the best candidate window Ω was selected as:

$$\Omega = \arg \max_{s,\tau} \left[\sum_y \sum_x \omega_{y,x}(s, \tau) \right] \quad (13)$$

Because these windows are used to determine the initial condition of the function adaptation, finding the best suitable window is primordial. The new optimized method is more related to matching pursuit [8], in that we will match the window content with the initial bump function $\xi_{s,\tau}(y, x) = \xi(A_{s,\tau}, h_{s,\tau}, l_{s,\tau}, f_{s,\tau}, t_{s,\tau}, y, x)$ with $f_{s,\tau} = h_{s,\tau} = H/2$, $x_{s,\tau} = l_{s,\tau} = L/2$, and $A_{s,\tau}$ the highest peak in the window:

$$A_{s,\tau} = \max_{x,y} [\omega_{y,x}(s, \tau)] \quad (14)$$

We compute the matrix $\Xi_{s,\tau}$ of values of this initial function for each time-frequency points of the window $\omega(s, \tau)$. The best window is thus matched as:

$$\Omega = \arg \max_{s,\tau} \left[\frac{\Xi_{s,\tau} : \omega_{s,\tau}}{\| \Xi_{s,\tau} \|_F} \right] \quad (15)$$

Where $:$ denotes the Frobenius inner product, and $\| . \|_F$ indicates the Frobenius norm. This generalize the matching criterion of matching pursuit methods to 2D data. For bump modeling high amplitude has priority on best fit. Therefore, contrary to matching pursuit, the product is not normalized by the norm of $\omega_{s,\tau}$.

2.9 EEG Data

We demonstrate the improved method with data collected from 23 patients with Alzheimer's disease. This is the same data that was modeled with bumps in [12]. EEG recording was done in an awake resting state with eyes closed, under vigilance control. Ag/AgCl electrodes (disks of diameter 8 mm) were placed on 21 sites according to 10/20 international system, with the reference electrode on the right ear-lobe. EEG was recorded with Biotop 6R12 (NEC Sanei, Tokyo, Japan) using sampling frequency of 200Hz.

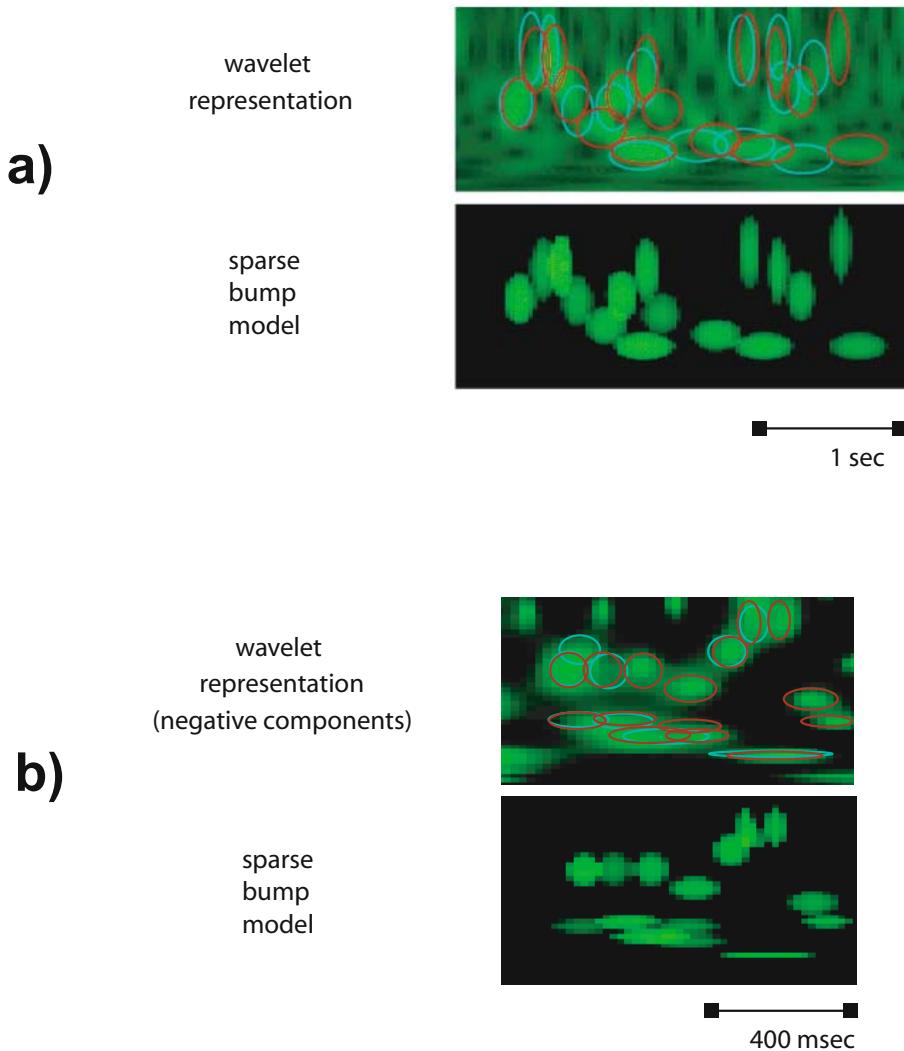


Fig. 1. Improved bump modeling, as compared to classical bump modeling. a) ERS-like components ($\phi = 3$), b) ERD-like components. Top: Wavelet transform, with superimposed positions of the previous model (blue) and the improved model (red). Bottom: improved bump model.

3 Results and Discussion

The purpose of the present paper was to describe an improved method for sparse time-frequency bump modeling, which addresses efficiently the problem of detecting automatically reproducible timefrequency burst events in electrophysiological data. We extracted ERS-like and ERD-like components from the

EEG signals using the improved modeling (improved adaptation and/or window matching), as compared with the previous method. The improved bump modeling allowed a more reliable extraction of time-frequency oscillatory bursts events as compared to the method described in [15]. We observed that:

1. The original algorithm already fits nicely the parametric functions to the map.
2. The improved window matching mainly changes the adaptation order, but sometimes selects windows with better location than the previous method. The difference with the previous method (*b vs. c*) is subtle but significant.
3. The improved adaptation fits better the parametric function's shapes to the signal (*b vs. d*).
4. With both improved methods, the representation becomes even more precise (*b vs. e*).

In order to illustrate this difference, we show on Fig. 10 an example of time-frequency modeling with a comparison of the previous and new model (ERS-like components, with $\phi = 3$, and ERD-like components).

Acknowledgments

Jordi Solé-Casals was supported by the Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya, and by the Spanish grant TEC2007-61535/TCM. Monique Maurice is supported by the JSPS fellowship P-08811. Justin Dauwels is supported by the Henri-Benedictus fellowship from the King Baudouin Foundation and the BAEF.

References

1. Başar, E.: EEG-brain dynamics: Relation between EEG and brain evoked potentials. Elsevier, Amsterdam (1980)
2. Başar, E., Demirrap, T., Schürmann, M., Başar-Eroglu, C., Ademoglu, A.: Oscillatory brain dynamics, wavelet analysis, and cognition. *Brain and Language* 66, 146–183 (1999)
3. Buscema, M., Rossini, P., Babiloni, C., Grossi, E.: The ifast model, a novel parallel nonlinear eeg analysis technique, distinguishes mild cognitive impairment and alzheimer's disease patients with high degree of accuracy. *Artificial Intelligence In Medicine* 40, 127–141 (2007)
4. Dauwels, J., Vialatte, F., Cichocki, A.: A novel measure for synchrony and its application to neural signals. In: Proceedings of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007), Honolulu, USA, April 15-20, vol. IV, pp. 1165–1168 (2007)
5. Dauwels, J., Vialatte, F., Cichocki, A.: On synchrony measures for the detection of alzheimer's disease based on eeg. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part I. LNCS, vol. 4984, pp. 112–125. Springer, Heidelberg (2008)

6. Dauwels, J., Vialatte, F., Rutkowski, T.M., Cichocki, A.: Measuring neural synchrony by message passing. In: Advances in Neural Information Processing Systems, NIPS 2007, Vancouver, Canada, December 6-9, 2007 (2008)
7. Kasabov, N.: Evolving connectionist systems: The knowledge engineering approach. Springer, Heidelberg (2007)
8. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing 12, 3397–3415 (1993)
9. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical recipes in C: The art of scientific computing, pp. 425–430. Cambridge University Press, New York (2002)
10. Dubois, R., Maison-Blanche, P., Quenet, B., Dreyfus, G.: Automatic ecg wave extraction in long-term recordings using gaussian mesa function models and non-linear probability estimators. Comput. Methods Programs Biomed. 88(3), 217–233 (2007)
11. Vialatte, F., Cichocki, A.: Sparse bump sonification: a new tool for multichannel eeg diagnosis of mental disorders; application to the detection of the early stage of alzheimer's disease. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4234, pp. 92–101. Springer, Heidelberg (2006)
12. Vialatte, F., Cichocki, A., Dreyfus, G., Musha, T., Rutkowski, T., Gervais, R.: Blind source separation and sparse bump modelling of time frequency representation of eeg signals: New tools for early detection of alzheimer's disease. In: Proceedings of the IEEE Workshop on Machine Learning for Signal Processing 2005 (MLSP 2005), Mystic CT, USA, September 28-30 (2005)
13. Vialatte, F., Cichocki, A., Dreyfus, G., Musha, T., Shishkin, S.L., Gervais, R.: Early detection of alzheimer's disease by blind source separation, time frequency representation, and bump modeling of eeg signals (invited presentation). In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 683–692. Springer, Heidelberg (2005)
14. Vialatte, F., Dauwels, J., Rutkowski, T.M., Cichocki, A.: Oscillatory event synchrony during steady state visual evoked potentials. In: Springer (ed.) Advances in Cognitive Neurodynamics, Proceedings of the First International Conference on Cognitive Neurodynamics (ICCD 2007), Shanghai, China, November 17-21, 2007 (2008)
15. Vialatte, F., Martin, C., Dubois, R., Haddad, J., Quenet, B., Gervais, R.: A machine learning approach to the analysis of time-frequency maps, and its application to neural dynamics. Neural Networks 20, 194–209 (2007)
16. Vialatte, F., Martin, C., Ravel, N., Quenet, B., Dreyfus, G., Gervais, R.: Oscillatory activity, behaviour and memory, new approaches for lfp signal analysis. In: Acta Neurobiologiae Experimentalis 2003. Proceedings of the 35th annual general meeting of the European Brain and Behaviour Neuroscience Society (EBBS 2003), Barcelona, Spain, September 17-20, vol. 63 (2003)

Classify Event-Related Motor Potentials of Cued Motor Actions

Somnuk Phon-Amnuaisuk

Special Interest Group: Perception and Simulation of Intelligent Behaviours,
Faculty of Information Technology, Multimedia University,
Jln Multimedia, 63100 Cyberjaya, Selangor Darul Ehsan, Malaysia
somnuk.amnuaisuk@mmu.edu.my

Abstract. Motor related potentials are generated when an individual is engaged in a task involving motor actions. The transient post-synaptical potential could be observed from the recorded electroencephalogram (EEG) signal. Properties derived from time domain and frequency domain such as event-related motor potential and suppression in band power could be useful EEG features. In this report, *lateralised motor potential (LMP)* and *band power ratio (BPR)* are used to classify cued left-fingers and right-fingers movements. Two classifiers are employed in this experiment: minimum distance classifier (MDC) and normal density Bayes classifier (NDBC). The results show that the features from LMP has more discriminative power than band power ratio. They also show that NDBC has a perfect performance in this task.

1 Introduction

It is known that event-related potentials (ERP) synchronise with the presentation of external or internal stimuli. Auditory, visual, somatosensory, motor evoked potentials and mental tasks are common ERP topics. From recent literatures, studying motor potentials seems to be one of the popular topics. Experiments on ERPs from motor tasks usually involve imaginary or actual movement of the limbs such as left and right hands and legs movements.

ERP has long been the subject of investigation and many text books have reported detailed morphologies of ERP (e.g., see [4]). Recently, due to decreasing computing cost and the advancement in signal-processing technique, studying of ERP through EEG is made accessible by many small research centers around the world. Most recent ERP studies in informatics research groups focus their studies on building classifiers either from various computing techniques (e.g., Linear discriminant Analysis, Self Organising Map, Support vectors machine, etc.); or from various features (e.g., time domain patterns, frequency band power, etc.); see [5], [1] for examples of computing techniques and features extraction techniques.

In this study, we investigate ERP patterns observed from a subject engaged in a cued motor task. We focus on (i) features extracted from movement-related potential LMP and (ii) changes in band power which is closely related to ERD

(event-related desynchronisation). Understanding the behaviours of these features could lead to better choices of features. Hence, better performances of the classifiers. We organise the presentation to the following topics: (i) Introduction, (ii) Background, (iii) Materials and Methods, (iv) Result and (v) Conclusion.

2 Background

Electroencephalogram (EEG) is a potential recorded on the scalp. These potentials originate from cognitive activities where activated neurons fire and act as electromagnetic dipoles. Event-related potentials (ERP) may, therefore, manifest mental activities. ERPs are usually the combined effects of many sensory stimuli. Hence, studying the effect of a particular stimulus involves the averaging of many repeated trials. Up to a hundred repeated trials are common in this kind of study. This also depends on the signal-to-noise ratio of the acquisition EEG signal.

Neurophysicologists believe *parietal* cortex area plays a major role in motor planning and actions. Right-fingers and left-fingers motor actions ERP should be best detected by placing electrodes in the parietal area (i.e., C3, C4 in the standard 10-20 nomenclature). It is known that the execution of motor actions induce a time-locked response that could be observed from EEG signals. It is also known that the execution of motor action suppresses the ongoing μ rhythm and β rhythms. The time lock response in time domain amplitude and frequency domain amplitude are, therefore, common features for classification tasks [6].

Extracted features from time domain and frequency domain could be used to build a classifier for the classification of EEG signal, Hjorth and Barlow are among the pioneers in clinical EEG analysis, Hjorth [3] suggests that parameters such as *activity*, *mobility* and *complexity* to be the basic properties for time domain analysis of EEG. Goncharowa and Barlow [4] suggest that parameters such as *mean amplitude*, *mean frequency* and *spectral width* to be the features for analysis of spontaneous alpha blocking. In this study, differences in lateralised motor potentials and differences in band power ratio are extracted from time domain and frequency domain respectively.

3 Materials and Methods

3.1 Experimental Set Up

Cued motor tasks. A healthy male subject aged 45 participated in the two-hour session experiment. Four bipolar EEG channels were acquired on the following positions: C3, C4, Fz and Cz (in 10-20 nomenclature). The subject was seated in a comfortable position with eyes open, approximately 0.5 meter from the computer screen. The subject was instructed to look at a blank computer screen. Three visual motor cues were displayed on screen and the subject was to perform according to the cues (i.e., left, right and baseline). These actions were (i) perform a brisk movement of left hand fingers without arm movement

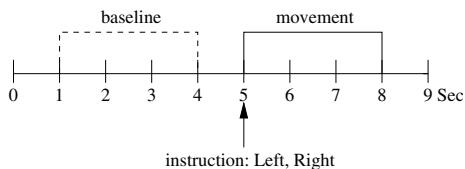


Fig. 1. Flow of experiment

(no specific direction is given on how this should be done, the subject may tap, may wiggle all fingers, may clench, etc.), (ii) perform a brisk movement of right hand fingers and (iii) all fingers are at rest. The actions were to be carried out in synchronisation with the visual cues. Figure 1 below shows the timing diagram of the experiment.

Rationale for the study. With these experiments, we expect to observe and identify the correlations between LMP, band power changes, etc., and the tasks (e.g., left hand fingers, right hand fingers and rest). The stronger the correlation between the patterns and the tasks, the better the accuracy of the classifiers built from these patterns.

3.2 Data Acquisition and Pre-processing

Four EEG data channels were acquired from four areas C3, C4, Fz and Cz. A single ground electrode was affixed at FPz. The cup electrodes were used and placed in a bipolar fashion, for example, channel C3 were acquired from two electrodes, one positive electrode affixed 2.5 cm from C3 toward FC3 and one negative electrode (reference electrode) affixed 2.5 cm from C3 toward CP3. The hair around the area was shaved. The impedance of each electrode was monitored before and after the sessions to confirm that the impedance was lower than 2.5Kohm . The specifications of the acquisition hardware claim that C3 and C4 acquired the EEG data with 256 Hz sampling rate, 0.5 to 100 Hz bandpass and with a sensitivity of $+/- 500\mu\text{V}$, while Fz and Cz acquired EEG data with the same sampling rate, but with 0.01 to 100 Hz bandpass and with a sensitivity of $+/- 2000\mu\text{V}$.

We carried out twelve sessions for a cued motor action task. Ten left hand finger movements and ten right hand finger movements were recorded for each session. This total up to 120 left hand finger and 120 right hand finger movement epochs.

Pre-processing. Raw EEG data were visually inspected and contaminated data were removed from the data set. Figure 3 shows the averaged raw signal of left hand, right hand and baseline epochs respectively. The windows from both cases were from 1 second before the motor cue to 2 seconds after the cue. From the averaged raw data, we could see a clearer negativity in all the channels around 400 ms after the onset time for the right hand epoch and the left hand epochs.

Wavelet denoise. EEG signal has a low signal to noise ratio. In the evaluation process, raw signal as well as wavelet-denoised signal were employed to verify the robustness of the classifiers. For this experiment, the evaluations were carried out with both raw-signal and with wavelet denoised signals (denoise using Daubechies wavelet decomposed at levels 1 and 4).

3.3 Building Classifiers

Features Extraction. Features are commonly extracted from time domain and frequency domain. In our experiment, we found that although there was a clear difference in the μ rhythm between background FFT spectrum and motor actions (see figure 2), it was disappointing to see that the (*global*) FFT spectrum from the left fingers and the right fingers epochs were similar (did not show the FFT spectrum from the right fingers here due to space limitation). For this reason, we built our classifiers from a better *localised* time domain LMP and from short time Fourier band power which should give better localised features.

Calculate LMP and BPR. Each left fingers and right fingers EEG epoch is three seconds long. We represent them as $X^n(i, j)_L$ and $X^n(i, j)_R$ respectively; i is the channel index corresponding to $\{C3, C4, Fz, Cz\}$, j denotes trial numbers, n denotes the number of samples, L denotes left fingers and R denotes right fingers. ERP is calculated by averaging all epochs together (see equations 1, 2) and band power ratio is calculated from the alpha band (8-12 Hz). Band power

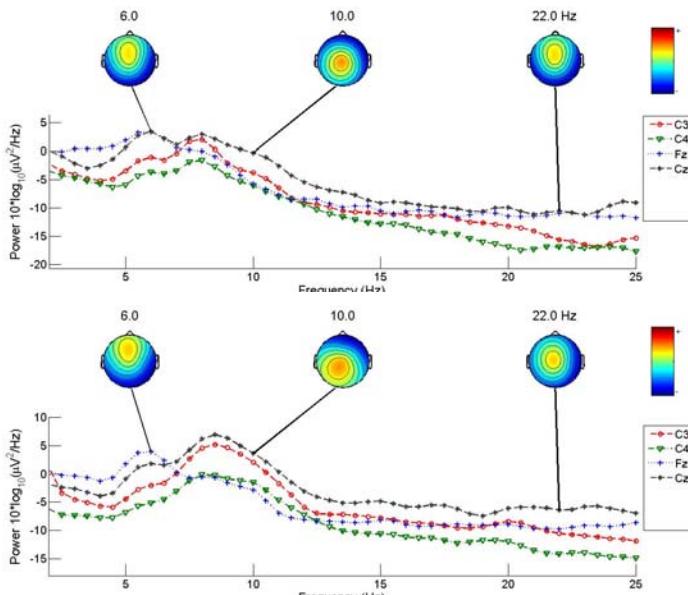


Fig. 2. Averaged frequency spectrum over Left fingers movement and baseline epochs

ratio (BPR) is the ratio of the current band power to the reference band power. The reference band power is the averaged power over a predefined window length before the onset time. Finally, the difference between the BPR of C3 and C4 is used as the feature (see equations 3, 4 and 5).

$$X_{ERP}^n(i, j) = \frac{1}{M} \sum_{j=1}^M X^n(i, j) \quad (1)$$

$$X_{LMP}(j) = X_{ERP}^n(C3, j) - X_{ERP}^n(C4, j) \quad (2)$$

$$X^k(i, j) = \sum_{n=0}^{N-1} X^n(i, j) e^{-2\pi nk/N} \quad (3)$$

$$BP(i, j) = \frac{1}{f_2 - f_1} \sum_{k=f_1}^{f_2} X^k(i, j); BPR = \frac{BP}{BP_{ref}} \quad (4)$$

$$X_{BPR}(j) = BPR(C3, j) - BPR(C4, j) \quad (5)$$

MDC and NDBC. We implement two classifiers *Minimum Distance Classifier (MDC)* and *Normal Density Bayes Classifier (NDBC)*. MDC classifies a new data point by measuring the distance between the new data point and the prototypes ω . New data point X is classified as a class ω_i if

$$\exists_i \forall_j \|X - \mu_{\omega_i}\| < \|X - \mu_{\omega_j}\| \quad (6)$$

NDBC is more sophisticated than MDC, it takes into account the variance among the trained data points as well. NDBC classifies a new data point X as ω_i if

$$\exists_i \forall_j P(\omega_i | X) > P(\omega_j | X) \quad (7)$$

$$\text{where } P(\omega_i | X) = \frac{P(X | \omega_i) P(\omega_i)}{P(X)} \quad (8)$$

$$\text{and } P(X | \omega_i) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2} (X - \mu_{\omega_i})^T \Sigma^{-1} (X - \mu_{\omega_i})} \quad (9)$$

4 Result and Discussion

Figure 3 is produced using EEGLab [2]. They show the LMP from left-fingers movements, right-fingers movements and rest (no finger movement). It can be clearly observed that the LMP around 200 ms to 700 ms (after the onset time) exhibits a clear negativity of C3 and C4 contralaterally to the responding hand. Therefore X_{LMP} is taken from the time interval between 0 to 600 ms of channels C3 and C4.

Figure 4 shows the averaged band power ratio from left fingers and right fingers movements. The region between samples number approx. 100-300 (approx.

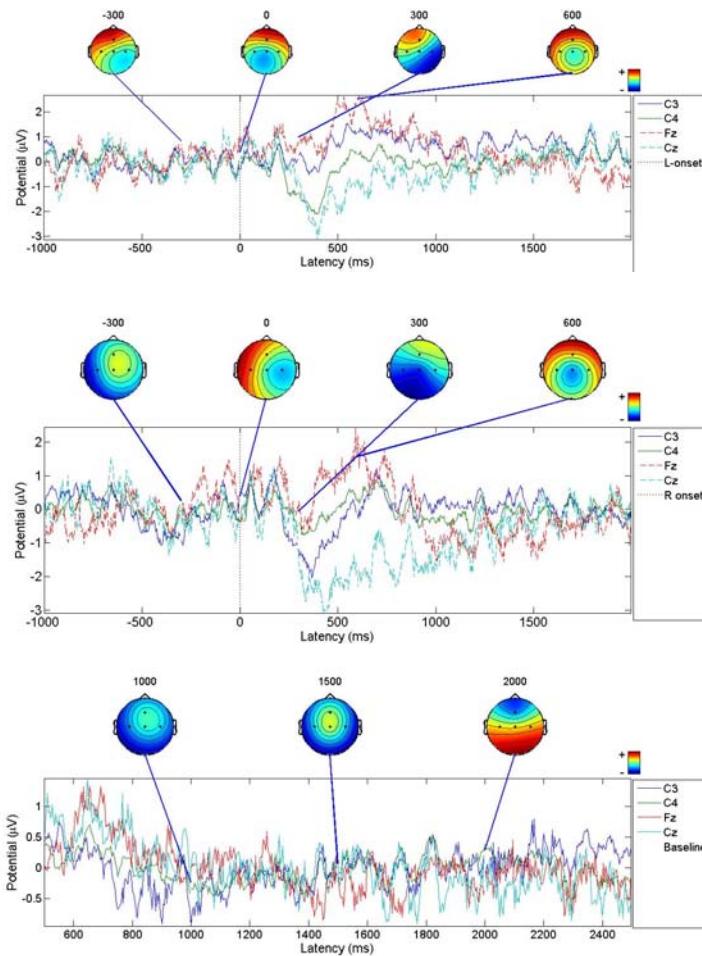
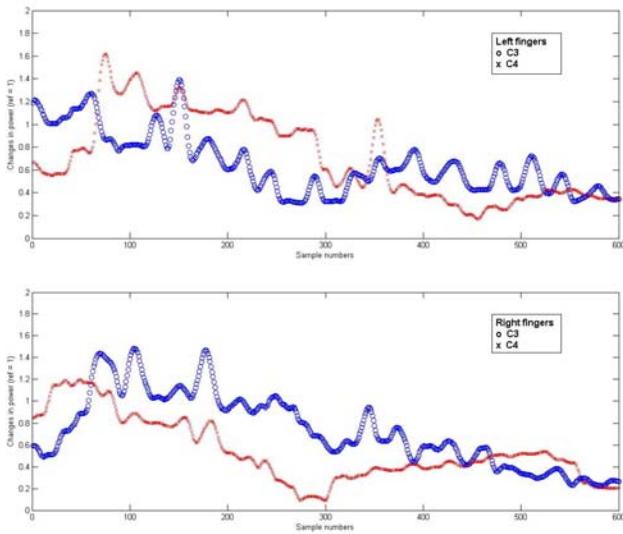


Fig. 3. ERP from brisk left fingers, right fingers movements and baseline

0-600 ms) shows changes in X_f/X_{ref} . Table II summarises the performances of our classifiers built using the features from ERP and BPR. For each run, 120 epochs are randomly partitioned into 10 equal partitions for 10 fold cross validation. The figures reported here are averaged over 10 runs of 10 fold cross validation (e.g., the score of 100% means the models correctly classify test data in all 1200 attempts). It is conclusive from the results that normal density Bayes classifier outperforms minimum distance classifier. This is not a surprise since NDBC also take into account the variance and covariance among components in the features. Minimum distance classifier, however, does not. MDC classifies new input merely on the Euclidean distance between the new input and the prototypes. Both classifiers are robust as their performance is consistent over smoothed (denoise using wavelet decomposition level 1 and 4) and raw signal.

**Fig. 4.** Band power ratio from left fingers and right fingers epochs**Table 1.** Performance of MDC and NDBC

ERP/BPR	MDC				NDBC				(TP+TN)/2	
	TP	TN	FP	FN	(TP+TN)/2	TP	TN	FP	FN	
X_{ERP}	61.8	63.7	38.2	36.3	62.8	100.0	100.0	0.0	0.0	100.0
σ^2	0.9	0.6	0.9	0.6		0	0	0	0	
$X_{ERPwaveletlevel1}$	61.0	64.2	39.0	35.7	62.6	100.0	100.0	0.0	0.0	100.0
σ^2	0.9	0.8	0.9	0.8		0	0	0	0	
$X_{ERPwaveletlevel4}$	60.9	64.5	39.1	35.5	62.7	100.0	100.0	0.0	0.0	100.0
σ^2	1.6	1.8	1.6	1.8		0	0	0	0	
X_{BPR}	60.1	60.1	39.9	39.9	60.1	94.8	78.6	5.2	21.4	86.7
σ^2	3.4	3.4	3.4	3.4		4.0	12.0	4.0	12.0	

5 Conclusion

In this report, we investigate event-related motor potential. We carefully study the EEG signal from voluntary cued motor actions of left-fingers and right-fingers movements. Two types of classifiers, minimum distance classifier and normal density Bayes classifier are employed. We conclude that *lateralised motor potential (LMP) observed within one second window after the onset of action*, is a robust feature for the imperative cued left-fingers and right-fingers movements tasks. The features derived from band power perform poorer than the features derived from ERP in both classifiers. This report shows evidence that an appropriate feature selection is the key to a good classification performance. That is, feature

choices are more crucial than classifier choices. We also suggest that standard classifiers such as normal density Bayes could perform well in this kind of task.

Acknowledgments. This material is based upon work supported by the Ministry of Science, Technology and Innovation of Malaysia (MOSTI) under Grant No. 01-02-01-SF0071.

References

1. Coyle, D., Prasad, G., McGinnity, T.M.: A time-frequency approach to feature extraction for a brain-computer interface with a comparative analysis of performance measures. *Applied Signal Processing* 19, 3141–3151 (2005)
2. Delorme, A., Makeig, S.: EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods* 134(1), 9–21 (2004)
3. Hjorth, B.: EEG analysis based on time domain properties. *Electroencephalography and Clinical Neurophysiology* 76, 306–310 (1970)
4. Goncharowa, I.I., Barlow, J.S.: Changes in EEG mean frequency and spectral purity during spontaneous alpha blocking. *Electroencephalography and Clinical Neurophysiology* 76, 197–204 (1990)
5. Pfurtscheller, G., Pregenzer, M., Flotzinger, D.: Classification of single EEG trials using machine learning methods and neural networks. In: Angeleri, F., Butler, S., Giaquinto, S., Majkowski, J. (eds.) *Analysis of the Electrical Activity of the Brain*. John Wiley & Sons, Chichester (1997)
6. Pfurtscheller, G., Neuper, C., Birbaumer, N.: Human brain-computer interface. In: Riehle, A., Vaadia, E. (eds.) *Motor Cortex in Voluntary Movements: A Distributed System for Distributed Functions*, ch. 14, pp. 367–401. CRC Press, Boca Raton (2005)
7. Zani, A., Proverbio, A.M.: *The cognitive electrophysiology of mind and brain*. Academic Press, London (2003)

A Neural Network Based Hierarchical Motor Schema of a Multi-finger Hand and Its Motion Diversity

Eiichi Inohira¹, Shiori Uota², and Hirokazu Yokoi¹

¹ Kyushu Institute of Technology, Hibikino 2-4, 808-0196 Kitakyushu, Japan

{inohira,yokoi}@life.kyutech.ac.jp

² Mitsubishi Electric Corporation, Japan

Abstract. This paper presents a neural network based hierarchical motor schema of a multi finger hand to generate suitable behavior for an unknown situation without retraining all neural networks and investigates its motion diversity by changing its input signals. Conventional neural networks are hard to generate desired movements in an unknown situation. Our hierarchical motor schema consists of the two layers. A lower schema is implemented by a recurrent neural network trained with primitive movement patterns and generates a finger movement from a command code sent from the upper schema. The upper schema generates command codes to each finger from a behavior command code such as grasping. We showed that though the lower schemata were fixed, diversity of generated finger movements can be obtained by changing a behavior code of the upper schema through computer simulation.

1 Introduction

Motion generation of a multi-finger hand by a mathematical model is a time-consuming task due to its computational complexity [1]. A neural network can generate finger movement patterns immediately. In [2], neuro-fuzzy networks are used to generate hand configuration to grasp an object. Moreover, desired patterns can be generated in a similar situation due to generalization capability of a trained neural network. However, generalization capability of conventional multi-layered neural networks is not so high that desired patterns can be generated in a different situation. On the other hand, it is considered that muscle movement patterns are generated through hierarchical pattern generators in a human brain. We focused on Schema theory [3] and proposed hierarchical motor schema which can generate similar patterns in continuity according to its input values [4]. We aimed at realizing diversity of generated patterns without retraining the whole motor schema.

The goal of this paper is to evaluate generalization capability of the our proposed hierarchical motor schema with measured finger movements through computer simulation. And also, we investigate diversity of generated patterns of the hierarchical motor schema by changing its input signals.

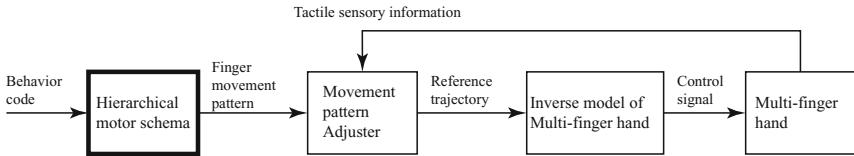


Fig. 1. An overview of the proposed motion control system of a multi-finger hand

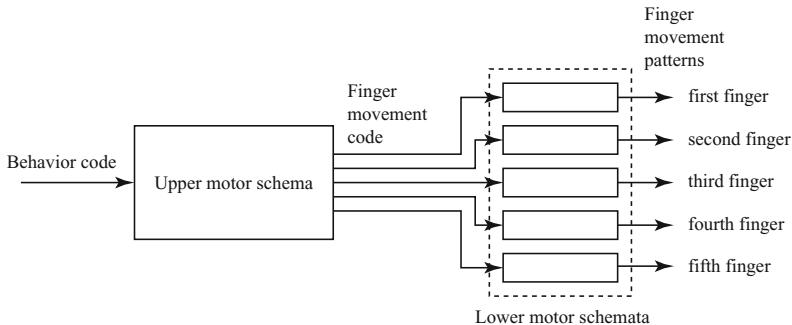


Fig. 2. A hierarchical motor schema

2 A Neural Network Based Hierarchical Motor Schema of a Multi-finger Hand

2.1 Our Proposed Hierarchical Motor Schema

Our proposed motion control system of a multi-finger hand [4] is shown in Fig. 1. In this paper, we focus on only the hierarchical motor schema. The hierarchical motor schema generates finger movement patterns from a behavior code. A behavior code represents a specific behavior such as grasping. Here, we assume that a behavior code is given from a external system. Generated finger movement patterns are sent to the motion pattern adjuster, which adjusts the input pattern to suppress awkward patterns to fail a task by using tactile sensory information of a multi-finger hand. Modified finger movement patterns are sent to the inverse model of a multi-finger hand to obtain a control signal.

The hierarchical motor schema consists of the two layers as shown in Fig. 2. The upper motor schema generates a finger movement code from a behavior code. Each of the lower motor schemata corresponds to each of the five fingers. A lower motor schema generates finger movement patterns from a finger movement code.

In the hierarchical motor schema, motor schemata are implemented by recurrent neural networks. We used a recurrent Volterra neuron network [5], which has high learning capability for dynamical systems. A Volterra neuron is a neuron

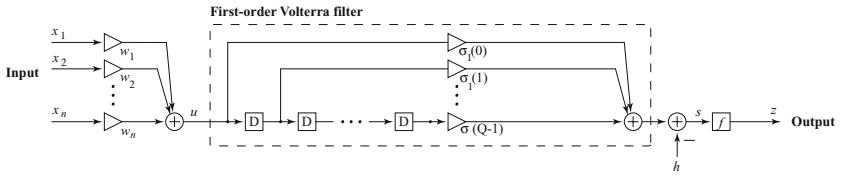


Fig. 3. A Volterra neuron

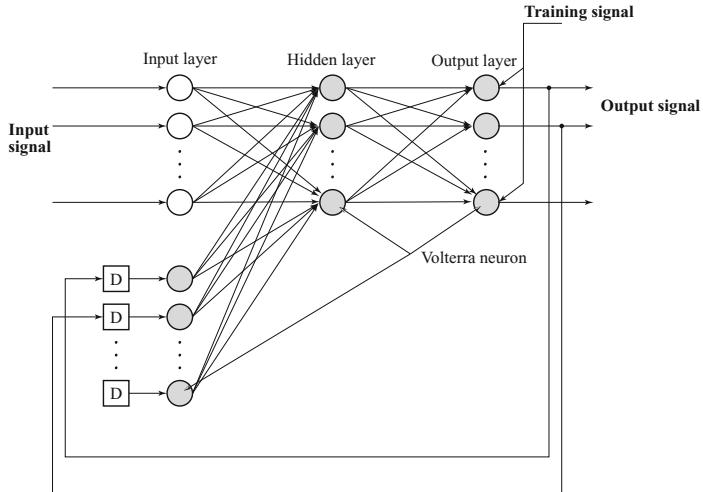


Fig. 4. A recurrent Volterra neuron network used in the lower motor schemata

which has an embedded Volterra filter as shown in Fig. 3. A first order Volterra filter is used in this paper and then equivalent to a FIR or IIR filter. A recurrent Volterra neuron network (RVNN) is a recurrent neural network composed of a Volterra neuron as shown in Fig. 4. A lower motor schema is implemented by a RVNN. The upper motor schema is also implemented by a RVNN. As shown in Fig. 5, the context layer of a RVNN in the upper motor schema is connected to output ports of the local schemata to give feedback of finger movement patterns to the upper motor schema.

The lower motor schemata learn a relationship between primitive finger movement patterns and their corresponding codes. The upper motor schema learns a relationship between behavior code and their corresponding finger movement codes. The upper motor schema is trained through the fixed lower motor schemata. The fixed lower motor schemata are trained in advance. The training signals of the upper neural network is not finger movement codes but finger movements, i.e., outputs of the lower motor schema. In training of the neural network in the upper motor schema, the lower motor schemata are invariable and only transfer signals of error back-propagation.

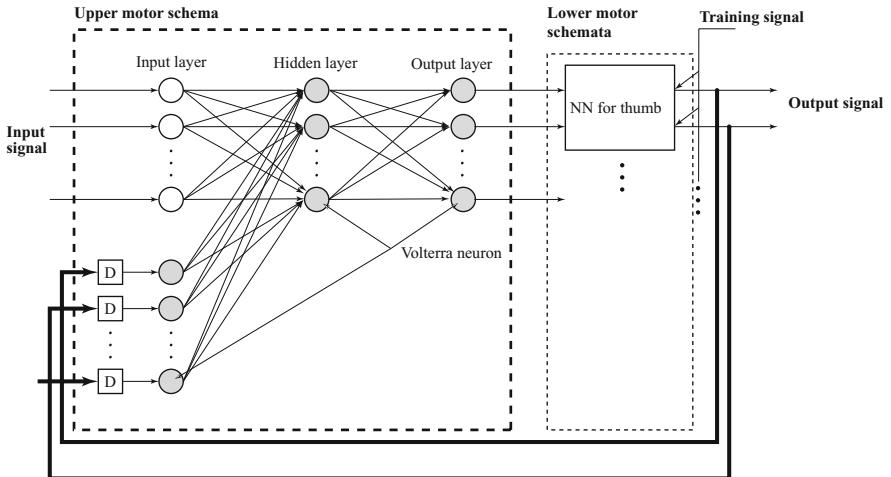


Fig. 5. A recurrent Volterra neuron network used in the upper motor schema (Note that feedback to this RVNN is sent from outputs of the lower motor schema)

3 Numerical Experiments

3.1 Setup

We used finger movement patterns acquired by the data glove. The finger movement patterns consists of the ten joint angles of the fingers. We measured the following movements as primitive movement patterns.

1. flexing and stretching each finger
2. grasping and releasing with all fingers

The first pair of movements is used for training of the lower motor schemata. We prepared the patterns with the three stage of speed such as a slow or medium or fast movement. Then the primitive movement patterns of the lower motor schemata amounts the six patterns. The second pair of movements is used for training of the upper motor schema. We prepared the patterns with the two different grasping objects. Then the primitive movement patterns of the upper motor schema amounts the four patterns. The sampling time is 13.7 millisecond in these movement patterns.

As shown in Table 1, A finger movement code consists of the five positive numbers. The fifth number indicates speed of a finger movement such as 0.8 for a slow movement, 1.0 a medium movement, and 1.2 a fast movement. The finger movement codes are defined to become short hamming distance between similar movements. Design parameters of neural networks in the lower motor schemata are shown in Table 2. Initial values of connection weights, coefficients of Volterra filters, and thresholds of the neural networks are given by using uniform random numbers with the range shown in Table 3. The neural networks are trained until

Table 1. Finger movement codes

Movement	Speed	Finger movement code
Flexing	Fast	[1 0 1 0 1.2]
Flexing	Medium	[1 0 1 0 1.0]
Flexing	Slow	[1 0 1 0 0.8]
Stretching	Fast	[0 1 0 1 1.2]
Stretching	Medium	[0 1 0 1 1.0]
Stretching	Slow	[0 1 0 1 0.8]

Table 2. Design parameters of RVNN in the lower motor schemata

Layer	Size	Filter length of a Volterra filter
Input layer	5	-
Hidden Layer	50	50
Output Layer	2	50
Context Layer	2	50

Table 3. Ranges of initial values of RVNN in the lower motor schemata

Connection weights	Volterra filter coefficients	Thresholds
-0.05 ~ 0.05	-0.05 ~ 0.05	-0.1 ~ 0.1

Table 4. Behavior codes

Behavior	Behavior code
Grasping an object 1	[1 0 1 1 1 1]
Grasping an object 2	[0 1 0 0 0 1]
Releasing an object 1	[1 0 1 1 0 1]
Releasing an object 2	[0 1 0 0 1 1]

Table 5. Design parameters of RVNN in the lower motor schemata

Layer	Size	Filter length of a Volterra filter
Input layer	6	-
Hidden Layer	50	50
Output Layer	2	50
Context Layer	10	50

10,000 learning cycles. One learning cycle is defined as 120 times of learning of a movements pattern, i.e. 20 times of repetitive learning of a movement pattern are sequentially performed for the six movement patterns.

A behavior code consists of the six positive numbers as shown in Table 4. Here, we view the two kinds of grasping movements with different objects as the same behavior. The releasing movements are handled in the same manner. Design parameters of the neural network in the upper motor schema are shown in Table 5. Initial values of connection weights, coefficients of Volterra filters, and thresholds of the neural network are given by using uniform random numbers with the range shown in Table 3.

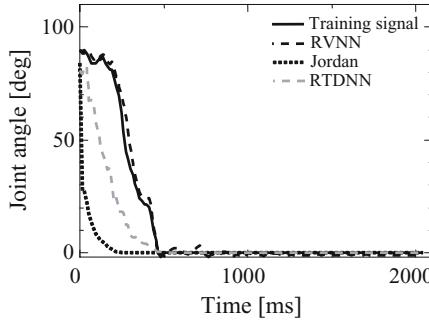


Fig. 6. A comparison between the three types of recurrent neural networks

In order to evaluate performance of a RVNN, we compared the three types of neural networks: RVNN, Jordan network[6] and RTDNN. A Jordan network is a widely used recurrent neural network and has feedback from an output layer to a hidden layer. A TDNN is also a well-known neural network for time series signal processing. We used RTDNN, which is a TDNN[7] based recurrent network. This is why TDNN has no feedback differently from RVNN and Jordan network.

We evaluated the best results of each type of neural networks with optimum learning rate, which was obtained through preliminary experiments. The reason is that the learning rate have much effect on training error. We used mean of sum of squared errors, which indicates training error, as performance index.

We also investigated generated patterns when command codes near training command codes was given to the upper motor schema in order to evaluate motion diversity.

3.2 Experimental Results

We showed that RVNN was the best among the three types of neural networks as shown in Fig. 6. Training error in the Jordan network was approximately 20 times larger than RVNN. The reason is that input signal of the hierarchical motor schema is constant. Thus learning of a relationship between behavior codes and finger movement patterns is very difficult. Each neuron of a RVNN has delay elements and can store past information. On the other hand, Jordan network and RTDNN store past information in only a context layer. A RVNN can learn the relationship accurately due to past information in each neuron.

We showed that training of the lower motor schemata was successfully done as shown in Fig. 7. Next, we showed that training of the upper motor schemata was also successful as shown in Fig. 8. It means that the trained hierarchical motor schema can generate finger movement patterns from behavior codes by using RVNN.

Fig. 9 showed results when input signals slightly different from the training data was given to the trained hierarchical motor schema. For instance, a behavior code [0.9 0.1 0.9 0.9 0.9 0.9], which is near a code [1 0 1 1 1 1] used at training. It

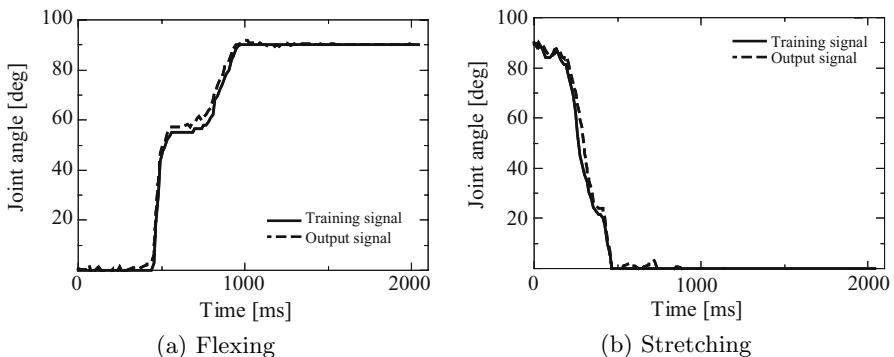


Fig. 7. Output signals of a trained lower motor schema (the first joint of the first finger)

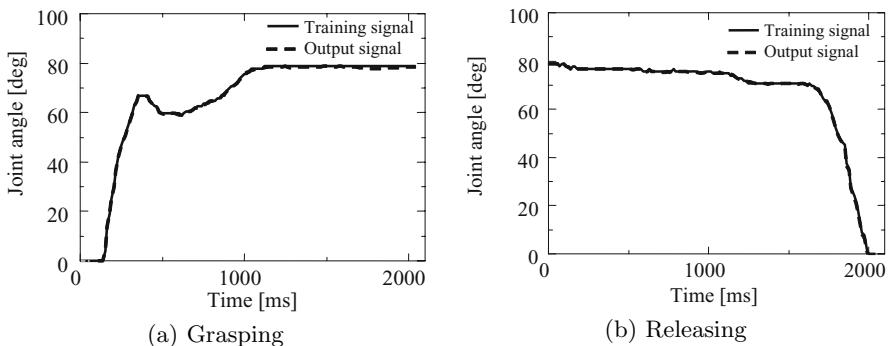


Fig. 8. Output signals of the trained hierarchical motor schema (the first joint of the second finger)

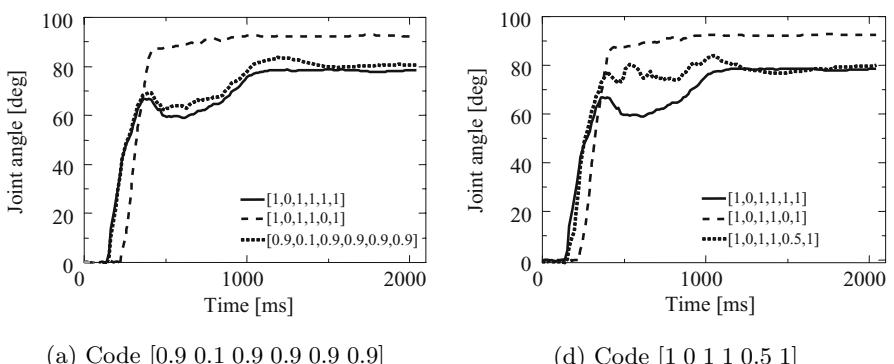


Fig. 9. Generated output signals when near behavior command codes are given

is clear that near behavior codes generate similar movements as shown in Fig. 9. The reason is that the trained neural networks has generalization capability. This result implies that a new type of behavior can be generated by only changing a behavior code without retraining the hierarchical motor schema.

4 Conclusions

Our proposed motion control system of a multi-finger hand was presented and evaluated with measured finger movement patters through computer simulation. The hierarchical motor schema learned a relationship between command codes to behavior and finger movements patterns at high accuracy. We showed that the hierarchical motor schema can generate similar movement patterns by slightly changing a behavior code. It means that hierarchical motor schema can generate various movement patterns.

References

1. Salimi, S., Bone, G.M.: Kinematic enveloping grasping method for robotic dexterous hands and three-dimensional objects. *Robotica* 26, 331–344 (2008)
2. Laschi, C., Asuni, G., Guglielmelli, E., Teti, G., Johansson, R., Konosu, H., Wasik, Z., Carrozza, M.C., Dario, P.: A bio-inspired predictive sensory-motor coordination scheme for robot reaching and preshaping. *Autonomous Robots* 25, 85–101 (2008)
3. Arbib, M.A.: *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge (1998)
4. Uota, S., Yokoi, H.: A Realization of motion diversity of the Robotic hand by the hierarchical motion schema, Technical Report of IEICE NC2003-75, pp. 25–28 (2003) (in Japanese)
5. Iwamoto, S., Yoshida, T., Yokoi, H.: Basic investigation associated with neural control of biped walking robot, Technical Report of IEICE 93, pp. 23–30 (1994) (in Japanese)
6. Jordan, M.I.: Attractor dynamics and parallelism in a connectionist sequential machine. In: Proc. of the English Annual Conference of the Cognitive Science Society, pp. 531–546 (1986)
7. Waibel, A.: Modular construction of time-delay neural networks for speech recognition. *Neural Computation* 1, 39–46 (1990)

Part III

Cognitive Neuroscience

Biological Plausibility of Spectral Domain Approach for Spatiotemporal Visual Saliency

Peng Bian and Liming Zhang

Department of Electronic Engineering, Fudan University
pengfish@gmail.com, lmzhang@fudan.edu.cn

Abstract. We provide a biological justification for the success of spectral domain models of visual attention and propose a refined spectral domain based spatiotemporal saliency map model including a more biologically plausible method for motion saliency generation. We base our approach on the idea of spectral whitening (SW), and show that this whitening process is an estimation of divisive normalization, a model of lateral surround inhibition. Experimental results reveal that SW is a better performer at predicitng eye fixation locations than other state-of-the-art spatial domain models for color images, achieving a 92% consistency with human behavior in urban environments. In addition, the model is simple and fast, capable of generating saliency maps in real-time.

1 Introduction

In human visual processing, there exists a bottom-up attention selection mechanism that can rapidly locate generally salient objects in a cluttered scene without any top-down guidance. Through continual breakthroughs in vision research, we are beginning to understand the principles of this mechanism, and spawning from the new found understanding are many computational models of visual attention which have been proposed over the past decade.

Itti et al. [1] proposed a biologically plausible model of bottom-up attention selection, but the model suffers from computational complexity, ad hoc design choices and over-parameterization. Recently, Bruce et al. [2] and others [3], [4] used more fundamental concepts such as self-information, Bayesian surprise and graph theory to provide novel approaches to modeling attention selection. While all these methods are different in their approach, it is evident that the goal is to provide normalization schemes for the decomposed features that promote locations of conspicuity or pop-out. This raises the question, is there another way to normalize without requiring a lengthy feature decomposition process? In this paper we show the normalization can be conducted in spectral domain, and is biologically plausible in that it is an approximation of divisive normalization [5] which accurately models the response due to lateral surround inhibition of simple cells [6].

The first spectral domain approach is named spectral residual (SR) due to its hypothesis that salience is the residual difference between the perceived spectrum and characteristic spectrum of natural images [7], but in its algorithm the phase component is kept separated and untouched. It was then discovered in [8] that simply taking the inverse Fourier transform of the phase spectrum alone and discarding the

amplitude produce equivalent results; calculation of the residual is not needed. This approach is named phase spectrum of Fourier transform (PFT), and was extended to color videos using quaternion Fourier transform (PQFT). In this paper we refine the model in [8] and provide biological justification. We also propose a more biologically plausible motion saliency generation capable of removing background ego-motion from localized, salient motion using the same unified spectral domain framework.

2 Biological Justification

It is recently believed that a bottom-up saliency map exists in the primary visual cortex (V1) where cellular response is directly correlated with saliency, with the location of the visual scene that corresponds with the most active cells the most likely candidate for the focus of attention [9]. Since orientation selectivity and lateral surround inhibition are well known properties exhibited by cells in V1, estimating the response due to these processes can provide us with an early stimulus driven saliency map, but such a process can be time consuming in spatial domain.

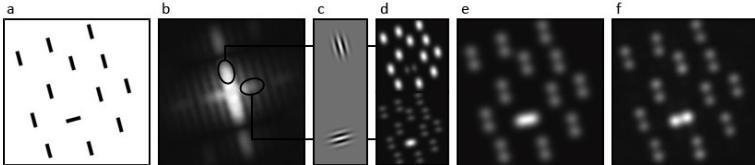


Fig. 1. (a) An intuitive psychophysical pattern of oriented bars. (b) The amplitude spectrum of (a). (c) Two oriented Gabor filters. (d) Response maps after convolution with corresponding Gabor filters in (c). (e) Recombined response map after orientation selectivity using a large bank of Gabor filters and lateral surround inhibition. (f) The spectral whitened response map.

Orientation selectivity is modeled by convolution with an oriented Gabor filter, which in spectral domain band-passes a region around the frequency and orientation the filter is tuned to. Fig. 1a is an intuitive pattern of oriented bars which we will use as example for the rest of the section, and its amplitude spectrum is given by Fig. 1b. If we convolve the image with the two Gabor filters given in Fig. 1c, then we effectively band-pass the two circled regions in Fig. 1b. The results of the convolution in spatial domain are given in Fig. 1d, where the total response in the off-vertical direction is larger than that in the off-horizontal direction. We can also see that the total amplitude for the band-passed regions is also larger for the off-vertical direction. In fact, there is an intimate relationship between the total response in spatial domain and spectral domain, which can be seen through Parseval's theorem. The total response D_a for one such Gabor convoluted feature component a is defined by:

$$D_a = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \|r_a(x, y)\|^2 = \frac{1}{NM} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \|R_a(u, v)\|^2 \quad a = 1, \dots, K \quad (1)$$

where R_a is the Fourier transform of response r_a for the feature component a of the K total feature components. N and M are the length and width of the image, respectively.

We can say that given an image as stimulus, the local amplitude around a particular frequency and orientation in spectral domain represents the total response of cells tuned to the same frequency and orientation.

Lateral surround inhibition suppresses responses of similarly tuned cells depending on the total response of those similarly tuned cells for a given image, and has been modeled using divisive normalization [5]. In its simplest form, the response for each feature component is divided by the total response of that feature component:

$$\|r'_a(x, y)\|^2 = \frac{1}{D_a} \|r_a(x, y)\|^2 \quad a = 1, \dots, K \quad (2)$$

A high total response (top of Fig. 1d) will result in greater suppression than a low total response (bot. of Fig. 1d) to the extent that the total response after lateral surround inhibition for each feature component is equal:

$$\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \|r'_a(x, y)\|^2 = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \|r'_b(x, y)\|^2 = 1 \quad a = 1, \dots, K \quad b = 1, \dots, K \quad (3)$$

Since we have shown previously that the total response in spatial and spectral domain is proportional, this means the divisive normalized band-pass regions in spectral domain have the same total amplitude. Given a complete bank of filters like the one found in V1, the amplitude spectrum is roughly flat after lateral surround suppression. The inverse Fourier transform of a flattened amplitude spectrum, which we define as spectral whitening (SW), will therefore approximate the recombined spatial response after orientation selectivity and lateral surround inhibition. This is illustrated by the similarity between Fig. 1e and Fig. 1f.

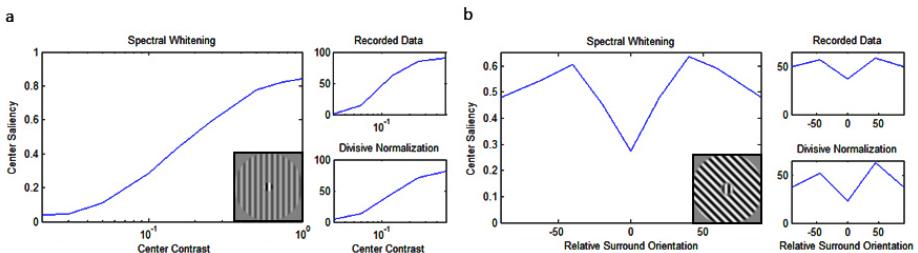


Fig. 2. (a) Center saliency as a function of center contrast with fixed surround contrast. (b) Center saliency as a function of center-surround orientation difference. *Recorded data* and *divisive normalization* plots are given in [5].

To further demonstrate the validity of the spectral whitening approach, we compare the electrophysiological recordings of macaque simple cells from [6] and the response curve modeled by divisive normalization [5] to the response curve modeled by SW. For each of the two types of sinusoidal grating patterns used, the fundamental quantity (center contrast or relative surround orientation) is varied and the response for the cell with its receptive field in the center is recorded. The resulting plots given in Fig. 2 show the similarity between the recorded and modeled curves.

3 The SW Model

Given an image I of any size, we first resize the image to a fixed scale under the assumption that the scale to be viewed is equal for all images. The ratio of the image is retained while the maximum length of the image is set to 64px, which is the heuristically optimized size used in [7], [8]. The saliency map of the resized image is then calculated as follows:

$$\begin{cases} f(u, v) = F[w(I(x, y))] \\ n(u, v) = f(u, v) / \|f(u, v)\| \\ S(x, y) = g(u, v) * \|F^{-1}[n(u, v)]\|^2 \end{cases} \quad (4)$$

F and F^{-1} denote Fourier and inverse Fourier transforms respectively and w is a windowing function. The flattened spectral response n is transformed into spatial domain and squared to further promote the salient regions. A Gaussian filter g is then convoluted with the resultant to model the spatial pooling operation of complex cells. This algorithm is much simpler than the proposed SR algorithm in [7] and further simplifies the PFT algorithm proposed in [8], which requires separation of the phase from f and multiplication with an exponential before inverse transform.

3.1 Vector Approach to Color

One approach for color image saliency map generation is to separate the image into three channels, and then compute the saliency map for each channel and combining them to form the final saliency map. The problem with this approach is that it assumes independence of each color channel, but they are in fact correlated. For example, if we separate the color image in Fig. 3a into three biologically plausible channels (intensity, red-green and blue-yellow) as proposed by [1] and compute its final saliency map using the approach stated above, then the green bar is not more salient than the red bars, as illustrated by Fig. 3b; however, if we were to use the normal RGB color channels to compute its final saliency map, then the green bar would be more salient. By assuming independence when the color space is very much correlated, the result will differ depending on how the color channels are separated.

By using a vector transform such as the quaternion Fourier transform, we can locate the salient color because such a method is not reliant on how we separate the color channels; each color is treated as a vector, and the color space is considered as a whole. While [8] did incorporate a quaternion Fourier transform in its PQFT algorithm, there was no explanation of motivation and no comparison made to separate computation of each scalar color channel. Here we show the advantage of using the vector approach.

For the vector approach, a quaternion Fourier transform is computed in place of the real-valued Fourier transform. We conduct a simple YUV color space transformation with each channel represented as a quaternion axis such that:

$$q(x, y) = w_y y(x, y)i + w_u u(x, y)j + w_v v(x, y)k \quad (5)$$

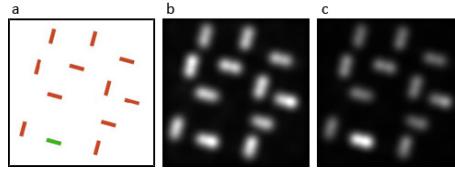


Fig. 3. (a) Intuitive color pattern. (b) Saliency map generated by computing each color channel separately. (c) Saliency map generated by considering the image as a quaternion.

where q is the quaternion representation which replaces I in procedure, y, u, v denote the luminance channel and the two chrominance channels respectively, and w_y, w_u, w_v are the weights for their respective color channels to compensate for the disparity in perceived salience due to biological dynamic range differences. Generally, giving the chrominance channels a higher weight gives better results, which is why we conducted a color space transformation to separate luminance from chrominance. The exact implementation for computing the quaternion Fourier transform and its inverse can be found in [10]. Fig. 3c shows the resulting saliency map from using the vector approach as described above, and the green bar is clearly more salient.

3.2 Temporal Extension

Dynamic saliency detection requires the ability to separate background motion from localized, salient motion, a process conducted in the MT area of our visual cortex [11]. This can be achieved by utilizing phase correlation [12], which provides the motion vector for panning movement by computing a normalized cross-correlation of two frames through a Fourier transformation, which is concurrent with our spectral domain framework. The equation to calculate the phase correlation plane is:

$$(\Delta x, \Delta y) = \arg \max_{(x, y)} \left\{ \mathcal{F}^{-1} \left[f_c(u, v) f_p^*(u, v) / \|f_c(u, v) f_p^*(u, v)\| \right] \right\} \quad (6)$$

where f_c and f_p are Fourier transforms of the two intensity frames. The two frames are then shifted by the global motion, given by Δx and Δy , to compensate for the panning motion followed by a difference frame calculation of the shifted frames, which is denoted as the motion component. We then extend the quaternion definition to include this new motion component replacing the simple difference of two frames motion component used by QPFT in [8]:

$$q(x, y) = w_m m(x, y) + w_y y(x, y) i + w_u u(x, y) j + w_v v(x, y) k \quad (7)$$

where m denotes the motion component calculated from the phase correlation compensated difference frame and w_m is the weight for the motion component. A qualitative comparison of our spatiotemporal saliency map approach for frames with camera shake and camera panning is illustrated by Fig. 4.

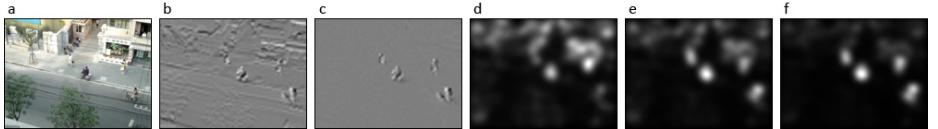


Fig. 4. (a) Motion frame of moving pedestrians. (b) Difference frame showing visible camera shake. (c) Phase correlation compensated difference frame. (d) Static saliency map of (a). (e) Dynamic saliency map of (a) and (b) as in [8]. (f) Dynamic saliency map of (a) and (c).

4 Experimental Validation

Since [13] proposed using a receiver operating characteristic (ROC) curve to measure the consistency of a visual attention model with human eye fixation, numerous papers [2], [4], [14] have used this method to gauge the performance of their model. Given the database from [2] containing 120 colored natural images of urban environments and eye fixation data from 20 subjects, we use the static saliency map generated by SW on the 120 natural images, obtain the ROC curve by using the fixation data as ground truth, and use the area under the curve (AUC) as consistency measure. The AUC is calculated for all fixations as well as the first 2 fixations as proposed in [14] which are more likely to be driven by bottom-up attention mechanisms. We also calculate the ratio between model AUC and the inter-subject AUC as proposed in [2] in order to compare consistency of our model with human consistency between subjects. We repeat the process for saliency maps generated by other state-of-the-art spatial domain models in [2], [14] and Walther's Saliency Toolbox (STB) from [15]. The results given in Table 1 show that SW outperforms all other models in all criteria, indicating better overall consistency.

Table 1. Results from the eye fixation prediction experiment

Model	SW	Bruce et al. [2]	Gao et al. [14]	STB [15]
AUC all fixations	0.8010	0.7816	0.7683	0.7341
AUC first 2 fixations	0.8252	0.7993	0.7981	0.7553
Model-human consistency	0.9227	0.8938	0.8924	0.8446

We also conduct a qualitative comparison of all the models for select images of moderate difficulty and an eye fixation density map as ground truth, which is illustrated by Fig. 5. Immediately we can notice a resemblance between our model and the model by Bruce et al. [2]. This is because our model utilizes similar concepts as the model in [2], such as solely using a large bank of ICA patches for feature decomposition. In addition, the self-information criteria in [2] give higher saliency to feature components with a low probability of occurrence, which as a result will have a low total response. It should be noted the results from the model in [2] are obtained by using 31×31 ICA patches, which take up to a minute to generate a single saliency map. Our method generates a single static saliency map in around 1/20 of a second in MATLAB.

SW is able to find small salient objects (third and fourth row) even though it uses a 64px wide resized image to calculate its saliency map. Furthermore, by using a vector approach to color and putting more weight in the chrominance channels, SW is able to find color pop-out (second to last row) that the other models cannot.

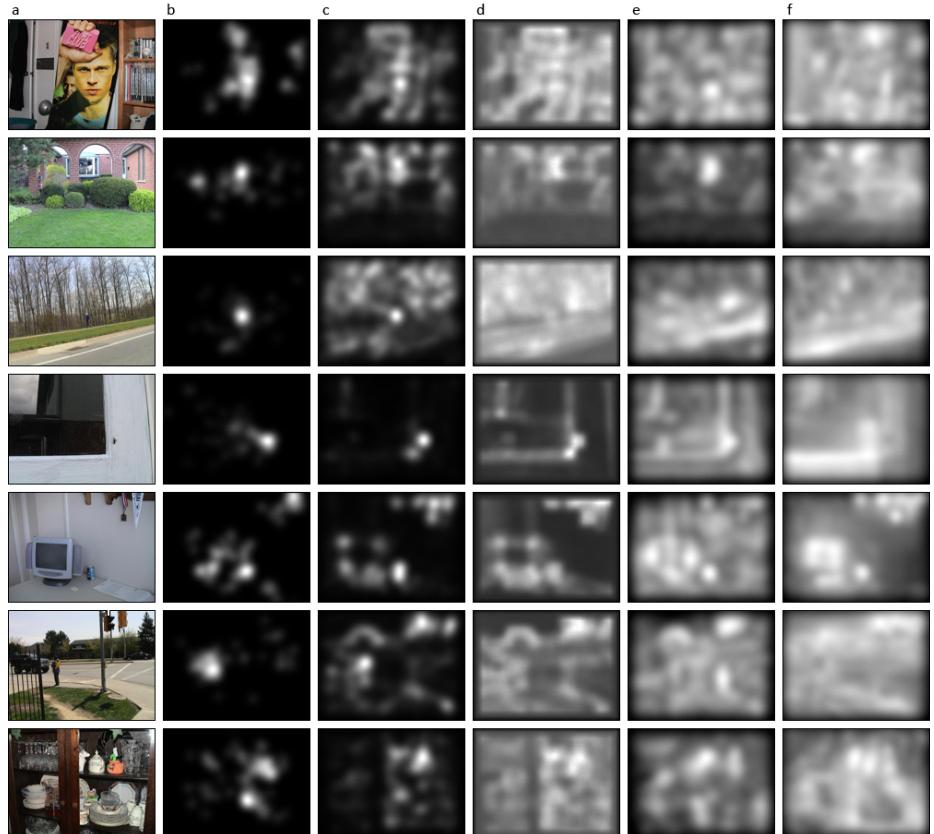


Fig. 5. (a) Images from dataset of [2]. (b) Eye fixation density maps. (c) Saliency maps generated by SW. (d) Saliency maps generated by Bruce et al. [2]. (e) Saliency maps generated by Gao et al. [14]. (f) Saliency maps generated using the STB [15].

5 Conclusion

We proposed a refined spatiotemporal model of visual saliency as well as provided biological justification for its ability to estimate bottom-up visual attention selection. The model shows good consistency with motion psychophysics, and is a better performer at predicting human eye fixations than other state-of-the-art models. Above all else, the model is simple, fast, and requires little parameter tuning. Furthermore, since digital signal processor chips capable of performing FFT are already widely available, our model can be easily implemented with existing hardware.

Acknowledgements

The authors thank Neil Bruce for kindly sharing the eye fixation data and saliency maps from [2], and Dashan Gao for saliency maps from [14]. This work is supported by the National Science Foundation of China (NSF60571052) and by Key Subject Foundation of Shanghai (B112).

References

1. Itti, L., Koch, C., Niebur, E.: A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Trans. PAMI* 20, 1254–1259 (1998)
2. Bruce, N.D., Tsotsos, J.K.: Saliency Based on Information Maximization. In: Proc. NIPS (2005)
3. Itti, L., Baldi, P.: Bayesian Surprise Attracts Human Attention. In: Proc. NIPS (2005)
4. Harel, J., Koch, C., Perona, P.: Graph-Based Visual Saliency. In: Proc. NIPS (2006)
5. Simoncelli, E.P., Schwartz, O.: Modeling Surround Suppression in V1 Neurons with a Statistically-Derived Normalization Model. In: Proc. NIPS (1998)
6. Cavanaugh, J.R., Bair, W., Movshon, J.A.: Orientation-Selective Setting of Contrast Gain by the Surrounds of Macaque Striate Cortex Neurons. *Neuroscience Abstracts* 23, 227.2 (1997)
7. Hou, X., Zhang, L.: Saliency Detection: A Spectral Residual Approach. In: Proc. CVPR (2007)
8. Guo, C.L., Ma, Q., Zhang, L.M.: Spatio-Temporal Saliency Detection Using Phase Spectrum of Quaternion Fourier Transform. In: Proc. CVPR (2008)
9. Li, Z., Dayan, P.: Pre-Attentive Visual Selection. *Neural Networks* 19, 143–1439 (2006)
10. Ell, T.A., Sangwine, S.J.: Hypercomplex Fourier Transforms of Color Images. *IEEE Trans. Image Processing* 16, 22–35 (2007)
11. Cao, A., Schiller, P.H.: Neural Responses to Relative Speed in the Primary Visual Cortex of Rhesus Monkey. *Visual Neuroscience* 20, 77–84 (2003)
12. De Castro, E., Morandi, C.: Registration of Translated and Rotated Images Using Finite Fourier Transforms. *IEEE Trans. PAMI* 9, 700–703 (1987)
13. Tatler, B.W., Baddeley, R.J., Gilchrist, I.D.: Visual Correlates of Fixation Selection: Effects of Scale and Time. *Vision Research* 45, 643–659 (2005)
14. Gao, D., Mahadevan, V., Vasconcelos, N.: The Discriminant Center-Surround Hypothesis for Bottom-Up Saliency. In: Proc. NIPS (2007)
15. Walther, D., Koch, C.: Modeling attention to salient proto-objects. *Neural Networks* 19, 139–1407 (2006)

A “Global Closure” Effect in Contour Integration

Kazuhiro Sakamoto, Hidekazu Nakajima, Takeshi Suzuki, 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. Evaluation of global closure of contour is important in object perception in natural scenes including many occlusions. Here, we conducted psychophysical experiments to test the closure effect in contour integration within a single spatial frequency range using Gabor patches. We found that closed arrangements of the patches embedded in noise patches were more salient than open arrangements. This effect was seen even in the experiment using stimuli that isolated global closure effect excluding effect of local orientational continuity among adjacent patches. These findings suggest that processing for global closure of contour exists separately from that for local orientational continuity in the visual system.

1 Introduction

Natural scenes include numbers of objects occluding each other. However, we can perceive an occluded object easily, by making use of its incomplete visual cues such as segments of the contour. The orientation change along the contour or the curvature is often discontinuous. In fact, we can perceive an occluded object by evaluating the global pattern of visible portion of the contour enclosing the object. Thus, evaluation of global closure of the contour segments is an important aspect in contour integration for object perception in natural scenes.

For more than a decade, “Gabor patches” have been used in psychophysical experiments on contour integration [1][2]. Gabor patches are blobs composed of a 1-D sinusoid multiplied by a 2-D Gaussian. They have advantages in investigating contour detection through interactions within one particular spatial frequency channel by excluding the influence from interactions between the different channels.

By utilizing Gabor patches, closure effects in contour integration were tested in some studies [2][3][4][5]. However, because, in typical images in their experiments, the target patches and their orientations were arranged along a circle or a closed continuous curve, the observed closure effects can be basically explained by “association field” theory or a sort of local interactions between adjacent patches. That is, the saliency of a Gabor patch is enhanced when another patch is placed in the association field and their orientations are collinear or tangential

to each other. As the number of these patches increases, the saliency is more enhanced. Thus, the circularly-arranged Gabor patches can be quite salient because the local interactions circulate around the circle as if these patches belonged to a patch-set of infinite number.

In our study, we used two kinds of Gabor-patch alignments that were closed but included some discontinuities in the orientation with neighboring patches. By using these stimuli, we attempted to separate “global closure” effects from “local” enhancing ones in detecting curved lines of Gabor patches embedded in noise patches.

2 Method

2.1 Apparatus

Stimuli were presented on a Nanao FlexScan 88F monitor in the 1280×1024 pixels mode. The size of a single stimulus was $17^\circ \times 17^\circ$. The mean luminance was 30 cd/m^2 . The Stimulus-presentation was controlled by a Packard-Bell computer with a Pentium 133 MHz processor, and the frame rate was 75 Hz. All experiments were conducted in a dark room.

2.2 Subjects

Five male subjects aged 24 to 28 including the two authors were used. All of the subjects had normal vision and were aware of the aim of the experiment, but they did not know the details of the experiments and the stimuli. The heads of the subjects were fixed on a custom-made device, and the viewing distance from the display to the subject was 50cm. They viewed the stimuli binocularly.

2.3 Stimuli

As mentioned above, one stimulus composed of Gabor patches whose luminance was modulated as

$$G(x, y, \Theta) = A \exp[-(x^2 + y^2)/2\sigma^2] \sin[2\pi/\lambda(x \cos \Theta + y \sin \Theta)] \quad (1)$$

where A is 24% of the mean luminance of the screen, λ is 0.13 degrees and σ is $\lambda/2$. Thus, the range of luminance was from 22.8 cd/m^2 to 37.2 cd/m^2 , and it was linearly modulated. The quantities (x, y) and Θ represent the position of the center of the patch on the screen and the orientation of the patch, respectively.

The alignment of the Gabor patches composing a target was determined in the same manner as employed by Field and his colleagues [1](Fig. 1A). First, a certain number of apparent line segments with distance D were connected to make a target. These segments were not displayed on the screen. The angular difference between neighboring line segments was $\pm\alpha$. The angular difference between a Gabor patch orientation and a line segment is represented by $\pm\beta$. In each experiment, the subjects were asked which screen contained an impressive

figure. There were two kinds of targets, closed alignments and open ones. One of the two alignments was decided randomly to be included in the stimulus displayed during one of the two stimulus presentation periods in each experiment. The center of the alignment was located within 0.5deg of the fixation point so that there was no eccentricity difference between a Gabor patch composing an open target and the corresponding patch belonging to a closed target with the same distance and angular conditions. That is to say, the two kinds of target differed only in open or close. The orientation of the target with respect to the fixation point was decided at random.

The placement of background elements also followed the method of [1]. The screen was latticed with pitch d . Each lattice element could be occupied by only one Gabor patch. That is, after Gabor patches composing the target were assigned to lattice elements, each remaining element was given a background patch whose orientation and location within the lattice was random. The mean distance between patches was set to 3λ . Each stimulus contained about 2,500 Gabor patches (Fig. 1B).

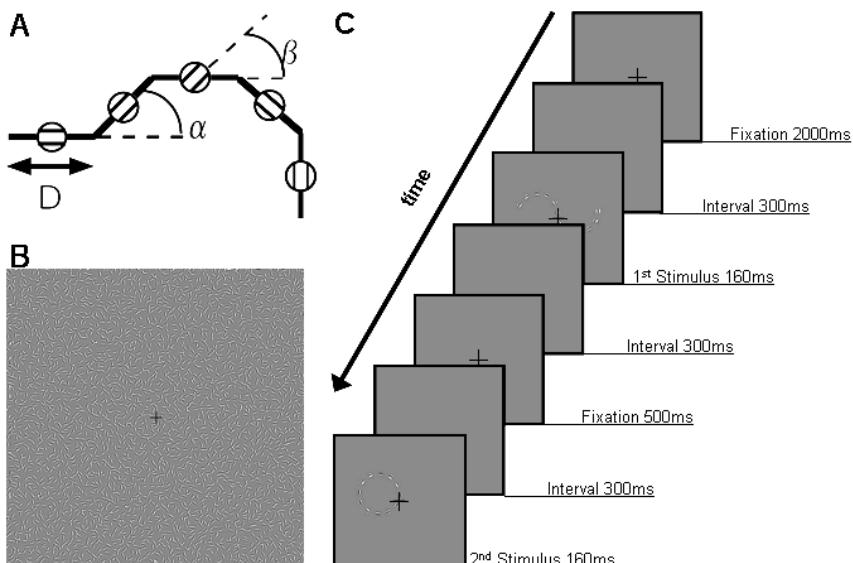


Fig. 1. Frameworks of the stimuli and the task. (A) A schematic diagram depicting the method of construction of a target used in all experiments. D: the length of apparent line segments connected to make a target figure. α : angular differences between neighboring line segments. β : angular differences between line segments and the orientations of the patches that are on the segments. (B) An example of the actual stimulus displayed for 160 msec. including a target. The stimulus consists of 2,500 Gabor patches. The size of the displayed stimulus was $17^\circ \times 17^\circ$. (C) Temporal sequence of the task. One of the two stimuli includes a target of an open or closed form.

2.4 Procedure

A temporal two-alternative forced-choice (2AFC) method was used in all experiments (Fig. 1C). After a fixation cross was presented for two seconds, the first stimulus was presented for 160msec. A 1.1 sec. inter-stimulus-interval (ISI) was followed by the second stimulus with a 160msec. duration. During the ISI period, the fixation cross was displayed for 500msec. The durations of the first and second stimulus presentation periods were set to be shorter than the latency of the saccade. One of the two stimuli included the target, and the subjects had to determine which screen contained the meaningful figure, and their determination was indicated by pressing a key. They had to maintain fixation during one trial, although the fixation cross disappeared during the first and the second stimulus presentation periods.

Each experiment used 10 kinds of targets, and each target was presented 100 times. Thus one experiment included 1000 trials. In order to help the subjects concentrate throughout the experiments, each experiment was divided into five sessions that had 220 trials, and each session was conducted on a different day. The first 20 trials in each session were used for training and darkness adaptation and were excluded from the analysis. The performance of each subject was tested using a binomial test.

3 Results

3.1 Experiment 1

Experiment 1 was conducted in order to verify the result of Kovacs and Julesz [2]. The alignment of the target was a continuous line (Fig. 2A). This implies that each angular difference between neighboring patches α was $\pm 30^\circ$. This is in the range of the association field of Field et al [1]. We prepared targets with five values of D: 3λ , 3.5λ , 4λ , 4.5λ and 5λ . Thus there were ten kinds of targets in Experiment 1, and each target contained 12 Gabor patches.

Our experiment was controlled more strictly than that in [2], in the sense that the eccentricity of each Gabor patch composing a closed target was equal to that of the corresponding patch for the open target with the same distance D. For this reason, a difference in the saliency between the two types of targets cannot be attributed to an eccentricity dependence of the detection threshold of the patches. Such a difference also cannot be caused by focal attention on a particular site in the visual field, since the direction of the target with respect to the fixation point was randomly decided. Also, the local features of the target cannot be used as a cue for discrimination of the two types.

The results exhibited in Fig. 2B was consistent with the results of [2]. The saliency of closed targets was significantly higher than that of open arrangements for all distances D. Hence, the effect of “closure” can be seen under the condition that local orientational continuities are preserved.

3.2 Experiment 2

The rule for patch alignment in the target used in Experiment 2 was quite similar to that in Experiment 1. However, in Experiment 2, each closed target had a “corner” of 100° , that is, an orientational discontinuity, so that the saliency of the closed target cannot be explained by local enhancement in an “infinite” length patch alignment. Namely, the saliency of a closed target should be the same as that of the corresponding open target according to the local mechanism or “association field” with the line length effect. Each target was composed of 13 patches, and α was $\pm 30^\circ$, 10° , or 0° degree, as shown in Fig. 2C.

The results are displayed in Fig. 2D. The saliency of closed targets in Experiment 2 was weaker than that in Experiment 1, but it was still significantly stronger than that of open targets in the case of large D ($D = 4.5$ degree). Although the fact that the open targets were also significantly more salient than would be the case if determinations made by the subjects could be made only by random choice suggests that the association field mechanism is at work, the significant difference between the saliency for the two types of targets cannot be attributed to local mechanisms like the association field. It must be caused by some global effects of the target shapes.

3.3 Experiment 3

In Experiment 3, the target was composed of 12 Gabor patches and the parameter α was $\pm 30^\circ$, as in Experiment 1. In contrast to Experiments 1 and 2, the patch orientation with respect to the apparent line segment, β , varied in Experiment 3. This enabled us to extract only the “global” effect, when the closed target is significantly salient, while subjects are no more successful in detecting the open target than they would be if their responses were made by random choice. Here we used $\beta = \pm 0^\circ, 15^\circ, 20^\circ, 25^\circ, 30^\circ$ degrees. Therefore the continuity of the patch orientation in a target was not assured at each point in the target (Fig. 2E). The value of D was fixed to 3λ .

As can be seen in Fig. 2F, for $\beta < 30^\circ$, the open targets were significantly more salient than in the case that subjects could only make random responses, and this means that the local enhancing mechanism should still be effective in this range of β in which the orientations of neighboring patches are considerably different.

For $\beta = 30^\circ$, appearance of the open targets resulted in random responses, while the closed targets were still significantly salient. This suggests that the local enhancing mechanism is no longer effective when $\beta = 30^\circ$. This saliency can only be the result of some global mechanisms used to evaluate global shapes of the targets.

4 Discussion

We succeeded in demonstrating a global effect in contour integration within a band-passed range of spatial filters in the visual system through Experiments 2

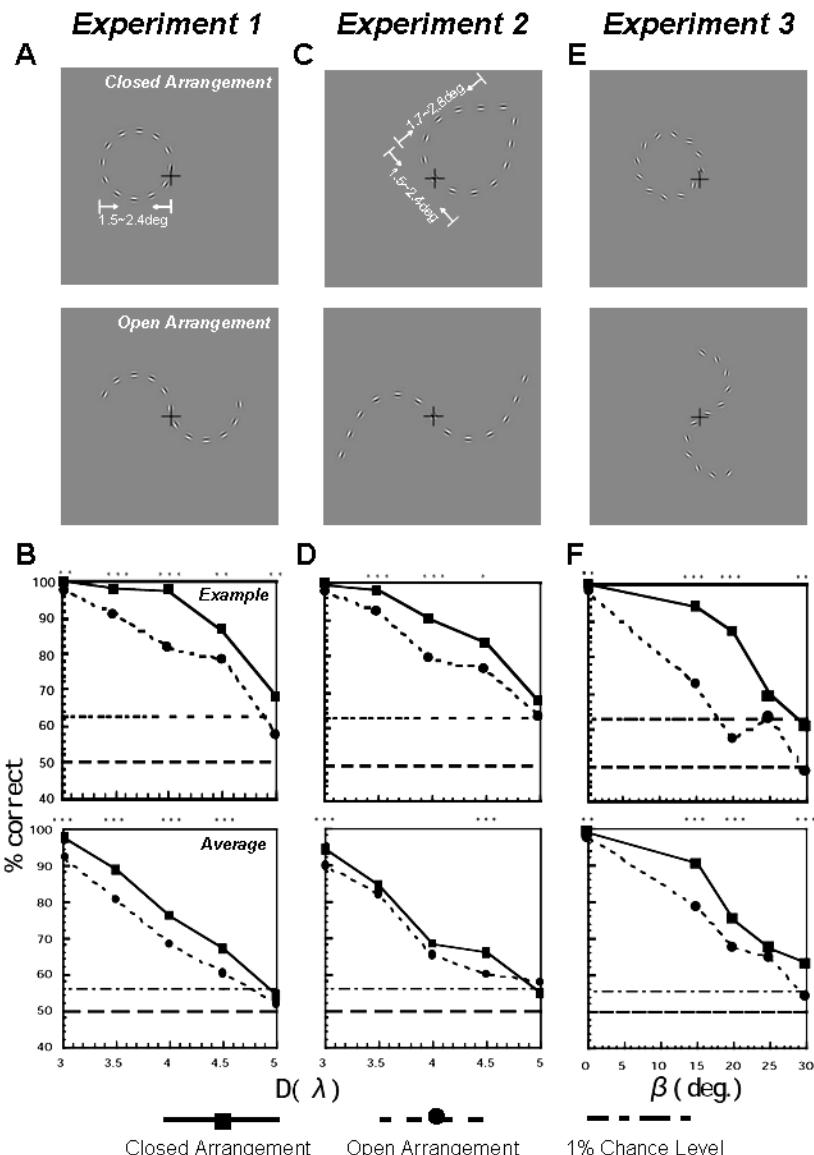


Fig. 2. Results of the experiments. (A)(C)(E) Examples of targets used in Experiment 1, 2 and 3, respectively. Upper and lower rows show closed and open arrangements in each experiment. The range of the target is also indicated. (B)(D)(F) The result from Experiment 1, 2 and 3, respectively. Upper and lower rows show the result of a subject and average across all subjects, respectively. Dashed lines indicate the 1% chance level of random choice. Here “*” corresponds $p < 0.05$ significant level in the one-sided binomial test, “**” to a $p < 0.05$ significant level in the two-sided binomial test, and “***” to a $p < 0.01$ significant level in the two-sided binomial test.

and 3. In Experiment 2, the stronger saliency of a closed target with a “corner”, or a local discontinuity in the adjacent patch orientation that does not cause saliency enhancement thorough the local association field mechanism compared with that for an open target strongly suggests the existence of some global effects of the target shapes. We found further support for the existence of such effects by isolating the global closure from the local association field mechanism in Experiment 3 by using a patch alignment in which the orientations of neighboring patches differed. If the saliency of each target in Experiments 2 and 3 had been caused only by local enhancing mechanisms including a line-length effect, we would have found the same results for open and closed arrangements. This was, however, not the case.

The contour of a given visual object almost always has some abrupt changes in orientation, and some parts of it are often covered or shaded by other objects. However, an object occupies a finite area in field of vision. So it must be useful to appreciate global closure of a visual object despite some discontinuities of local orientation of its contour.

However, all globally-closed arrangements of Gabor patches seem not be segregated from noise patches. The fact that a moon-like arrangement of patches did not cause saliency enhancement suggests that a large concavity in global closure of patch arrangement disturbs the global-closure effect [3]. On the other hand, recent studies insisted that concentric arrangement of Gabor patches is important in causing patch saliency [6]. By this idea, lack of saliency in moon-like arrangement can be understood easily. That is, if an arrangement contains a large portion of concavity, this portion will incite a “center” outside of the closed area, which may disrupt the saliency of the closed arrangement of patches.

Such an enhancement of saliency for concentric arrangements of Gabor patches may reflect a visual processing for reference point in object-centered representation of features. In fact, some extrastriate neurons seem to encode local features depending on in which side or direction the feature belongs to a shape [7][8]. And this fact seems consistent with the fact that the concentric arrangements of patches activated extrastriate areas [6]. Though it still remains to be investigated what is, in particular, calculated for global closure, the processing discussed here should serve as a basis for the positional invariant representation of shape.

Acknowledgments. This work was supported by the grant from the Flexible Information System Research Center of the Research Institute of Electrical Communication, Tohoku University.

References

1. Field, D.J., Hayes, A., Hess, A., Hess, R.F.: Contour Integration by the Human Visual System: Evidence for a Local “Association Field”. *Vision Res.* 33, 173–193 (1993)
2. Kovacs, I., Julesz, B.: A Closed Curve is Much More than an Incomplete One: Effect of Closure in Figure-Ground Segmentation. *Proc. Natl. Acad. Sci. USA* 90, 7495–7497 (1993)

3. Pettet, M.W., McKee, S.P., Grzywacz, N.M.: Constraints on Long Range Interactions Mediating Contour Detection. *Vision Res.* 38, 865–879 (1998)
4. Achtman, R.L., Hess, R.F., Wang, Y.Z.: Sensitivity for Global Shape Detection. *J. Vision* 3, 616–624 (2003)
5. Mathes, B., Fahle, M.: Closure Facilitates Contour Integration. *Vision Res.* 47, 818–827 (2007)
6. Dumoulin, S.O., Hess, R.F.: Cortical Specialization for Concentric Shape Processing. *Vision Res.* 47, 1608–1613 (2007)
7. Zhou, H., Friedman, H.S., von der Heydt, R.: Coding of Border Ownership in Monkey Visual Cortex. *J. Neurosci.* 20, 6594–6611 (2000)
8. Pasupathy, A., Connor, C.E.: Shape Representation in Area V4: Position-Specific Tuning for Boundary Conformation. *J. Neurophysiol.* 86, 2505–2519 (2001)

Modeling of Associative Dynamics in Hippocampal Contributions to Heuristic Decision Making

Miki Hirabayashi* and Hirodata Ohashi**

Department of Quantum Engineering and Systems Science, The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
miki@crimson.q.t.u-tokyo.ac.jp, ohashi@q.t.u-tokyo.ac.jp

Abstract. We present a new analysis on the heuristic strategy developed in the hippocampal circuit through the memory and learning process. A heuristic approach rapidly leads a solution close to the best possible answer utilizing easy-access information under the situation in which it is difficult to find the best answer. Focusing on the day trading, which needs the rapid decision making within a restricted time, we demonstrate that the heuristic strategy emerges in the process of the memory integration through the compensation for the limit of the information processing ability of the brain. We expect that findings from our trials will help to reveal the hippocampal role on the establishment of decision making strategies and provide the new idea in order to predict the social behavior or improve the current computer power.

1 Introduction

A heuristic approach is a strategy to solve a problem by utilizing easy-access information [1,2]. It rapidly leads a solution close to the best possible answer under the situation in which it is difficult to find the best answer due to the complexity of the problem, the incompleteness of the information or the insufficiency of the time to consider.

In this paper, we present an analysis on the formation of the heuristic decision making strategy by the contribution of the hippocampus. The hippocampus extracts information features, associates them and converts to long-term memories [3-7]. Focusing on the day trading, we demonstrate that heuristic strategy can emerge through the memory and learning process in the hippocampus by the Hopfield's associative model [8]. The day trading needs the rapid judgment in a limited time and therefore heuristic strategy is often effective when it is difficult to find the best solution.

* Present address: Kobe Advanced ICT Research Center, National Institute of Information and Communications Technology, 588-2 Iwaoka, Iwaoka-chou, Nishi-ku, Kobe, Hyogo 651-2492, Japan. m_hirabayashi@office.so-net.ne.jp

** Present address: Department of systems innovation, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan. ohashi@sys.t.u-tokyo.ac.jp

By introducing the pattern expression on the lattice, we visualize the process of the memory recall and memory integration during day trading. Our results show that the memory integration on the heuristic judgment leads a rapid possible answer and provides a practical solution for the problem of the limit of memory capacity or information processing ability of the brain.

We expect that new findings in this kind of brain strategy will make it possible to predict the social heuristic behavior and improve the processing power of the computer effectively.

2 Model

We consider the following finite state automaton with specified state transition rules as the mathematical abstraction of neural function.

2.1 Hopfield's Associative Model

Here we consider the basic Hopfield model [8]: All connections need to be symmetric, no autoconnection can exist, and the update rule must be synchronous. These constraints are sufficient to define a Lyapunov function H , a lower-bounded function whose derivative is decreasing in time,

$$H = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n T_{ij} x_i x_j , \quad (1)$$

where n is the number of neurons, T_{ij} is the neural interconnection weight between neuron x_i and neuron x_j when neuron x_i transmits information to neuron x_j . This decreasing function of nonpositive variation, $\Delta H < 0$, ensures convergence to fixed-point attractors. Each local minimum of the Lyapunov function represents one fixed point of the dynamics. These local minima can be used to store patterns. The Hopfield model is a content-addressable memory since any stored item will be retrieved when the network dynamics is initiated with a vector of activation values sufficiently overlapping the stored pattern in the case of learning a few uncorrelated patterns [9]. We consider the following simple Hopfield model [10]. The data set \mathbf{x} is an m -set of n -dimensional binary (± 1) column vectors written as $\mathbf{x} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}\}$, where m is the number of units composing the stimulus or fundamental memories. Each datum $\mathbf{x}^{(\alpha)}$ is written by $\mathbf{x}^{(\alpha)} = (x_1, x_2, \dots, x_n)^T$ (superscript T denotes transpose between a column vector and a row vector). For neural associative nets, the i th neuron state x_i is determined by the rule

$$x_i = \text{sgn} \left\{ \sum_{j=1}^n T_{ij} x_j \right\} = \begin{cases} +1, & \text{if } \sum_{j=1}^n T_{ij} x_j \geq 0 \\ -1, & \text{if } \sum_{j=1}^n T_{ij} x_j < 0 \end{cases} . \quad (2)$$

The Hopfield connection matrix \mathbf{T} for the set of m memories is defined as

$$\mathbf{T} = \sum_{\alpha=1}^m \mathbf{T}_{\alpha} = \sum_{\alpha=1}^m (\mathbf{x}^{(\alpha)}) ((\mathbf{x}^{(\alpha)})^T - \mathbf{I}_n) , \quad (3)$$

where I_n denotes the $n \times n$ identity matrix. In synchronous operation, each of the n neurons simultaneously evaluates and updates its state according to the rule shown as Equation (2).

Figure 1 illustrates an example of the structure of a neural network for associative memory. A four-neuron interconnected neural network is shown. The circles represent neurons, and the directed lines represent the direction of interneuronal information flow through the corresponding synaptic weight T_{ij} from neuron j to neuron i . Here the neural interconnection weights T_{ij} are considered to be fixed; i.e., learning of associations has already taken place, and no further synaptic modifications are made in the neurobiological interpretation. The connection matrix is also assumed to be symmetric with zero diagonals according to Ref. [10].

2.2 Hippocampal Model

The hippocampus is a brain region involved in learning and memory. As shown in Fig. 2, it contains parallel pathways: trisynaptic pathway [entorhinal cortex (EC) \rightarrow dentate gyrus (DG) \rightarrow CA3 \rightarrow CA1 \rightarrow EC] and monosynaptic pathway (EC \rightarrow CA1 \rightarrow EC). The full trisynaptic pathway is required for rapid one-trial contextual learning and for pattern completion-based memory recall [11]. It is thought that the feed-forward pathway from EC \rightarrow DG \rightarrow CA3, a part of trisynaptic pathway, plays an important role on pattern separation and a recurrent network within CA3 are substantial for the rapid association of diverse sets of information and pattern completion [3-6, 12]. The CA1 region may be required in recognizing the novelty of an event or context [13, 14]. Here we deal with this trisynaptic pathway as follows.

Entorhinal Cortex (EC). The neocortex and the hippocampus are connected through EC. The neuron in the neocortex receives an output from EC. The long-term memories are thought to be formed by modifications of synaptic weights in the neocortex after the information processing in the hippocampus. The EC acts as an interface connecting between the neocortex and the hippocampus. Thus we assume that the synaptic weights remain fixed. The EC receives outputs from the neocortex and outputs from CA1.

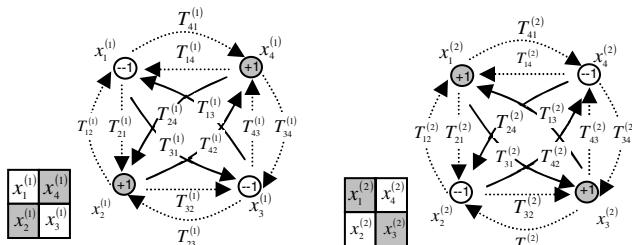


Fig. 1. A fully recurrent neural network ($n = 4$). Two shots of the network's states. Each represents the internal state of the network by a different external input.

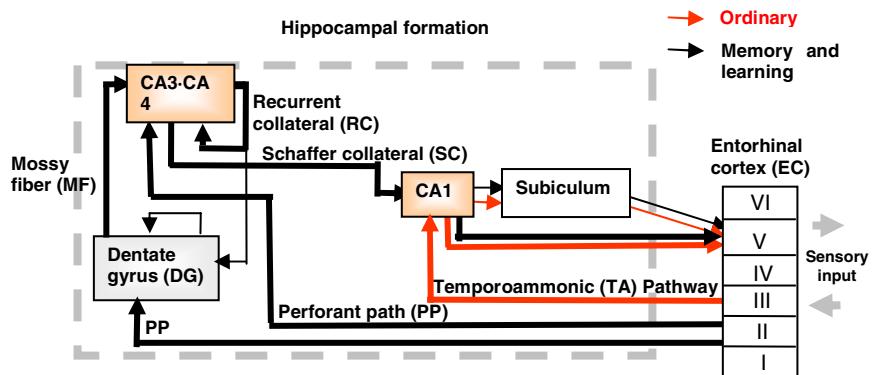


Fig. 2. The hippocampus and its connections

Dentate Gyrus (DG). It is reported that there is a dual decorrelation mechanism for pattern separation in signals from EC: changes in coincidence patterns in DG and recruitment of nonoverlapping cell assemblies in CA3 [15]. It has been suggested that, in DG, the input from EC is dispersed onto a more extensive layer of sparsely firing granule cells, enabling each granule cell to carry only a small and distinct fraction of the total input [16]. By way of the sparse level of firing in the granule cell population [17, 18] and the sparse connections between the granule cells and the pyramidal cells in CA3 [19], the segregation of incoming inputs from the cortex might be retained as the inputs entering the CA3 network. The outputs from EC are projected over all the granular cells in DG through perforant paths. The i th neuron state of one of the fundamental networks in DG is determined by

$$x_i^{DG} = \text{sgn} \left(\sum_{j=1}^n T_{ij} x_j^{DG-EC} \right). \quad (4)$$

Here x_j^{DG-EC} denotes the input from EC to the j th neuron in DG.

CA3. In the CA areas, pattern separation is expressed as a substantial reorganization of the collective firing pattern that is induced when sensory or motivational inputs to the network exceed a certain difference threshold. The CA3 network has the ability to distinguish overlapping sensory input patterns by the transformation of representations where sensory input patterns are made progressively more different. It is also suggested that the recurrent interconnected pyramidal cells of the CA3 subregion operate as an auto-association network capable of reestablishing previously stored patterns of activation based on noisy or degraded cues (pattern completion) [3, 4, 7, 20, 21]. Each neuron in CA3 receives outputs from DG through mossy fibers and outputs from CA3 themselves through the recurrent collateral connections. Here we assume that the CA3 region checks the signal from DG by comparing with the signal from EC and reconstructs the neuron state based on this comparing result in order to induce long-term memories.

CA1. The synaptic weight from the EC to CA1 remains fixed. It is thought that CA1 plays the role of decoding the orthogonal patterns from CA3 to the original input patterns by use of the outputs from the EC as teacher signals. Here we do not deal with CA1 dynamics in detail.

2.3 Day-Trading Model

The day trading is a suitable model to demonstrate the formation of the memory related on the heuristic strategy. Traders who participate in the day trading buy and sell the stock or financial instruments within the same trading day. Because it is often difficult to know the best timing to trade within a limited time, the heuristic strategy is effective to find a solution close to the best possible answer rapidly.

For the modeling of the day trading, we use two kinds of information for decision making on the selling of stocks: the upward energy of the stock price judging from the ordering information board (Fig. 3-I) and the variation rhythm judging from the tick chart of the stock price (Fig. 3-IV). Here we visualize the memory on the neural network as a pattern on the lattice by utilizing the expression shown in Fig. 1.

For example, the memory pattern “a” of the memory set I in Fig. 3-I represents the memory that we can know that the stock price has the enough upward energy when there are few sell and many buy orders with the high execution speed on the ordering information board. Similarly, the memory pattern “b” represents the memory that we can know that the stock price does not have enough upward energy when there are many sell and few buy orders with low execution speed on the ordering information board. On the other hand, the memory pattern “c” of the memory set II in Fig. 3-IV represents the memory that we can sell the stock when the profit is over 1% during a rising cycle judging from the tick chart of the stock price. Similarly the memory pattern “d” represents the memory that we should wait to sell if we have the loss over 1% when it is not in a rising cycle judging from the tick chart of the stock price. Here we must note that there is the limit of the memory capacity: The four-neuron network can store only two kinds of memory as a memory set in order to recall correctly [10].

3 Results

We demonstrate that heuristic decision making strategy in the day trading can emerge through the compensation process of the limit of information-processing ability of the brain. Here we deal with the case of the stock selling.

Figure 3-II indicates the ability of the memory recall of the Hopfield’s network by using the pattern expression on the lattice. It can be thought that the long-term potentiation (LTP) is induced and maintained in DG by this memory recall through the trial and error process. Figure 3-III shows that the probe 2 cannot recall the complete memory pattern “a” in the four-neuron network. It may be thought that this corresponds to the situation in which we cannot know whether we have the similar experience or what kind of additional information is need to judge the timing of selling.

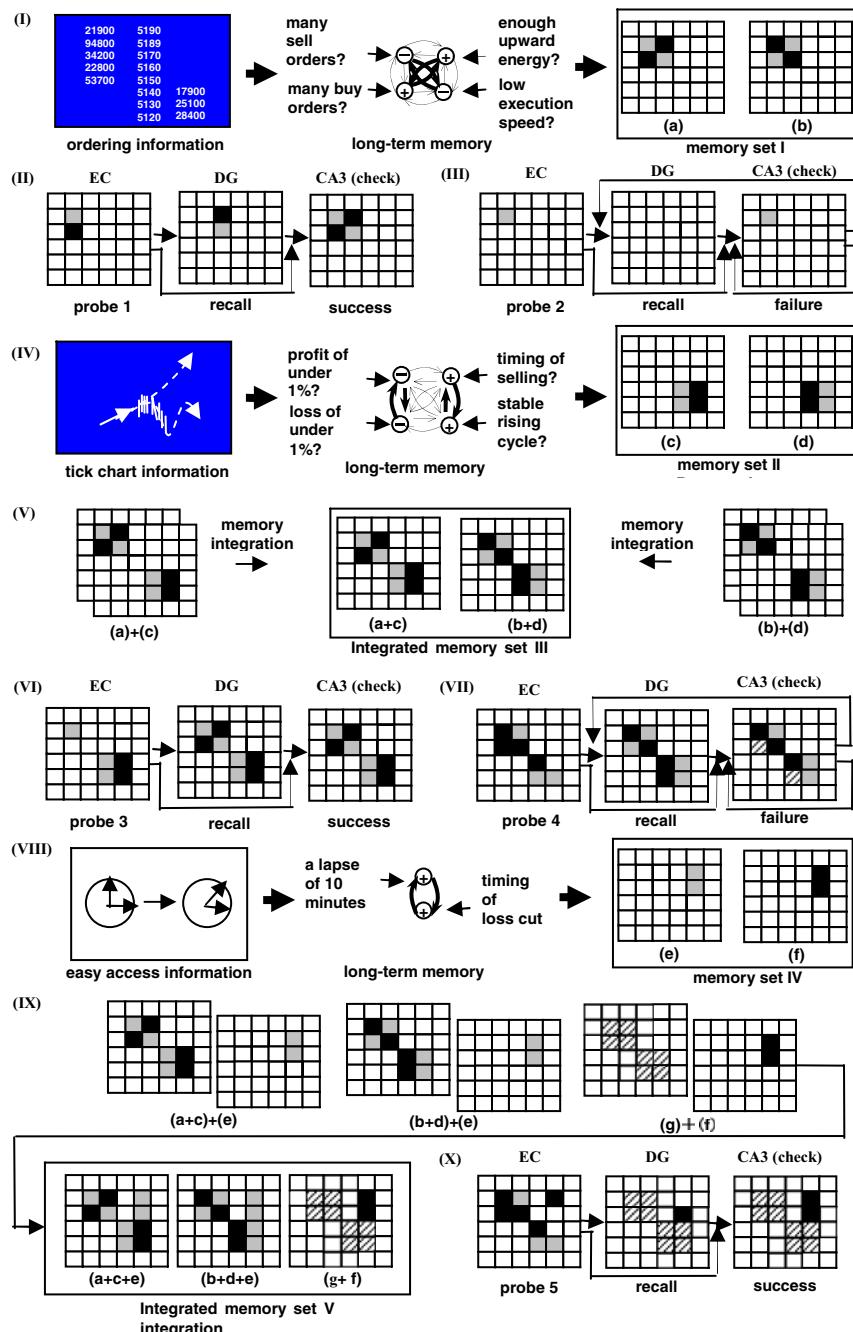


Fig. 3. The memory formation process through the trial and error process on the day trading

On the other hand, Fig. 3-VI indicates that the memory integration shown in Fig. 3-V improves the recall ability by the incomplete information such as the probe 2. Compare to the probe 2 alone in Fig. 3-III, the complete memory pattern “a” can be recalled easier by utilizing the correlated information such as the memory pattern “c”. We think that the synchronized input itself helps to induce this kind of memory integration.

As for the data conflicting with the existing memory set (probe 4 in Fig. 3-VII), the hippocampus cannot deal with it correctly. Such conflicting information may stay in the recurrent loop until the network finds a new solution to deal with it. Through the trial and error process, we may find that we can deal with this conflicting information by introduction of the easy-access information such as the time lapse (Fig. 3-VIII). For example, if the good timing of selling does not come within 10 minutes, we sell the stock in order to cut a loss ignoring other information. Figure 3-IX represents the memory integration for this strategy. The memory sets III and IV are integrated into the new memory set V. The memory pattern “g+f” in the memory set V represents that we should ignore other inputs and sell the stock after a lapse of 10 minutes. Using this memory set V, the probe 5 (probe 4 accompanied by the time-lapse information) accomplishes the successful memory recall (Fig. 3-X). Through the trial and error process, this integrated memory becomes a long-term memory as a heuristic strategy. We think this kind of integration is also performed in the trisynaptic pathway utilizing the function such as synchronization, however, we did not discuss its detail here.

We showed that the heuristic decision making strategy can emerge in order to exit from the loop. By utilizing this practical strategy, we do not need to memorize all strategies for complex situations. Our visualization model based on the hippocampal memory and learning algorithm will provide the fundamental technology to find new possible and effective heuristic strategies.

4 Discussion

We presented an analysis on the physiological basis of the heuristic strategy developed in the hippocampal circuit by introducing the pattern expression on the lattice for the visualization.

As for the biological motivation, we aim to reveal a part of the hippocampal role based on such findings that the full trisynaptic pathway is required for rapid one-trial contextual learning [11]. For the comparison with other neuro-genetic models [22, 23], we need to discuss the encoding details of the hippocampal function on the neural network considering the synaptic plasticity. For the precise evaluation of our assumption, the increase of the physiological and anatomical data of hippocampus will be necessary.

Our results show that we can detect new effective heuristic strategies by utilizing the hippocampal algorithm under the limit of memory capacity. In addition to this, we expect that our trials would be useful to predict the social heuristic behavior and improve the information processing power of the computer.

References

1. Judea, P.: *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, New York (1999)
2. Gigerenzer, G., Engel, C.: *Heuristics and the Law*. MIT Press, Cambridge (2006)

3. Marr, D.: Simple Memory: A Theory for Archicortex. *Philos. Trans. R. Soc. London Ser. B* 262, 23–81 (1971)
4. O'Reilly, R.C., McClelland, J.L.: Hippocampal Conjunctive Encoding, Storage, and Recall: Avoiding a Trade-Off. *Hippocampus* 4, 661–682 (1994)
5. Treves, A., Rolls, E.T.: Computational Analysis of the Role of the Hippocampus in Memory. *Hippocampus* 4, 374–391 (1994)
6. McClelland, J.L., Goddard, N.H.: Considerations Arising from a Complementary Learning Systems Perspective on Hippocampus and Neocortex. *Hippocampus* 6, 654–665 (1996)
7. O'Reilly, R.C., Rudy, J.W.: Conjunctive Representations in Learning and Memory: Principles of Cortical and Hippocampal Function. *Psych. Rev.* 108, 311–345 (2001)
8. Hopfield, J.J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci. U S A* 79, 2554–2558 (1982)
9. Molter, C., Salihoglu, U., Bersini, H.: The Road to Chaos by Time-Asymmetric Hebbian Learning in Recurrent Neural Networks. *Neural Comput.* 19, 80–110 (2007)
10. McEliece, R., Posner, E., Rodemich, E., Venkatesh, S.: The capacity of the Hopfield associative memory. *IEEE Transactions on Information Theory*, IT-33, 461–482 (1987)
11. Nakashiba, T., Young, J.Z., McHugh, T.J., Buhl, D.L., Tonegawa, S.: Transgenic Inhibition of Synaptic Transmission Reveals Role of CA3 Output in Hippocampal Learning. *Science* 319, 1260–1264 (2008)
12. McNaughton, B.L., Morris, R.G.M.: Hippocampal Synaptic Enhancement and Information-Storage within a Distributed Memory System. *Trends Neurosci.* 10, 408–415 (1987)
13. Lisman, J.E., Otmakhova, N.A.: Storage, Recall, and Novelty Detection of Sequences by the Hippocampus: Elaborating on the SOCRATIC Model to Account for Normal and Aberrant Effects of Dopamine. *Hippocampus* 11, 551–568 (2001)
14. Kumaran, D., Maguire, E.A.: Which Computational Mechanisms Operate in the Hippocampus during Novelty Detection? *Hippocampus* 17, 735–748 (2007)
15. Leutgeb, J.K., Leutgeb, S., Moser, M.B., Moser, E.I.: Pattern Separation in the Dentate Gyrus and CA3 of the Hippocampus. *Science* 315, 961–966 (2007)
16. McNaughton, B.L., Nadel, L.: In: Gluck, M.A., Rumelhart, D.E. (eds.) *Neuroscience and Connectionist Theory*, pp. 1–63. Lawrence Erlbaum, Hillsdale (1989)
17. Jung, M.W., McNaughton, B.L.: Spatial Selectivity of Unit Activity in the Hippocampal Granular Layer. *Hippocampus* 3, 165–182 (1993)
18. Chawla, M.K., Guzowski, J.F., Ramirez-Amaya, V., Lipa, P., Hoffman, K.L., Marriott, L.K., Worley, P.F., McNaughton, B.L., Barnes, C.A.: Sparse, Environmentally Selective Expression of Arc RNA in the Upper Blade of the Rodent Fascia Dentata by Brief Spatial Experience. *Hippocampus* 15, 579–586 (2005)
19. Amaral, D.G., Ishizuka, N., Claiborne, B.: Neurons, Numbers and the Hippocampal Network. *Prog. Brain Res.* 83, 1–11 (1990)
20. Amaral, D.G., Witter, M.P.: The Three-Dimensional Organization of the Hippocampal Formation: a Review of Anatomical Data. *Neuroscience* 31, 571–591 (1989)
21. Norman, K.A., O'Reilly, R.C.: Modeling Hippocampal and Neocortical Contributions to Recognition Memory: a Complementary-Learning-Systems Approach. *Psychol. Rev.* 110, 611–646 (2003)
22. Benuskova, L., Kasabov, N.: Modeling L-LTP based on changes in concentration of pCREB transcription factor. *Neurocomputing* 70, 2035–2040 (2007)
23. Benuskova, L., Kasabov, N.: Computational neuro-genetic modeling. Springer, New York (2007)

Tracking with Depth-from-Size

Chen Zhang¹, Volker Willert², and Julian Eggert²

¹ Darmstadt University of Technology, Darmstadt D-64283, Germany

² Honda Research Institute Europe GmbH, Offenbach D-63073, Germany

Abstract. Tracking an object in depth is an important task, since the distance to an object often correlates with an imminent danger, e.g. in the case of an approaching vehicle. A common way to estimate the depth of a tracked object is to utilize binocular methods like stereo disparity. In practice, however, depth measurement using binocular methods is technically expensive due to the need of camera calibration and rectification. In addition, higher depths are difficult to estimate because of an inverse relationship between disparity and depth. In this paper a new approach for depth estimation, depth-from-sizes (DFS), is introduced. We present a human-inspired monocular method where the depth, the physical size and the retinal size of the object are estimated in a mutually interdependent manner. For each of the three terms specific measurement and estimation methods are probabilistically combined. In two evaluation scenarios it is shown that this approach is a reliable alternative to the standard stereo disparity approach for depth estimation with several advantages: 1) simultaneous estimation of depth, physical size and retinal size; 2) no stereo camera calibration and rectification; 3) good depth estimation at higher depth ranges.

1 Introduction

Visually tracking an object in depth is a sophisticated task, since a vision system has no native depth information source. Hence, the third dimension must be constructed from two dimensional information. Different methods exist for measuring object depth from two dimensional images which can be distinguished into a) monocular and binocular and b) into static and dynamic methods. Static and dynamic methods differ from each other in the point whether depth estimation comes from the camera system ego motion itself; monocular or binocular methods in the point whether information from one resp. two cameras is involved in the estimation process. In this paper we address a static monocular method for depth estimation, **depth-from-size**.

In most current state-of-the-art vision applications, the favorite measurement method for depth estimation is a static binocular one, named stereo disparity (). It relies on the inversely-proportional relationship of object depth z and object disparity d according to

$$z = f \cdot \frac{b}{d}. \quad (1)$$

The disparity d is the distance of projected positions of the object on two rectified cameras with a distance of b between their focal points (each of the cameras having a focal length of f). The advantage of this approach is the direct measurability of disparity and so, of object depth. Unfortunately stereo disparity suffers from several drawbacks. It relies on a correspondence search of a pattern on the two camera images whose quality depends on which algorithm is applied. Before the correspondence search can be applied, a technically expensive calibration and rectification of the stereo camera system is necessary to transform the images into an epipolar plane. On the other hand side, the power of stereo disparity is limited in higher depth ranges. With increasing depth and correspondingly decreasing disparity, based on the inverse proportionality, the depth estimations get more and more imprecise.

Here, we introduce a static monocular approach which combines tracking of depth and sizes of an object in a probabilistic way. This is thought in the first line not as a replacement but as an additional or alternative cue to stereo disparity. Basically the proposed method relies on the following assumed relationship between depth, retinal size and physical size of the target object:

$$z = f \cdot \frac{r_{\text{phy}}}{r}. \quad (2)$$

Depth z is determined here by combining the retinal size r - the object size in the image - and the physical size r_{phy} - the cross-section size of the real 3D object. The difficulty here is, since the retinal radius and the physical radius are also unknown in advance, they have to be simultaneously estimated together with the depth. In this approach, the estimation of each of the three terms is implemented by means of the remaining two terms in a mutually interdependent way.

In psychophysics this human-inspired static monocular depth estimation approach is known as the so-called familiar size cue (e.g. see [2]). In case of humans several experiments provide evidence about size constancy ([3]) showing that in many cases object size is perceived as constant, if the object is familiar. In this case an inversely proportional relationship between apparent size and apparent distance to the object has been found ([4], [7]). Similar effects are also observed in experiments with infants ([5]). The size constancy assumption breaks down, however, in experiments with unfamiliar objects ([6]). It is also suggested that familiar size cue plays a supporting role for disparity-based depth estimation, taking over particularly when the latter fails ([4]).

The advantage of the presented depth-from-size method is that it is applicable to monocular camera settings (avoiding multi-camera calibrations). Furthermore, it maintains a reliable depth estimation over periods where no direct depth measurement (e.g. via disparity) is available. In addition, it provides good depth hints for larger distances. Comparing equations 1 and 2 it can be seen that depth-from-size leads to more accurate results if $r_{\text{phy}} > b$, i.e. as soon as the physical object size is larger than the stereo camera baseline.

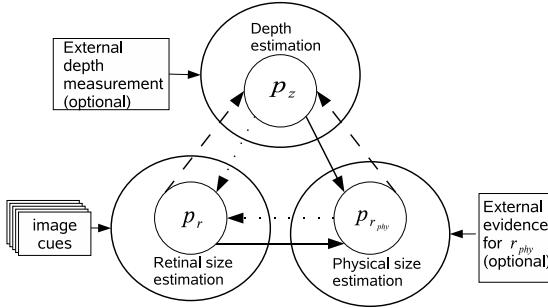


Fig. 1. Overview of the proposed system. Three components work together and are responsible for estimating the depth probability density function (PDF) p_z , the physical size PDF $p_{r_{phy}}$ and the retinal size PDF p_r . Each pair of components delivers its results to the remaining component. Optional measurements (e.g. for z from a disparity-based depth estimation) or higher level prior knowledge (e.g. for the physical size) is occasionally injected into the system. The measurement of retinal size is assumed to be more frequent, and is the leading force that drives the depth estimation. In addition, we rely on a multi-cue 2D tracking component which takes care of localizing the object in the image (not shown).

2 Approach

The proposed approach consists of three mutually interdependent components for estimating the depth, the retinal size and the physical size of the target object, respectively. Each of the three components consists of two parts: measurement and state estimation, implemented with probabilistic methods to capture uncertainties of the process. Figure 1 shows an overview of the entire system. The system is fed by optional measurements for z , r and r_{phy} which provide correcting evidence. In addition, the components are coupled by the constraint (2).

2.1 Retinal Size Estimation

The first component for estimating the retinal size r contains two steps, as shown in figure 2. First, it actively measures the current object size in the image using a pyramidal KLT algorithm as described in [8]. The KLT algorithm ([9], [10]), is a gradient-descendent optimization method for estimating the affine transformation between two image patches. Iteratively applying KLT on a resolution pyramid ([8]) allows us to estimate larger transformation parameter ranges, which are necessary for our system. Utilizing the pyramidal KLT algorithm, a scaling factor of the current retinal object size relative to an initial retinal object size is obtained from which the current retinal size is calculated. Setting the found retinal size as mean and the matching value of the scaling factor as variance a Gaussian PDF is generated as KLTs measurement result.

Since KLT is a differential gradient-descendent approach, a good starting point as prior is essential to get to a reasonable local minimum. To find such a

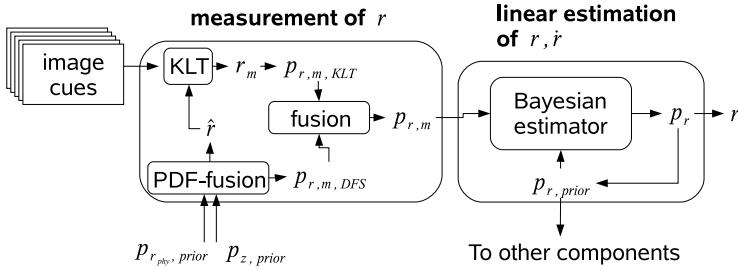


Fig. 2. Retinal size estimation. First, from the fusion of the current $p_{r,phy,prior}$ and $p_{z,prior}$ a guess of the retinal size PDF $p_{r,m,DFS}$ is obtained. From this, an expected retinal size \hat{r} is extracted as starting point to initialize the pyramidal KLT algorithm for multiple incoming cues. From the result of the KLT algorithm a measurement PDF $p_{r,m,KLT}$ is generated, which is combined multiplicatively with $p_{r,m,DFS}$, the fusion result of $p_{r,phy,prior}$ and $p_{z,prior}$, to get a measurement PDF $p_{r,m}$ for the retinal size. This is smoothed by a Bayesian estimator, in order to get the posterior PDF for retinal size p_r .

starting point, we use the information about the current physical size and depth in form of their PDFs to calculate a guess of the current retinal size for the KLT algorithm. Details about the fusion of PDFs are explained in section 2.5. In addition, in our system the KLT approach is applied on multiple cues (color, structure tensor, ...) simultaneously.

The final retinal size measurement PDF is gained by a further combination with the fusion result (see figure 2).

In a second step, the measured retinal size PDF is processed by a recursive Bayesian filter (e.g. see [1]) with an internal linear motion prediction model. The output of the filter, the posterior PDF, involves the measurement and prediction influences and represents a belief about the distribution of retinal size.

2.2 Physical Size Estimation

The second component, shown in figure 3, is used for physical size estimation and contains two similar steps as the retinal size estimation from section 2.1. In a first step, the measurement of the physical size PDF is done by just fusing the PDFs of retinal size and depth provided by the other two components (see details about how to fuse the PDFs in section 2.5). If external knowledge or evidence for the physical size is available the information is also incorporated. In a second step, we assume that the physical size of the visible object front remains constant over time. That motivates us to use a weighted mean estimator to estimate the physical size PDF from previously measured physical size PDFs.

2.3 Depth Estimation

The third component, shown in figure 4, estimates the object depth as before in two steps. First, the object depth is measured by fusing the physical and retinal

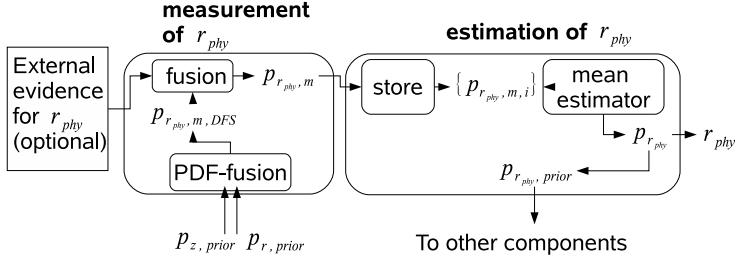


Fig. 3. Physical size estimation. The incoming PDFs $p_{z, prior}$ and $p_{r, prior}$ are combined in order to measure the current physical size PDF $p_{r_{phy}, m}$. If an external evidence of physical size exists, it is incorporated as an additional prior. All measured physical size PDFs are stored in a pool. The mean of weighted previous PDFs is then used as an estimate of the current physical size PDF $p_{r_{phy}}$.

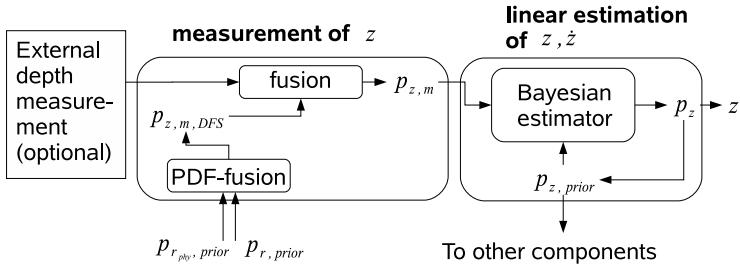


Fig. 4. This figure shows the depth estimation. From the fusion of the incoming PDFs $p_{r_{phy}, prior}$ and $p_{r, prior}$ the current depth is measured using the depth-from-size method as $p_{z, m, DFS}$. Afterwards optional external depth measurement results, e.g. from stereo disparity, can be integrated into one overall depth measurement PDF $p_{z, m}$. This PDF is fed into a Bayesian estimator where the depth is estimated by involving a linear prediction model for z .

size PDFs delivered by the other two components taking into consideration the constraint of formula 2. Optional depth measurements are again incorporated as additional cues, e.g. a depth measurement result by utilizing stereo disparity. Integration of additional external cues improves the reliability of the depth estimation and so of the entire system.

In the second step, the result of the depth measurement cues fused into a single PDF is fed into a Bayesian estimator with a linear motion prediction model for z similar to the case of the retinal size estimation.

2.4 System Operation and Initialization

The constraint (2) couples one state, e.g. the desired object depth z , with the two other states, e.g. the retinal size of the appearance of the object and its physical size. In standard operation mode, only the retinal size is injected into the system

in form of new measurements. If the system has acquired a good estimate of an objects physical size and if the physical size constancy assumption holds, the depth can then be estimated when the objects appearance size changes.

If, however, two parameters are uncertain, e.g. the object depth z **and** the physical size r_{phy} , there is an inherent ambiguity in the system: e.g. for the same retinal size an object can be smaller and nearer, or larger and further apart. For grounding the system, it is therefore necessary to provide a second measurement - either of the depth z or of the physical size r_{phy} - for a short time period. This is especially the case when the system is started or when the tracked object rotates in 3D, and so the size of its visible front varies.

In the same way, the proposed system with an additional disparity-based depth measurement would be able to bootstrap itself when an object is near and disparity measurement is reliable. For larger distances, disparity-based depth measurement gets increasingly unreliable and depth-from-size then dominates the depth estimation results, allowing a better depth estimation for far distances.

2.5 Fusion of PDFs

In an ideal probabilistic setting, the state variables z , r and r_{phy} are coupled in a joint PDF $p(z, r, r_{\text{phy}})$. In the system proposed in this paper, however, we use three separate estimation lines, each with its own PDF, but with a mutual constraint between them. The mutual influence of the PDFs is motivated as follows. We start with $p(z)$ (for $p(r)$ and $p(r_{\text{phy}})$ the procedure is analogous) and use:

$$p(z) = \int \int \underbrace{p(z, r, r_{\text{phy}})}_{:=p(z|r, r_{\text{phy}}) p(r, r_{\text{phy}})} dr dr_{\text{phy}} \quad (3)$$

and assume the unconditional prior $p(r, r_{\text{phy}}) = p(r) p(r_{\text{phy}})$ to factorize. Instead of using the “true“ $p(r)$ resp. $p(r_{\text{phy}})$, we now take the priors $p_{\text{prior}}(r)$ resp. $p_{\text{prior}}(r_{\text{phy}})$ from the estimators (see $p_{r, \text{prior}}$, $p_{r_{\text{phy}}, \text{prior}}$ from figures 2, 3), to get

$$p(z) \approx \int \int p(z|r, r_{\text{phy}}) p_{\text{prior}}(r) p_{\text{prior}}(r_{\text{phy}}) dr dr_{\text{phy}}. \quad (4)$$

For $p(z|r, r_{\text{phy}})$, we use the condition ② in a form

$$p(z|r, r_{\text{phy}}) = \delta(z - \hat{z}(r_{\text{phy}}, r)) \quad \text{with} \quad \hat{z}(r_{\text{phy}}, r) = f \cdot \frac{r_{\text{phy}}}{r} \quad (5)$$

to arrive at

$$p(z) = \int \int \delta(z - \hat{z}(r_{\text{phy}}, r)) p_{\text{prior}}(r) p_{\text{prior}}(r_{\text{phy}}) dr dr_{\text{phy}} \quad (6)$$

to get the fused PDF that is used in the measurement step of the component for the depth estimation, shown in figure ①. As an approximation, in equation 6 we use priors generated from r and r_{phy} of the last frame in order to avoid cyclic dependencies for the Bayesian inference.

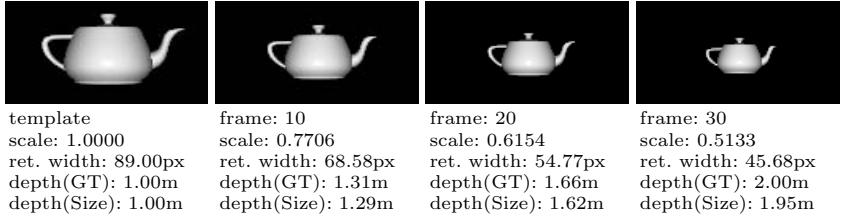


Fig. 5. In this artificial scenario with available ground truth data a teapot with a width of 32 cm is moving in depth from 1 m to 2 m with a constant velocity. The object is tracked using depth-from-size. The retinal size, the depth ground truth and the depth estimated via depth-from-size (in this order) are shown.

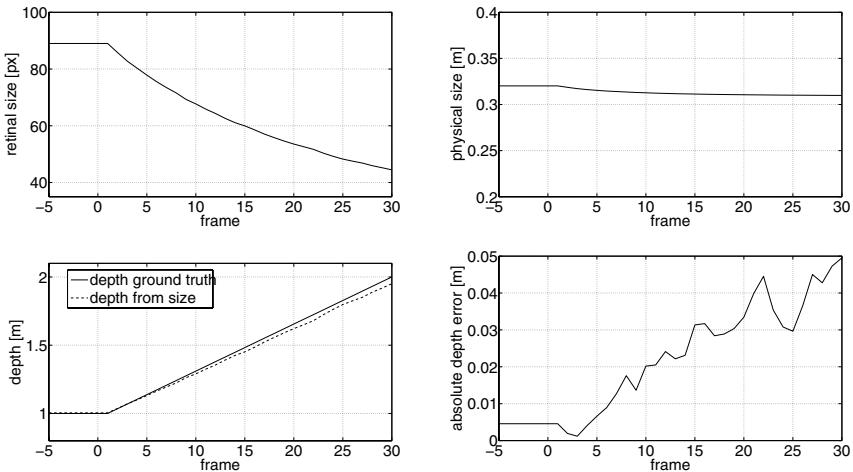


Fig. 6. Tracking result for the artificial scenario. Upper left: the estimated retinal size during the motion process; upper right: the estimated physical size; lower left: the depth ground truth and the estimated depth using depth-from-size; lower right: the absolute depth estimation error. In the first 5 frames stereo disparity is used to adjust the physical size. In the following 30 frames the stereo disparity depth cue is switched off, so depth-from-size is used as the only depth cue.

3 Evaluation Results

For showing the performance of the proposed approach this system is evaluated in two scenarios: An artificial with ground truth and a real-world scenario. In the latter, depth estimation performance of this system is compared with that of independently measured stereo disparity.

Using PDFs with a quantization of 500 values, a computation time of less than 1 second/frame was achieved during the evaluation in both scenarios running on an Intel Xeon CPU of 2.66GHz.

3.1 Artificial Scenario with Ground Truth

In the first evaluation scenario an artificial scene is taken. A teapot with a width from left to right of 32 cm is moved into the distant depth with a constant velocity. From a fixed camera with a focal length of 4.9104 mm and pixel width of 0.0176 mm/px images are grabbed as shown in figure 5. Depth is estimated using depth-from-size where first in an initialization phase of 5 frames physical size is inferred by using stereo disparity as an additional cue to actively measure depth. After the initialization phase, the stereo disparity cue is switched off, and the object is moved from 1 m to 2 m distance within 30 frames. In figure 6 the estimation results by depth-from-size can be seen in comparison to the ground truth depth data.

3.2 Real-World Scenario

In a second experiment the system is evaluated in a real-world scenario where images (with a pixel width of 0.0176 mm/px) are grabbed from a camera (with a focal length of 8.4304 mm) installed in a car, while the car is driving backwards as shown in figure 7. The distance to a car standing in front is measured by means of stereo disparity on one hand and is estimated merely by means of depth-from-size



Fig. 7. In this real-world scenario the camera system is moving backwards while images from a parking car are grabbed. The depth of the car is tracked using the proposed depth-from-size method on one hand and a stereo disparity method on the other hand. See also figure 8.

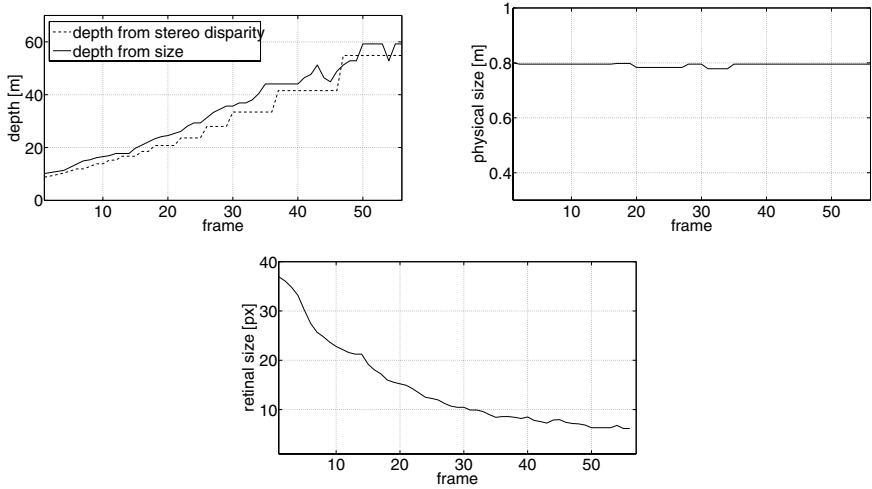


Fig. 8. Tracking results for the real-world scenario. In the upper left graph the depth tracking results from stereo disparity and depth-from-size are shown. The upper right graph shows the estimated physical size, the lower graph the retinal size.

on the other hand side. In case of the depth-from-size approach a prior about the cars physical size (0.80 m radius) is involved merely in the first frame. Afterwards the system runs without any other depth cues or further information about physical size. The depth obtained by the depth-from-size method is compared to the depth obtained by a stereo disparity method (with two identical cameras of the same focal length of 8.4304 mm and a baseline of 0.348 m) in figure 8. In the same figure the obtained physical and retinal sizes are also shown. One can see that the proposed system works quite reliably in this scenario.

4 Conclusion

In this paper we present a system for depth estimation from object size. It mutually estimates depth, retinal and physical sizes in a bootstrapping way. Each of the three components for estimating these three terms consists of two steps, one for the measurement and one for the state estimation. A probabilistic treatment of the results of single components allows to propagate them and so their quality measures to the other components, imposing a constraint that couples the three states.

In our experiments, the proposed system successfully tracks target objects in depth. Its performance in higher depth is comparable or better than that of stereo disparity, since it can use the precise region covered by the objects appearance, whereas stereo methods rely on the comparison of small image patches, on the one hand and it is easier to measure size change than disparity changes for objects which are larger compared to the stereo baseline on the other hand.

Without support of additional depth cues during runtime, an initial learning phase for the physical size is necessary either by injecting an additional depth cue for inferring the physical size or by a direct prior on the physical size. Although the physical size is adjusted in this initialization phase, it is not fixed and will still be adapted during runtime. This behavior is analogous to findings about human depth estimation of familiar objects, where prior knowledge about the absolute size of a target object is utilized in conjunction with the retinal size of an object to infer its distance.

References

1. Qian, N.: Binocular disparity and the perception of depth. *Neuron* 18, 359–368 (1997)
2. Palmer, S.E.: *Vision Science: photons to Phenomenology*. MIT Press, Cambridge (1999)
3. Kaufman, L., Kaufman, J.H., Noble, R., Edlund, S., Bai, S., King, T.: Perceptual distance and the constancy of size and stereoscopic depth. *Spatial vision* 19, 439–457 (2006)
4. DeLucia, P.R.: Does binocular disparity or familiar size information override effects of relative size on judgements of time to contact? *The Quarterly Journal of Experimental Psychology* 58, 865–886 (2005)
5. Marotta, J.J., Goodale, M.A.: Role of Familiar Size in the Control of Grasping. *J. Cognitive Neuroscience* 13, 8–17 (2001)
6. Gogel, W.C., Newton, R.E.: Perception of off-sized objects. *Perception and Psychophysics* 5, 7–9 (1969)
7. Gogel, W.C.: The effect of object familiarity on the perception of size and distance. *The Quarterly Journal of Experimental Psychology* 21, 239–247 (1969)
8. Eggert, J., Zhang, C., Körner, E.: Template Matching for Large Transformations. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) *ICANN 2007. LNCS*, vol. 4669, pp. 169–179. Springer, Heidelberg (2007)
9. Lucas, B.D., Kanade, T.: An iterative Image Registration Technique with an Application to Stereo Vision. In: *International Joint Conference on Artificial Intelligence*, pp. 674–679 (1981)
10. Shi, J., Tomasi, C.: Good Features to Track. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600 (1994)
11. Ristic, B., Arulampalam, S., Gordon, N.: *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers (2004)

Training Recurrent Connectionist Models on Symbolic Time Series*

Michal Černánský¹ and Ľubica Beňušková²

¹ Faculty of Informatics and Information Technologies, STU Bratislava, Slovakia

² Department of Computer Science, University of Otago, Dunedin, New Zealand

cernansky@fiit.stuba.sk, lubica@cs.otago.ac.nz

Abstract. This work provide a short study of training algorithms useful for adaptation of recurrent connectionist models for symbolic time series modeling tasks. We show that approaches based on Kalman filtration outperform standard gradinet based training algorithms. We propose simple approximation to the Kalman filtration with favorable computational requirements and on several linguistic time series taken from recently published papers we demonstrate superior ability of the proposed method.

1 Introduction

To process data with spatio-temporal structure recurrent neural networks (RNNs) were suggested. RNNs were successfully applied in many real-life applications where processing time-dependent information was necessary. Unlike feedforward neural networks, units in RNNs are fed by activities from previous time steps through recurrent connections. In this way contextual information can be kept in units' activities, enabling RNNs to process time series.

Many commonly used real-world data with time structure can be expressed as a sequence of symbols from finite alphabet - symbolic time series. Since their emergence neural networks were applied to symbolic time series analysis. Especially popular is to use connectionist models for processing of complex language structures. One of the main driving forces behind such studies has been formulating the models of human performance in processing linguistic patterns of various complexity [12]. Other works study what kind of dynamical behavior has to be acquired by RNNs to solve particular tasks such as processing strings of context-free languages, where counting mechanism is needed [34].

Gradient descent approaches such as backpropagation through time or real-time recurrent learning algorithms are widely used by researchers working with symbolic sequences. The aim of this paper is to show how KF-based techniques used for training RNNs can deal with symbolic time series. We also compare standard means square error cost function with cost function based on the entropy. Simple approximation significantly reducing the complexity of the Kalman filtration is proposed and results on the linguistic datasets taken from recently published paper are shown.

* This work was supported by the grants VG-1/0848/08 and VG-1/0822/08.

2 Recurrent Neural Network Training

Elman's simple recurrent network (SRN) proposed in [1] is probably the most widely used RNN architecture.

It was trained using well-known error backpropagation algorithm. Although simple backpropagation is not appropriate algorithm for training recurrent networks it can be used for simple tasks or as a reference method for other approaches.

Backpropagation Through Time

Backpropagation through time algorithm (BPTT) provides precise error gradient computation for RNNs training. The trick consists of unfolding an RNN in time into regular feedforward network and then apply standard backpropagation directly. Usually approximation to the precise calculation is performed and recurrent network is unfolded into feedforward network only for several time steps called time window. Later theoretical results revealed that when acquiring long-term dependencies the error information of gradient-based training algorithms tends to vanish or blow up, therefore unfolding the recurrent network backwards for long time interval is not necessary [5]. Probably the most contributive article regarding BPTT is [6].

Adaptation of Extended Kalman Filter to Neural Network Training

Training of Elman's SRN, and generally any other multilayer perceptron network (recurrent or not), can be regarded as an optimal filtering problem [7]. Multilayer perceptron network can be described as nonlinear system by

$$\mathbf{x}(t) = \mathbf{x}(t - 1), \quad (1)$$

$$\mathbf{z}(t) = h(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{v}(t), \quad (2)$$

where state $\mathbf{x}(t)$ is vector of network weights. Weights of trained network do not change and so state transition matrix $\mathbf{F} = \mathbf{I}$ where \mathbf{I} is identity matrix. Measurement $\mathbf{z}(t)$ stands for desired values and measurement function $\mathbf{H}(t)$ is nonlinear function of network weights $\mathbf{x}(t)$ and input $\mathbf{u}(t)$. Jacobian matrix $\mathbf{H}(t)$ is calculated in every time step k as

$$\mathbf{H}(t) = \frac{\partial \mathbf{H}(\hat{\mathbf{x}}(t), \mathbf{u}(t))}{\partial \mathbf{x}}. \quad (3)$$

The set of EKF equations for the network training can be formulated as follows:

$$\mathbf{K}(t) = \mathbf{P}(t - 1)\mathbf{H}^T(t)[\mathbf{H}(t)\mathbf{P}(t - 1)\mathbf{H}^T(t) + \mathbf{R}(t)]^{-1}, \quad (4)$$

$$\mathbf{P}(t) = \mathbf{P}(t - 1) - \mathbf{K}(t)\mathbf{H}(t)\mathbf{P}(t - 1) + \mathbf{Q}(t), \quad (5)$$

$$\mathbf{W}(t) = \mathbf{W}(t - 1) + \mathbf{K}(t)[\mathbf{D}(t) - \mathbf{O}(t)]. \quad (6)$$

Let n_w and n_o denote the number of all network weights and number of output units, respectively. \mathbf{W} is a vector of all weights (concatenated from matrices \mathbf{W}^{RI} , \mathbf{W}^{RC} , \mathbf{W}^{OR}) of the length n_w . \mathbf{H} is the Jacobian matrix, $n_o \times n_w$, calculated in every time step and containing in rows the derivatives of corresponding output activity with respect

to all weights. These derivatives can be calculated by routines similar to BPTT [6]. \mathbf{P} is the $n_w \times n_w$ error covariance matrix, it holds error covariances corresponding to each pair of network weights. The $n_w \times n_o$ matrix \mathbf{K} called the Kalman gain is used in updating the weights \mathbf{W} according to the difference between the desired output vector \mathbf{D} and actual network output \mathbf{O} . The $n_o \times n_o$ matrix \mathbf{R} stands for the measurement noise covariance matrix and similarly to the learning rate in RTRL or BPTT can control the training speed of EKF. Higher values represent higher amount of uncertainty attributed to the difference $\mathbf{D}(t) - \mathbf{O}(t)$ leading to slower training. Note, that small process noise is still considered: the $n_w \times n_w$ matrix \mathbf{Q} stands for the process noise covariance matrix. Nonzero process noise improves convergence of the filter.

Entropy Cost Function Training Algorithms

Output units of RNNs are often equipped with logistic sigmoid or linear activation function and quadratic-error cost function was minimized in order to derive equations for weight adaptation. Alternative approach better suited to the symbolic time series processing scenario is to use output units with linear activation function and soft-max combining function. Predictive performance is evaluated using the probabilities $p(t)$ calculated using soft-max combining function:

$$p(t) = \frac{\exp(o_i(t))}{\sum_j \exp(o_j(t))}, \quad (7)$$

where o_i is the activation of output unit corresponding to the observed target symbol. Training algorithms are modified in order to minimize entropy cost function $E(t)$ in each weight update step t :

$$E(t) = -\log_{|A|} p(t). \quad (8)$$

Entropy Cost Function KF Approximation

Kalman filter is derived to minimize quadratic error cost function. Approximation technique is proposed to minimize the entropy. Neural network can be considered as a system with a single output of $-\log p(t)$ where the probability $p(t)$ is calculated from output unit activities by soft-max combining function (Eq. 7). Advantage of the single output system is that no matrix inversion need to be calculated what can significantly reduce the time complexity of the algorithm. The price for reducing the computational complexity may be poorer training performance. In this work this approximation to the Kalman filtration for RNN training will be called SM-EKF (soft-max EKF). Similar KF approximation as SM-EKF called “scalar error training” was proposed in [7].

3 Experiments

Standard gradient descent training techniques represented by simple backpropagation and backpropagation through time algorithms are first compared with standard extended Kalman filter adopted for RNN training with derivatives calculated by BPTT-like algorithm. Standard versions of algorithms derived using quadratic-error cost function and

also versions using soft-max combining function minimizing entropy cost function were compared on simple symbolic dataset: sequence of quantized activations of real laser and the sequence generated by context free grammar. Then proposed approximation to the Kalman filtration is used for processing linguistic datasets and resulting performance is compared to published results.

Simple Symbolic Datasets

The Laser dataset was obtained by quantizing activity changes of laser in chaotic regime, where relatively predictable subsequences are followed by hardly predictable events. The original real-valued time series was composed of 10000 differences between the successive activations of a real laser. The series was quantized into a symbolic sequence over four symbols corresponding to low and high positive/negative laser activity change. The first 8000 symbols are used as the training set and the remaining 2000 symbols form the test data set [8].

Deep Recursion data set is composed of strings of context-free language L_G . The set of production rules is composed of three simple rules: $R \rightarrow aRb | R \rightarrow ARB | R \rightarrow e$ where e is the empty string. This language is in [3] called palindrome language. The training and testing data sets consist of 1000 randomly generated concatenated strings. No end-of-string symbol was used. Shorter strings were more frequent in the training set than the longer ones. The total length of the training set was 6156 symbols and the length of the testing set was 6190 symbols.

Linguistic Datasets

For our experiments we have chosen two languages from recently published papers. The first language (labeled EG) was generated by Elman's grammar [9] and was also used in [10]. Dataset alphabet consists of 24 words including end-of-sentence marker. The second dataset (labeled CG) was generated using the grammar from [11]. Language was composed of 72 words including end-of-sentence marker. For both tasks the training and the test set were composed of 10000 words. Ground true probabilities were recorded during datasets creation.

Language entropy H can be estimated using generated samples from the probabilistic grammars:

$$H = -\frac{1}{T} \sum_{t=1}^T \sum_{a \in A} G_t(a) \log_{|A|} G_t(a), \quad (9)$$

where $G_t(a)$ is the ground true probability of generating symbol a in time step t and $|A|$ is the size of the language alphabet A . The longer the sample, the more accurate the entropy estimation, nevertheless we have used test sets to provide entropy estimations: 0.535 for EG dataset and 0.398 for CG dataset.

Method

First BP, BPTT and standard EKF with both quadratic error cost function and entropy cost function (prefix SM: SM-BP, SM-BPTT and SM-EKF-BPTT, where SM is an abbreviation for soft-max) were used to train Elman's SRN on Laser and Deep Recursion datasets. 10 training epochs for EKF and up to 100 epochs for BP and BPTT were done.

Symbols were encoded using one-hot-encoding, i.e. all input or target activities were set to 0, except the one corresponding to given symbol, which was set to 1. Unipolar 0-1 sigmoidal activation function was used.

Different parameters for BP and BPTT were used for each data set to obtain the best results. We improved training by using scheduled learning rate. We used linearly decreasing learning rate in predefined intervals. But no improvements made the training as stable and fast as the EKF training (taking into account the number of epochs). For EKF training, error covariance matrix \mathbf{P} was set to $\mathbf{P} = 1000 * \mathbf{I}$, the measurement noise covariance matrix \mathbf{R} was set to $\mathbf{R} = 100 * \mathbf{I}$, and the process noise covariance matrix \mathbf{Q} was set to $\mathbf{Q} = 0.0001 * \mathbf{I}$, where \mathbf{I} stands for the identity matrix. These values were used throughout all experiments. Matrices \mathbf{Q} and \mathbf{R} remained fixed during training. Measurement matrix \mathbf{H} was calculated in every time step using algorithm almost identical to BPTT with window size set to 10 [7].

Predictive performance was evaluated using a normalized negative log-likelihood (NNL) calculated over symbolic sequence $s_1 s_2 \dots s_t \dots s_T$ used for testing as

$$\text{NNL} = -\frac{1}{T} \sum_{t=1}^T \log_{|A|} P_t(s_{t+1}) \approx -\frac{1}{T} \sum_{t=1}^T \sum_{a \in A} G_t(a) \log_{|A|} P_t(a), \quad (10)$$

where the base of the logarithm $|A|$ is the number of symbols in the alphabet A and the $P_t(a)$ is the probability of predicting symbol a in the time step t . NNL = 0 corresponds to the 100% next-symbol prediction.

We also trained SRN on linguistic dataset using SM-BP, SM-BPTT and SM-EKF-BPTT. 10 training epochs were performed for all algorithms. The same parameters as in the previous experiments were used for SM-EKF-BPTT. For SM-BP and SM-BPTT exponentially decaying learning rate was used. We also provide results for alternative performance criteria used by the cognitive science community, such as cosine scores defined as:

$$\text{COS} = \frac{1}{T} \sum_{t=1}^T \cos(G_t(s_t), P_t(s_t)), \quad (11)$$

where \cos is the cosine between two (in our case normalized) vectors \mathbf{x} and \mathbf{y} :

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}| |\mathbf{y}|}. \quad (12)$$

Another alternative measure is averaged Kullback-Leibler divergence defined as:

$$\text{KLD} = \frac{1}{T} \sum_{t=1}^T \sum_{a \in A} G_t(a) \log_{|A|} \frac{G_t(a)}{P_t(a)}. \quad (13)$$

There is a strong correspondence between measures KLD and NNL: $\text{KLD} \approx \text{NNL} - H$, where H is the language entropy.

Results

For simple symbolic datasets we present mean and standard deviations of 10 simulations. Results for both Laser dataset and Deep Recursion are shown in Fig. 1. Unsatisfactory simulations with significantly low performance were thrown away for BP

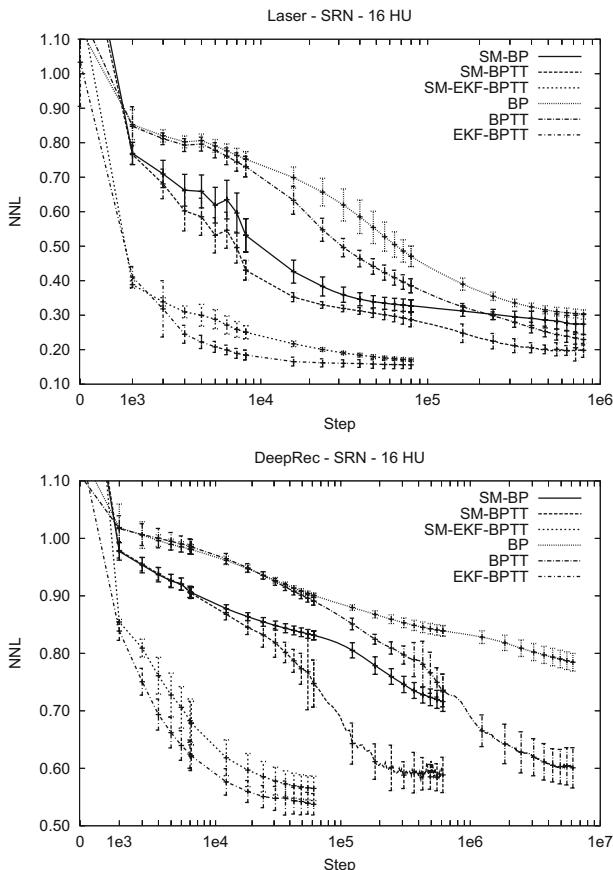


Fig. 1. Performance of Elman's SRN with 16 hidden units trained on Laser and DeepRec dataset

and BPTT algorithms, which seems to be sensitive to get stuck in local minima or to diverge. EKF approach to training RNNs on symbolic sequences shows higher robustness and better resulting performance. BP algorithm is too weak to give satisfactory results. NNL performances are significantly worse in comparing with algorithms that take into account the recurrent nature of network architectures. SMNNL version of BP and BPTT algorithms converge faster and finally better NNL error is usually achieved in comparing with standard BP pr BPTT. On the other hand SMNNL version of EKF shows slightly inferior NNL results in comparing with EKF-BPTT.

For linguistic datasets we provide results of SRN trained by the proposed approximation to the Kalman filtration - SM-EKF-BPTT. Computational requirements of standard EKF RNN training are very high because of high number of output units. Proposed approximation SM-EKF-BPTT helped us to obtain results in reasonable time.

The Table II summarizes results obtained by training SRNs using SM-BP, SM-BPTT and SM-EKF-BPTT on linguistic datasets. We also provide results using variable length Markov models (VLMMS), since RNNs and VLMMS are similar [12]. Results reported

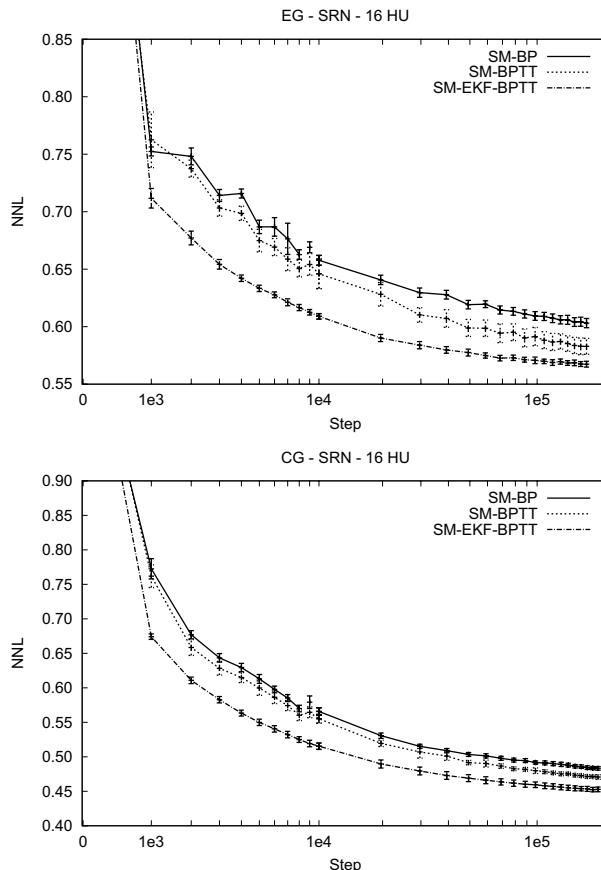


Fig. 2. Performance of Elman's SRN with 16 hidden units trained on EG dataset and CG dataset

Table 1. Summary of achieved results on linguistic datasets

	EG Dataset			CG Dataset		
	NNL	KLD	COS	NNL	KLD	COS
VLMM	0.63	0.097	0.91	0.51	0.114	0.86
SM-BP	0.60	0.070	0.93	0.48	0.084	0.90
SM-BPTT	0.58	0.046	0.96	0.47	0.072	0.92
SM-EKF-BPTT	0.56	0.031	0.96	0.45	0.053	0.93

here are not directly comparable to [10,11], nevertheless we have also conducted experiments with modifications such as limiting maximal sequence length to 11 words for EG dataset, using 10 datasets and removing duplicated sentences from test sets for CG dataset and better results than reported were achieved using SM-EKF-BPTT algorithm.

4 Conclusion

Significantly better results can be achieved by algorithms based on the Kalman filtration. Extended Kalman filter shows much faster convergence in terms of number of epochs and resulting performance is better. Results obtained with architectures with soft-max combining function and gradient-based algorithms minimizing the entropy proved to be better alternative than their standard counterparts based on quadratic error cost function. Faster convergence and better resulting performance was obtained. SM-EKF-BPTT did not achieve the same performance as standard EKF, the resulting NNL errors were slightly higher. On the other side the computational requirements of this algorithm are very favorable, comparable to the standard BPTT algorithm.

References

1. Elman, J.L.: Finding structure in time. *Cognitive Science* 14(2), 179–211 (1990)
2. Christiansen, M., Chater, N.: Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science* 23, 417–437 (1999)
3. Rodriguez, P.: Simple recurrent networks learn context-free and context-sensitive languages by counting. *Neural Computation* 13, 2093–2118 (2001)
4. Bodén, M., Wiles, J.: On learning context free and context sensitive languages. *IEEE Transactions on Neural Networks* 13(2), 491–493 (2002)
5. Hochreiter, J., Schmidhuber, J.: Long short term memory. *Neural Computation* 9(8), 1735–1780 (1997)
6. Werbos, P.: Backpropagation through time; what it does and how to do it. *Proceedings of the IEEE* 78, 1550–1560 (1990)
7. Prokhorov, D.V.: Kalman filter training of neural networks: Methodology and applications. In: *Tutorial on IJCNN 2004*, Budapest, Hungary (2004)
8. Tiňo, P., Dorffner, G.: Predicting the future of discrete sequences from fractal representations of the past. *Machine Learning* 45(2), 187–218 (2001)
9. Elman, J.: Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning* 7, 195–225 (1991)
10. Tong, M.H., Bickett, A.D., Christiansen, E.M., Cottrell, G.W.: Learning grammatical structure with Echo State Networks 20, 424–432 (2007)
11. Farkaš, I., Crocker, M.: Recurrent networks and natural language: exploiting self-organization. In: *Proceedings of the 28th Cognitive Science Conference*, Vancouver, Canada, pp. 1275–1280 (2006)
12. Tiňo, P., Čerňanský, M., Beňušková, Ľ.: Markovian architectural bias of recurrent neural networks. *IEEE Transactions on Neural Networks* 15(1), 6–15 (2004)

Computational Modeling of Risk–Related Eye Movement of Car Drivers

Masayoshi Sato¹, Yuki Togashi¹, Takashi Omori², Koichiro Yamauchi¹,
Satoru Ishikawa³, and Toshihiro Wakita⁴

¹ Graduate School of Information Science and Technology, Hokkaido University
`{justice,y_togashi,yamauchi}@complex.eng.hokudai.ac.jp`

² College of Engineering Tamagawa University

`omori@lab.tamagawa.ac.jp`

³ Hokusei Gakuen University

`ishi_s@hokusei.ac.jp`

⁴ Toyota Central R&D Labs., Inc.

`wakita@mosk.tytlabs.co.jp`

Abstract. A computational model of a car driver’s cognitive process was developed to create more useful assistance technology for making driving safer. The model’s status is mainly determined by the driver’s eye movement. In this model, we defined two types of risk : explicit risk for visible objects and inexplicit risk for non-visible areas. In our simulation, we attempted to reconstruct the driver’s eye movement while driving using our model.

1 Introduction

Many kinds of driving assist systems are being developed and innovated as a part of an intelligent transport system (ITS) for improving automobile safety. To effectively make driving safer, it is important that the system assist drivers based on their own cognitive processes[1][2][3]. To achieve this, we need a model of the human cognitive process for driving an automobile.

Many of models for human driving behaviors have been proposed[4][5][6]. However, no one model has been able to duplicate all human driving situations because there is no way to directly measure the cognitive processes involved. In this study, we focused on eye movement, which constitutes major part of information gathering while driving a car, and established a cognitive model of a car driver based on these eye movements during driving. In our experiments, we showed that the reconstructed eye movements from the cognitive model were similar to those of actual risk-related eye movements of human drivers.

2 Eye Movement Model

We present a broad overview of the proposed model in Fig. 1. First, the visual perception system builds a distribution calculation system based on the images

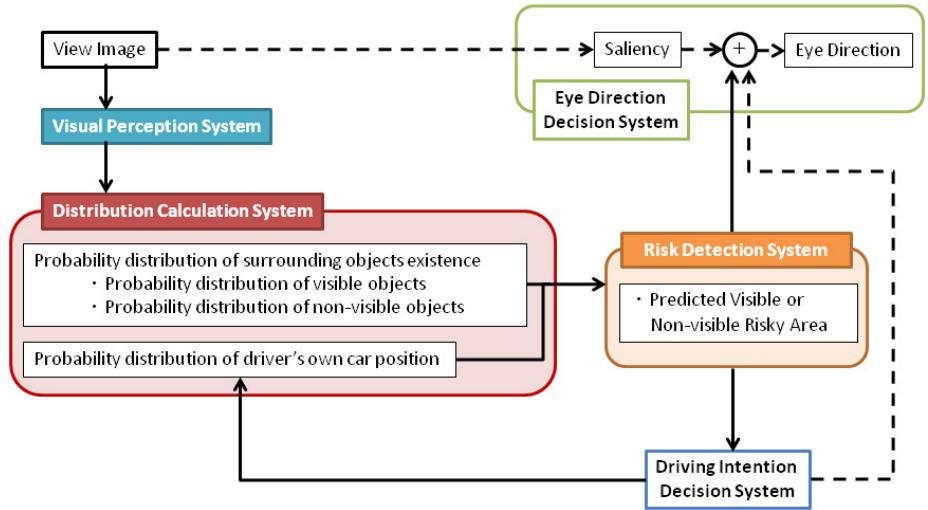


Fig. 1. Overview of Proposed Model

viewed. The distribution calculation system includes probability distributions of visible and non-visible objects. Specifically, the driver derives a probability distribution of visible objects for visible area such as pedestrians and oncoming cars according to the laws of physics, and probability distribution of non-visible objects for non-visible areas such as blind intersections and backward of obstacles. Then, the driver detects risk by estimating the overlapping area between one of those two distributions and a probability distribution of his own car's position which is based on the driver's intention. We call those overlapping areas the predictive visible risky area (PVRA) and predictive non-visible risky area (PNRA), respectively (Fig. 2). The driver tries to see the highly risky areas to predict the precise course of any approaching the visible or non-visible objects [7]. As a result of this observation, he changes his course to avoid an accident if he judges it dangerous to go ahead without doing so.

In sections 2.2 and 2.3, we explain in detail the probability distributions of visible and non-visible objects.

2.1 Probability Distribution of Visible Objects

To calculate the PVRA, the position of visible objects several seconds in the future is predicted using a Kalman filter [8]. The predicted position is represented by a normal distribution that we call the probability distribution of visual objects.

Moreover, several sets of such position predictors are used to allow for various possible approaching courses of the visible objects. In this study, we used three Kalman filters – “go straight”, “right-turn”, and “left-turn” –, simultaneously. The weighted sum of these three predicted distributions was used to calculate

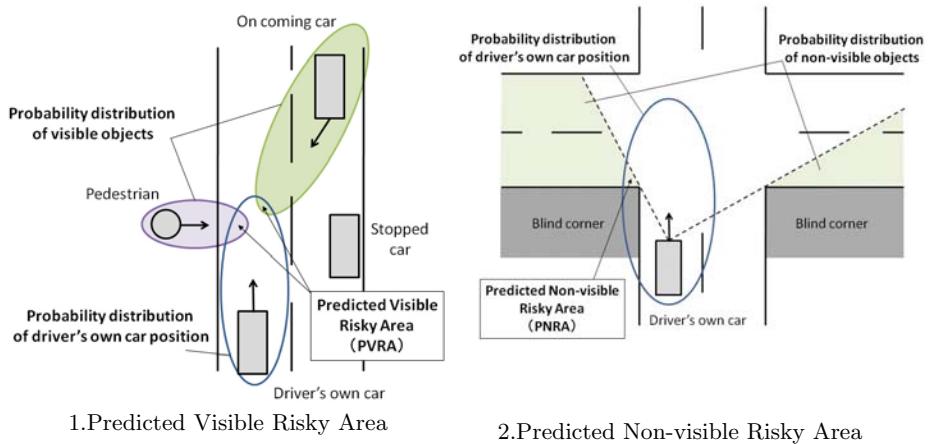


Fig. 2. Predicted Visible Risky Area (1) and Predicted Non-visible Risky Area (2)

the predicted distribution. The variance of each normal distribution accounts for the ambiguity of prediction. The weights for the three distributions were determined by the error between each predicted position and the actual position of the visible object. For example, the "go straight" Kalman filter is expressed as below.

$$\mathbf{K}(t) \leftarrow \hat{\Sigma}(t) \left(\hat{\Sigma}(t) + \Sigma_v \right)^{-1}, \quad (1)$$

$$\hat{\mathbf{x}}(t) \leftarrow \hat{\mathbf{x}}(t) + \mathbf{K}(t) \{ \mathbf{y}(t) - \hat{\mathbf{x}}(t) \}, \quad (2)$$

$$\hat{\Sigma}(t) \leftarrow \hat{\Sigma}(t) - \mathbf{K}(t) \hat{\Sigma}(t), \quad (3)$$

$$\hat{\mathbf{x}}(t+1) \leftarrow \mathbf{F}_{\text{go straight}} \hat{\mathbf{x}}(t), \quad (4)$$

$$\hat{\Sigma}(t+1) \leftarrow \mathbf{F}_{\text{go straight}} \hat{\Sigma}(t) \mathbf{F}_{\text{go straight}}^T + \mathbf{G} \Sigma_w \mathbf{G}^T. \quad (5)$$

Where $\mathbf{x}(t)$, $\mathbf{K}(t)$ and $\hat{\Sigma}(t)$ denote inner states consisting of velocity, position of the object, Kalman Gain at time t and variance-covariance matrix of the inner states, respectively. $\mathbf{F}_{\text{go straight}}$ denotes the state transition matrix of "go straight", \mathbf{G} denotes white noise, and $\hat{\Sigma}_w$ denotes a variance-covariance matrix of error. Kalman Gain is updated by Eq. (1). The inner state and variance-covariance matrix of error are updated using the observed target information $\mathbf{y}(t)$ by Eq. (2) and (3), respectively. Furthermore, the inner state and variance-covariance matrix at time $t+1$ are predicted by Eq. (4) and (5), respectively.

2.2 Probability Distribution of Non-visible Objects

The probability distribution of non-visible objects distributes empirically obtained prior probability adjusted by the driver's visual field and blind area, which represents an assumption that something exists at a certain probability in blind areas. The distribution is expressed as a maximum of these adjusted prior probabilities. Here, we denote $p(\text{Object}|x)$ as the prior probability that the Object exists, where x denotes the position of the object.

Adjusting the Prior Probability Based on the Driver's Visual Field

By using an approximate perceptive determination formula that expresses the effective range of the driver's observation, we have to consider the effect of the driver's point of gaze and peripheral field [9].

Figure 3 shows an example of driver's gaze and peripheral field. Eq. (6) expresses the approximated apparent size of the object which is separated from the driver by $R[m]$, where s denotes deflection from the central fovea of the image view on the retina.

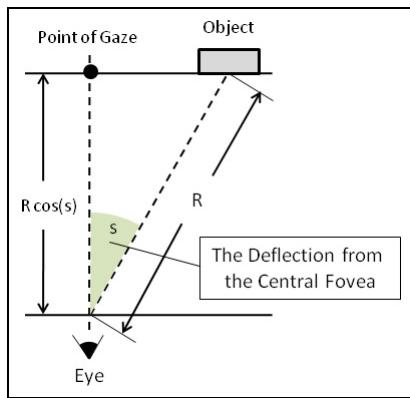


Fig. 3. Effective range of the driver's observation

$$\text{visible}(s, R) = \frac{R \cos(s)}{5} x(s), \quad (6)$$

$$x(s) = 5 \left(\tan\left\{\frac{\pi}{180}\left(x + \frac{a(s)}{2}\right)\right\} - \tan\left\{\frac{\pi}{180}\left(x - \frac{a(s)}{2}\right)\right\} \right), \quad (7)$$

$$a(s) = 0.03 \exp(0.07s). \quad (8)$$

Therefore, the adjusted prior probability is

$$p_{\text{AdjustedVisualField}}(\text{Object}|x) = \{1 - \exp(-\text{visible}(s, R))\} p(\text{Object}|x). \quad (9)$$

Equation (9) shows that the prior probability in the observable area is almost the same as $p(\text{Object}|x)$ but is decreased by the effect of the unobservable area.

Adjusting the Prior Probability Based on the Driver's Blind Area

We also have to consider the effect of the blind area. Therefore, we should assume that there is an object in any blind area. An example of this is depicted in Fig. 4. In this figure, we represent the obstacle's relative position x in a relative coordinate system between the position of the driver's own car ($x_0 = 0$) and that of the obstacle (x_{Obstacle}). The sigmoid function used in this study is

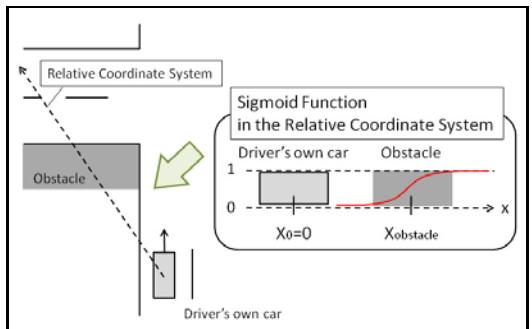


Fig. 4. Effective range of the driver's observation

$$\frac{1}{1 - \exp(-(x - x_{Obstacle}))}. \quad (10)$$

Where $x_{Obstacle}$ denotes the position of the obstacle. Then the adjusted prior probability is.

$$p_{AdjustedBlindArea}(Object|x) = \frac{1}{1 - \exp(-(x - x_{Obstacle}))} p(Object|x). \quad (11)$$

Equation (11) shows that if $x < x_{obstacle}$, the $p_{adjustedBlindarea}(Object|x)$ is reduced. On the other hand, if $x > x_{Obstacle}$, the prior probability is closed to $p(Object|x)$.

2.3 Risk Detection Mechanism

Risk detection is done by detecting the overlapped area of the probability distribution of objects and the probability distribution of the driver's own car position. First of all, the driver's own car position several seconds in the future is predicted using a Kalman filter, as far calculating the probability distribution of visible objects.

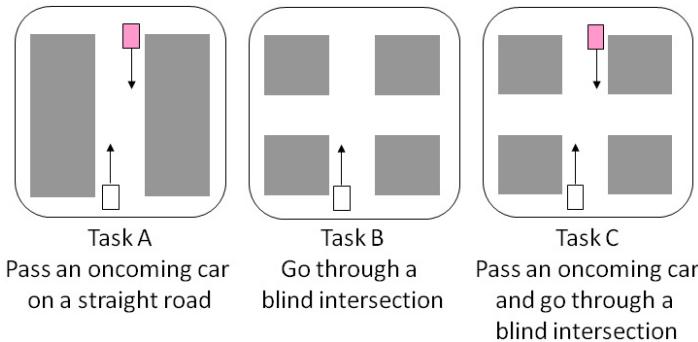
Then, the overlapping area of the probability distribution of driver's own car position and probability distribution of visible and non-visible objects" is detected. Those areas are called the predictive visible risky area (PVRA) and predictive non-visible risky area (PNRA), respectively.

The overlapped area is calculated by multiplying the predicted probability distribution of non-visible objects and that of the driver's own car's position. Therefore, if the multiplied value is larger than a threshold value, a target area with a large predictive visible or non-visible risky area is regarded as a dangerous area. In that case, the driver tries to look at the object which is the cause of dangerous area to obtain more precise information. As a result, the variance of the normal distribution for the target's existing probability distribution decreases and the PVRA disappears. When, the driver determines he is safe, he turns back to his original line of sight.

3 Experiment

3.1 Overview

To verify the validity of our proposed model, we compared reconstructed eye movements from our model with the actual eye movements of a human driver by using a driving simulator. The driving simulator described 3D space with VRML and used MATLAB's Virtual Reality Toolbox for drawing and controlling the 3D environment. There were eight male subjects and we projected three driving scenes on a screen in front of the subjects and tracked their eye movements. Then we observed the eye movements of our model for the same scenes.

**Fig. 5.** Driving scene task

3.2 Driving Simulation Procedure

We set up three driving scene tasks. The driver's own car did not crash into oncoming cars in any of the tasks, and both driver's own car and oncoming car run at 40 kilometers per hour.

Task A. Pass an oncoming car, represented by a red box, on a straight road (Fig. 5-A).

Task B. Go through a blind intersection with no oncoming car (Fig. 5-B).

Task C. Pass an oncoming car and go through a blind intersection (Fig. 5-C).

Note that tasks A, B, and C represent a PVRA, PNVA, and a predictive visible and non-visible risky area, respectively. We presented these driving scenes to the subjects and tracked their eye movements. Figure 6(A) is an example of the driving scenes we presented to the subjects. Here, the size of the scene image was 90×73 cm. The white dot represents the driver's actual eye direction. Figure 6(B) shows the behavior of our model for the same three driving scenes. The white dot represents the eye direction generated by our model. We can see that the actual and model-based eye directions are very similar to each other.

3.3 Results and Discussion

Detailed results of the experiments are shown in Fig. 7. The horizontal axis of each graph corresponds to the horizontal coordinate of the projected driving scene image, and the vertical axis corresponds to time from the beginning. Red, green, and black lines denote the average eye direction of the eight subjects when they looked at the oncoming car, the average eye direction of the eight subjects when they were not looking at the oncoming car, and the eye direction from our model, respectively.

To begin with, we discuss the eye movement of the subjects. In Fig. 7A, the subjects' eye direction as the oncoming car approached (red solid line) was deflected to the right. Then, the oncoming car disappeared from their sight

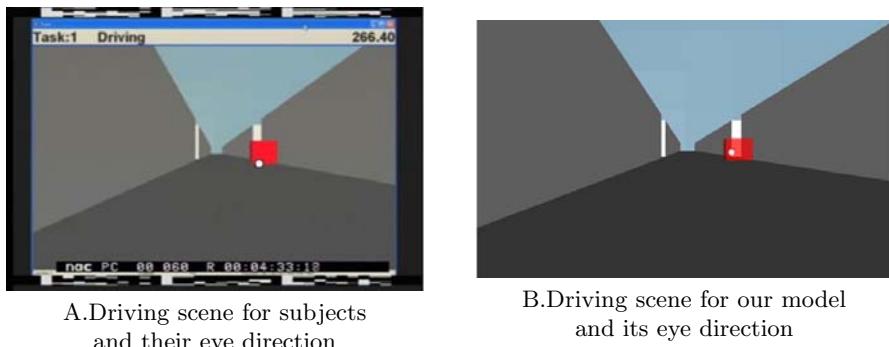


Fig. 6. An example of driving scene and eye direction of an actual driver (A) and predicted eye direction by our model (B)

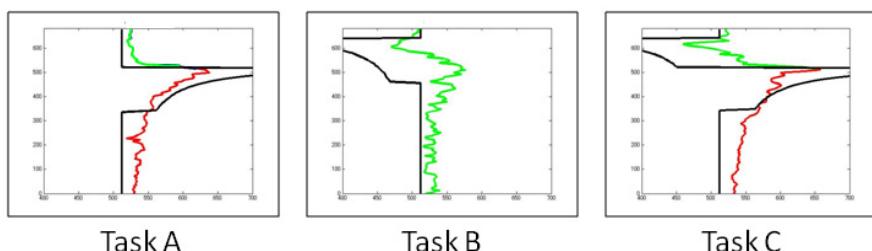


Fig. 7. Averaged eye direction for each task. colored lines denote the average eye direction of the eight subjects and black lines denote the eye direction from our model.

(green solid line), the eye direction returned to the center of the screen. This behavior is thought to be a response to the PVRA.

In Fig. 7B, as the drivers approached to the blind intersection, their eye direction deflected to the right, and then moved to the left immediately after passing through the intersection. This is because subjects looked at the intersection from right to left. This eye movement was thought to be a response to the PNRA.

Figure 7C shows that at first the subjects looked at the oncoming car (red solid line), and, after the car passed by, they looked at the left corner of the intersection (green solid line).

We compared the eye movement of the subjects with those of our model. In Tasks A and C, the timing of the eye movement of the subjects is almost identical to that of our model. So, our eye movement model reconstructed the subjects' behavior well. In Task B, although our model successfully reconstructed the eye movement for the left corner, it failed to reconstruct it for the right corner.

4 Conclusion

In this paper, we proposed a computational model for eye movement while driving a car. In our model, probability distributions of surrounding objects' existence and a probability distribution of a driver's own car position were computed

by predicting the behavior of objects using Kalman filters and prior probability distributions. Then, two types of risky area, PVRA and PNRA were calculated using these probability distributions. Drivers moved their eyes toward risk areas to obtain more precise information.

In addition, we compared the reconstructed eye movements of the model and the corresponding actual eye movements of the human drivers. Our model could explain the eye movement for PVRA and for part of PNRA, but could not reconstruct the eye movement for the PNRA of the right corner of the intersection.

We plan to improve our model to overcome the above drawback. In the near future, we plan to establish a cognitive model that can reconstruct actual human behavior with high accuracy to contribute to the development of a safely motorized society.

References

1. Kentaro, M., Genki, S., Takashi, O., Akitoshi, O.: A Feasibility Study of Cognitive Computation Model for Driver's Process Estimation from Driving Behavior. IEICE technical report. Neurocomputing 103(734), 25–30 (20040312)
2. Mizutani, K., Omori, T.: A Feasibility Study of Driver's Cognitive Process Estimation from Driving Behavior. IEEJ. Trans. EIS 125(6), 967–975 (2005)
3. Togashi, Y., Omori, T., Yamauchi, K.: Model of driver eye motion based on driving plan and prediction of changes in the environment. In: Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence (FOCI 2007), pp. 289–295 (2007)
4. Al-Shihabi, T., Mourant, R.: A Framework For Modeling Human-Like Driving Behavior For Autonomous Vehicles in Driving Simulators. In: Proc. The Fifth International Conference on Autonomous Agents, Montreal, Canada, pp. 286–291 (2001)
5. Salvucci, D.D.: Modeling Driver Behaviro in a Cognitive Architecture. Human Factors 48(2), 362–380 (Summer 2006)
6. Salvucci, D.D., Mandalia, K., Yamamura: Lane-change detection using a computational driver model. Human Factors 49(3), 532–542 (2007)
7. Sprague, N., Ballard, D.: Eye Movements for Reward Maximization. In: Advances in Neural Information Processing Systems 16 (2004)
8. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problem. Transactions of American Society of Mechanical Engineers, Journal of Basic Engineering, 82D, 34–45 (1960)
9. Suzuki, T.: The text of neuronal physiology,
http://www.tmd.ac.jp/med/phy1/ptext/ptext_index.html

Robust Detection of Medial-Axis by Onset Synchronization of Border-Ownership Selective Cells and Shape Reconstruction from Its Medial-Axis

Yasuhiro Hatori and Ko Sakai

Graduate School of Systems and Information Engineering, University of Tsukuba,
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573 Japan
hatori@cvs.cs.tsukuba.ac.jp, sakai@cs.tsukuba.ac.jp
<http://www.cvs.cs.tsukuba.ac.jp/>

Abstract. There is little understanding on representation and reconstruction of object shape in the cortex. Physiological studies with macaque suggested that neurons in V1 respond to Medial-Axis (MA). We investigated whether (1) early visual areas could provide basis for MA representation, and (2) we could reconstruct the original shape from its MA, with a physiologically realistic computational model consisting of early- to intermediate-level visual areas. Assuming the synchronization of border-ownership selective cells at stimulus onset, our model was capable of detecting MA, indicating that early visual area could provide basis for MA representation. Furthermore, we propose a physiologically plausible reconstruction algorithm with the summation of distinct gaussians.

1 Introduction

Visual cortical areas along the ventral pathway appear to play a key role in shape representation. Neurons in the primary visual cortex (V1) are selective to simple features such as orientation and spatial frequency [1], and those in intermediate-level visual cortex (V2, V4) respond selectively to more complex shape features such as arc, curvature and simple shapes [2, 3]. On a population level, neurons in inferior temporal (IT) cortex represent abstract shapes with the intermediate-level complexity [4]. However, it is still largely unknown that how the population of neurons in early- to intermediate-level visual areas contributes to the representation of shape.

Figure/ground segregation might be the first step toward shape perception and recognition. Recent physiological experiments in macaque have reported that 60% of neurons in V2 and V4 are selective to Border-Ownership (BO) that tells which side of region owns the border [5]. However, only 20% of neurons in V1 showed such selectivity [5]. Another physiological experiment in macaque has reported that neurons in V1 represent an object shape by Medial-Axis (MA) representation [6]. MA is one of the most robust shape descriptor that represents shape by a set of inscribed circles whose are equidistance from its contour. An arbitrary shape would be reconstructed from the centers of circles and their diameters. In this paper, we investigate whether neural circuits in early- to intermediate-level visual areas could

provide a basis for MA representation, and whether we could reconstruct the original shape from its MA representation.

We propose that onset synchronization of BO selective cells is crucial for MA representation. A physiological study [6] has showed that V1 neurons responded to figure shape around 40 ms after the stimulus onset, while the latency of response to MA was about 100 ms after the onset. Physiological study on BO selective neurons reported that their latency is around 70 ms [5]. Moreover, in the cat visual cortex, stimulus onset is known to play an important role for synchronization of cell assembly [7, 8]. These results give rise to the possibility that neurons detect contours and BO, and then cells located around the object center respond to MA from the BO signals propagated from its contours. If a neuron is located equidistance to the nearby contours, the cell will receive signals from BO cells at the same time so that it responds rigorously to represent MA.

To examine this proposal, we have constructed the computational model including V1 and V2 layer [9]. Although this model showed certain agreement with the physiological result of Lee et al [6], its physiological reality was questionable, for example, it did not realize conduction velocity, anatomical distance between V1 and V2, size of classical receptive field (cRF), morphology of cells. Here, we construct, based on the previous model, a physiologically realistic computational model that overcomes its drawbacks, and carry out simulations under the assumption that BO-selective cells are synchronized at stimulus onset. Our results indicate that simultaneous firing of BO selective cells is essence for the MA representation.

We also provide shape reconstruction algorithm from the detected MA. In our model, each V1 cell has distinct excitatory integration field that is defined by a gaussian. With the summation of distinct gaussians, we obtain the reconstruction of arbitrary shape. For quantitative evaluation, we computed reconstruction error from the reconstructed image and the original image.

2 The Proposed Model: MA Detection

The activity of model MA cell is determined from four stages: (1) contrast detection, (2) BO determination, (3) MA detection, and (4) integration of the activity of MA cells. A schematic diagram of the model is given in Fig. 1 (left). These stages mimic early visual areas such as V1 and V2, as similar to the previous model [9]. In addition, the present model has taken into account fundamental characteristics of physiology and anatomy such as synaptic conduction velocity, and spatial organization of early visual areas.

2.1 Model Neurons

We implemented single-component firing neurons and their connections through their synapses on NEURON simulator [11], to simulate the precise spatiotemporal properties of model cell. The soma of the model neuron is approximated by a homogeneous sphere because there is no physiological evidence in morphology of MA cell and BO selective cell. We set the radius and the membrane resistance of sphere to $23\mu\text{m}$ and $230\Omega\text{cm}$, respectively. Each model neuron calculates its

membrane potential following the Hodgkin-Huxley equation [12] with constant parameter as shown in the right panel of Fig. 1 [13, 14].

2.2 Contrast Detection and Integration

The model cells in this stage mimic the response properties of V1 cells, including the contrast detection, dynamic contrast normalization and static compressive nonlinearity. This stage is identical in essence to [9].

The model cells in this stage detect the luminance contrast from oriented Gabor filters with distinct four orientations. We take a convolution of an input image with the Gabor filters, with dynamic contrast normalization [15] including a static, compressive nonlinear function.

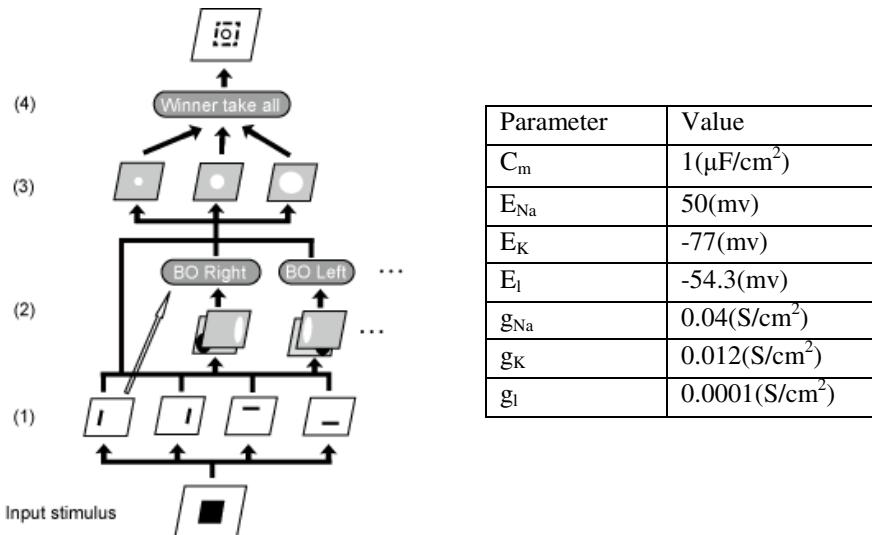


Fig. 1. An illustration of the proposed model (left). The activity of model MA cell is computed through four stages: (1) contrast detection, (2) BO determination, (3) MA detection, and (4) integration of the activity of MA cells. The constant values for the model cells used in the simulations (right).

2.3 Determination of the Activity of BO Selective Cells

The model cells in the second stage mimic BO-selective neurons, which determine the direction of BO with respect to the border at the CRF based on the modulation from surrounding contrast up to 5 deg in visual angle from the CRF. The determination of BO from surround modulation is known to show good agreement with physiological result [10]. Each model BO cell has asymmetric excitatory and inhibitory region with respect to the CRF to modulate its activity. In our model, we realize surround modulation by excitatory and inhibitory synaptic connection (EPSP or IPSP) from surrounding cells. Each EPSP is calculated by:

$$E(x, y, t - del) = w(v - e) \{ \exp(-(t - del)/\tau_{decay}) - \exp(-(t - del)/\tau_{rise}) \}, \quad (1)$$

where x and y represent spatial positions, t represents time, del represents the conduction delay, w represents the weight defined by gauss function, v and e represent the membrane and the reversal potentials, respectively, τ_{decay} and τ_{rise} represent the time constants of decay and rise, respectively. The activity of a BO cell at time t ($O^2(x_1, y_1, t)$) is determined by:

$$O^2(x_1, y_1, t) = input(x_1, y_1) + c \sum_{x,y} \{ E(x, y, t - del) + I(x, y, t - del) \}, \quad (2)$$

where $input$ shows the output of the first stage, c is the weight for surround modulation, E and I represent EPSP and IPSP, respectively. We realized IPSP by setting the w in equation (1) to a negative value. We set the value of c , del , τ_{rise} , τ_{decay} , e to 0.06, 10ms, 0.09ms, 1.5ms, 0mv, respectively.

2.4 Determination of the Activity of MA Cells

The third stage pools spatially the responses from the second stage to test the MA representation from simultaneous firing of BO cells. Each model MA cell has distinct integration fields defined by three types of gaussians to integrate pulses from BO cells through feedback connections and V1 cells through horizontal connections. For the sake of simplicity, we use three scales of gaussian ($\sigma = 1, 3, 5$). The activity of a MA cell is calculated by:

$$O_\sigma^2(x_2, y_2, t) = \sum_{x,y} \{ F_\sigma(t, D(x_2, y_2, x, y, V2)) + H(t, D(x_2, y_2, x, y, V1)) \}, \quad (3)$$

where x_2 and y_2 represent the spatial positions of the MA cell, F_σ represents EPSP from BO cells through the feedback connections, H represents EPSP from V1 cells responding to contrast. F_σ and H are defined as equations (4) and (5), respectively:

$$F_\sigma(t, D) = w_{feedback} E(x, y, t - D), \quad (4)$$

$$H(t, D) = w_{horizontal} E(x, y, t - D), \quad (5)$$

where $w_{feedback}$ and $w_{horizontal}$ represent the weight for feedback connection and horizontal connection, respectively. We set $w_{feedback}$ three times larger than $w_{horizontal}$. D represents the delay depending on type of connections (feedback or horizontal) and spatial length that is defined as:

$$D(x_2, y_2, x, y, L) = \sqrt{(x_2 - x)^2 + (y_2 - y)^2 + d_{V1,L}^2} / v, \quad (6)$$

where L represents the layer (V1 or V2), $d_{V1,L}$ represents the distance between V1 and V1 or V2. $d_{V1,L}$ is set to 0mm if L is V1, and 30mm if V2. We set the conduction velocity, v , to 3mm/ms for feedback connection and 0.1mm/ms for horizontal connection.

2.5 Integration of MA Cell Activities

The final output of the model from three types of MA cells with distinct effective size. We use the winner-take all method [16] that choose the maximum response among $O^3_\sigma(x_2, y_2)$.

3 Results: MA Detection

We carried out simulations of the model with several types of stimuli including natural images in order to test our hypothesis. The results for three types of stimuli, a square, a rectangle and L-shaped tree branch as examples are shown in Figs.2, 3, and 4, respectively. In Fig. 2, we observe clear MA of the square with a clear peak at MA in 2D plot along horizontal middle (Fig. 2, right), as similar to the physiological report by Lee, et al. [6]. In Fig. 3, we observe MA of rectangle as similar to the square. In Fig. 4, we observe the strong response of cells located along the MA, although the shape is not regular as taken from a natural scene, indicating that our model has the ability for robust detection of MA. These results suggest that onset synchronization of BO cells is capable of generating MA without any other particular constraints.

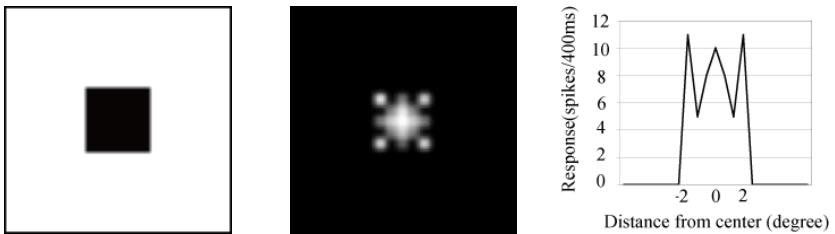


Fig. 2. MA detection with a single square as a stimulus (left). Result of the model with brighter gray showing higher activity (center). For presentation purpose, responses to the edges are not shown. 2D plot of responses of cells along the horizontal midline (right).

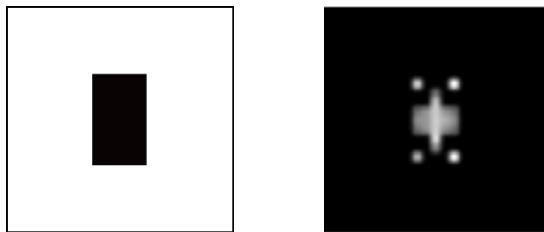


Fig. 3. MA detection with a rectangle. The conventions are the same as Fig. 2.

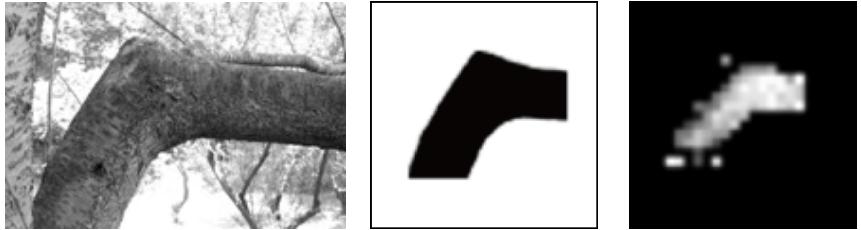


Fig. 4. MA detection with a bended tree from a natural scene. Original picture (left). Binarized image that is segmented from the original picture (center). Simulation result (right). The conventions are the same as Fig. 2.

4 Shape Reconstruction

We propose a physiologically plausible algorithm to reconstruct shape from the responses of the MA detection model. A schematic illustration of the proposed algorithm is shown in Fig. 5.

4.1 The Algorithm

We assign two matrices: MA matrix (M) and Gaussian matrix (G). Each element of M is the number of spikes computed by MA detection model. Each element of G is the standard deviation (SD) of a MA cell that responds most strongly at the given spatial position. G is defined as:

$$G(x, y) = \arg \max_{\sigma} (O_1^3(x, y), O_3^3(x, y), O_5^3(x, y)). \quad (7)$$

We calculated Reconstruction matrix (R) from M and G . R is defined as summation of gaussians:

$$R(x, y) = \sum_{(m,n) \in (M, N)} \sum_{\sigma} w_{\sigma} \times G(m, n) \times M(m, n) \times \text{gauss}(m, n, G(x, n)), \quad (8)$$

where m and n represent spatial positions, M and N represent the numbers of row and column of R , respectively, and gauss represents the SD of gaussian.

4.2 Reconstruction Error

For quantitative evaluation of the algorithm, we calculated the error of reconstruction. We defined reconstruction error as the difference between the original image and the reconstructed image:

$$\text{Error} = \frac{\sum_{x,y} [I(x, y) - R(x, y)]^2}{\sum_{x,y} I(x, y)^2}. \quad (9)$$

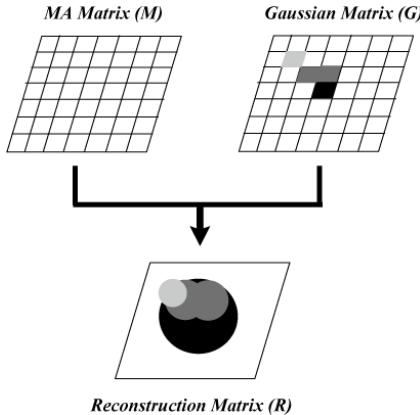


Fig. 5. A graphical representation of the proposed algorithm. MA matrix is the result of MA detection model. Each element of Gaussian matrix is the standard deviation of the MA cell that showed the strongest response at that point. Reconstruction matrix is calculated by summation of weighted gaussians.

5 Results: Shape Reconstruction

We reconstructed shapes from the detected MA shown in Figs. 2, 3 and 4. Reconstructed images are shown in Fig. 6. We observe a rough shape of each object. The reconstruction error of square, rectangle and L-tree were 0.1255, 0.4805 and 0.5397, respectively.

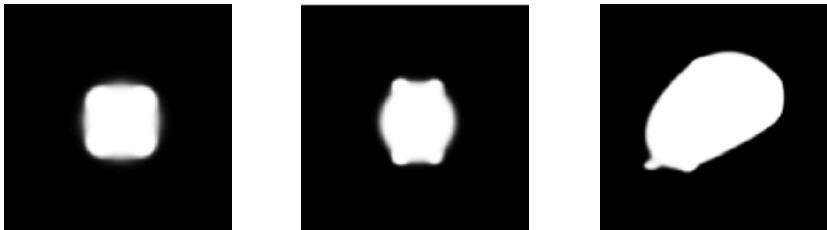


Fig. 6. Result of shape reconstruction. Reconstructed square (left), rectangle (center), L-tree (right). Rough shape of each object is reconstructed.

6 Conclusions

We proposed that onset synchronization of BO selective cells is crucial for MA representation. To examine the hypothesis, we constructed a physiologically realistic computational model including conduction velocity of synaptic connection, anatomical distance between V1 and V2, size of cRF, but without morphology of cell. We carried out simulations with several types of stimuli. With the onset synchronization of BO selective cells, the proposed model provided MA representation by V1, showing good

agreements with the results of Lee et al [6]. Even if an input stimulus was complicated such as L-tree, accuracy of MA detection was not decreased. These results showed that potential ability to detect MA from any shape including natural object, indicating that robustness of MA detection by our model. Our simulation results showed plausibility of our proposal.

Furthermore, we examined whether object shape is reconstructed from the detected MA. To answer this question, we proposed the algorithm that uses the response of MA cells and its SD of gaussian. With summation of gaussians, we observed the certain recovery of original shape.

These results suggest that neural circuit including early- to intermediate level visual area could provide a basis of MA representation with onset synchronization of BO selective cells. Furthermore, arbitrary shape could be reconstructed from its detected MA representation.

We and Lee et al demonstrated that physiological plausibility of MA representation, however, biological plausibility of MA representation for shape recognition is not demonstrated. It has been thought that IT cells which respond to spatial arrangement of surfaces are involved in shape recognition [17]. We should focus on whether MA representation is useful for shape representation in IT cells. It might provide the further understanding for shape recognition.

Acknowledgment

This work was supported by Grant-in-aid for Scientific Research from the Brain Science Foundation, the Okawa Foundation, JSPS (KAKENHI 19530648), and MEXT of Japan (KAKENHI, Joho Bakuhatsu, 19024011).

References

1. Hubel, D.H., Wiesel, T.N.: Receptive Fields and Functional Architecture of Monkey Striate Cortex. *The Journal of Physiology* 195, 215–243 (1968)
2. Pasupathy, A., Connor, C.E.: Shape representation in Area V4: Position-Specific Tuning for Boundary Conformation. *The Journal of Neurophysiology* 86, 2505–2519 (2001)
3. Hegde, J., Van Essen, D.C.: A Comparative Study of Shape Representation in Macaque Visual Areas V2 and V4. *Cerebral Cortex* 17, 1100–1116 (2006)
4. Yamane, Y., Tsunoda, K., Matsumoto, M., Phillips, A.N., Tanifugi, M.: Representation of the Spatial Relationship among Object Parts by Neurons in Macaque Inferotemporal Cortex. *The Journal of Neurophysiology* 96, 3147–3156 (2006)
5. Zhou, H., Friedman, H.S., Heydt, R.: Coding of Border Ownership in Monkey Visual Cortex. *The Journal of Neuroscience* 20, 6594–6611 (2000)
6. Lee, T.S., Mumford, D., Romero, R., Lamme, V.A.F.: The role of the primary visual cortex in higher level vision. *Vision Research* 38, 2429–2454 (1998)
7. Samonds, J.M., Bonds, A.B.: Gamma Oscillation Maintains Stimulus Structure-Dependent Synchronization in Cat Visual Cortex. *The Journal of Neurophysiology* 93, 223–236 (2005)

8. Zhou, Z., Bernard, M.R., Bonds, A.B.: Deconstruction of Spatial Integrity in Visual Stimulus Detected by Modulation of Synchronized Activity in Cat Visual Cortex. *The Journal of Neuroscience* 28, 3759–3768 (2008)
9. Hatori, Y., Sakai, K.: Representation of Medial Axis from Synchronous Firing of Border-Ownership Selective Cells. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part I. LNCS, vol. 4984, pp. 18–26. Springer, Heidelberg (2008)
10. Sakai, K., Nishimura, H.: Surrounding Suppression and Facilitation in the determination of Border Ownership. *The Journal of Cognitive Neuroscience* 18, 562–579 (2006)
11. NEURON, <http://www.neuron.yale.edu/neuron/>
12. Koch, C. (ed.): Biophysics of Computation. Oxford University Press, Oxford (1999)
13. Archie, K.A., Mel, B.W.: A model for intradendritic computation of binocular disparity. *Nature* 3, 54–63 (2000)
14. Peters, A., Rockland, K.S. (eds.): CEREBRAL CORTEX. Primary Visual Cortex in Primates, vol. 10. Plenum Press (1994)
15. Carandini, M., Heeger, D.J., Movshon, J.A.: Linearity and Normalization in Simple Cells of the Macaque Primary Visual Cortex. *The Journal of Neuroscience* 17, 2796–2808 (1997)
16. Rolls, E.T., Deco, G. (eds.): Computational Neuroscience of Vision. Oxford University Press, Oxford (2002)
17. Yamane, Y., Carlson, E.T., Bowman, K.C., Wang, Z., Connor, C.E.: A neural code for three dimensional shape in macaque inferotemporal cortex. *Nature Neuroscience* 11, 1352–1360 (2008)

Synaptic Cooperation and Competition in STDP Learning Rule

Shigeru Kubota and Tatsuo Kitajima

Department of Bio-System Engineering, Yamagata University,
4-3-16 Jonan, Yonezawa, Yamagata, 992-8510, Japan
kubota@yz.yamagata-u.ac.jp

Abstract. The correlation-based rule of plasticity has been widely believed to be involved in the organization of functional synaptic circuits. However, recent studies have suggested that the direction of plasticity in the sensory-deprived barrel cortex can be reversed, depending on the stimulus environment, from that predicted by the correlation-based plasticity. To investigate whether spike-timing-dependent plasticity (STDP) may underlie such reversal in cortical plasticity, we study the influence of the correlation time on the synaptic cooperative and competitive mechanisms based on the input correlation. The results show that in the presence of activity-dependent feedback modification of the STDP window function, the increase in the correlation time can reverse the plasticity outcome such that for shorter correlation time, more frequently activated synapses are strengthened while, as the correlation time is sufficiently prolonged, less frequently activated synapses become strengthened.

1 Introduction

There is much evidence that the correlation-based rule of plasticity is followed in many brain systems [1, 2]. This plasticity rule predicts that the synapses that work together and therefore can evoke postsynaptic firing are strengthened. However, such mechanism of cooperation indicates that synapses that accidentally work together are strengthened and finally lead to saturation; therefore, for generating functional circuits, synaptic modifications must have a competitive side also. In the case that the competition is concurrent with the cooperation, synaptic modifications result in the outcome “synapses that work together are strengthened while others are weakened.” Such competitive interaction between the groups of cooperative inputs can explain the evidence that the deprivation of sensory inputs induces an expansion of adjacent receptor surface representations into the deprived cortical region [3, 4]. For example, when all but a single “spared” whisker is repeatedly plucked in the face of the rodent, the representation of the intact whisker in the barrel cortex expands to include regions of the cortex that are no longer activated by the deprived whiskers [5].

However, a recent experiment that examined map plasticity in the adult barrel cortex [6] has revealed that the direction of map plasticity does not necessarily obey the correlation-based rule but is reversed depending on the stimulus environment: when the rat remains in its home cage, the removal of all but a single whisker leads to an expansion of the spared whisker’s cortical representation. However, the same

manipulation results in a large-scale contraction of its representation if the animal is given a brief opportunity to use whiskers for active exploration of a new environment. The fact that changing the animal's environment may be sufficient to reverse the direction of cortical plasticity implies that this direction may be regulated by the characteristics inherent in the sensory stimuli. Similar use-dependent contraction of cortical whisker representation also can be found to occur in the absence of whisker deprivation [7].

Spike-timing-dependent plasticity (STDP), wherein the millisecond-scale timing of action potentials regulates plasticity [8-10], has been suggested to provide a physiological mechanism that induces synaptic competition [11]. However, for STDP to activate its competitive function, relatively precise tuning of the LTP/LTD balance is required such that the area under the STDP curve for the LTP portion becomes slightly smaller than that for the LTD portion [11]. Otherwise, all the synapses are potentiated when LTP exceeds LTD, whereas all the synapses are depressed when LTD dominates over LTP. However, it has been revealed that when STDP is combined with the activity-dependent feedback (ADFB) mechanism, in which LTP/LTD balance is dynamically modified depending on the postsynaptic activity, the overall effects of LTP and LTD are nearly balanced, introducing the competition among synapses [12, 13].

Therefore, in this study, we theoretically examine the cooperation and competition arising from the STDP model with the ADFB mechanism to address the issue of how the direction of cortical plasticity can be reversed depending on the characteristics of sensory stimuli [6, 7]. Through the numerical experiments using a biophysical conductance-based neuron model, we show that a longer correlation time can reverse the effects of synaptic cooperation and competition from those expected from the correlation-based rule.

2 Model

We used a conductance-based pyramidal neuron model consisting of the two compartments representing the soma and the dendritic region [13]. The neuron receives random inputs, generated by Poisson processes, from 4000 excitatory (AMPA and NMDA) and 800 inhibitory (GABA) synapses, with the strength of AMPA synapses modified by STDP [13]. Excitatory synapses were divided into 2 equally sized groups. We simulated the following 2 cases of excitatory inputs: (1) A case where the spike trains for inputs from one group are correlated, while those from the other group left uncorrelated (Fig. 2). (2) A case where correlations of equal magnitude are introduced into both groups (Figs. 3 and 4). In both cases, the correlated inputs were generated by the method given in [13, 14], wherein presynaptic firing rates have a correlation function that decays exponentially with a time constant τ_c (correlation time). Inhibitory synapses remain uncorrelated.

When the potentiation of presynaptic inputs enhances the total input activity, higher postsynaptic activity results in the increased intracellular Ca^{2+} level, which, in turn, induces the desensitization of NMDARs. This activity-dependent desensitization of NMDARs has been suggested to stabilize the learning dynamics of STDP [13]. Since the desensitization of NMDARs decreases not only its peak conductance

(g_{NMDA}) but also its decay time constant (τ_{decay}^{NMDA}), the ADFB modulation of the NMDAR activity can be described as follows [13]:

$$\tau_{decay}^{NMDA} = \tau_{decay,0}^{NMDA} - k_1 f_{post} \quad (1)$$

$$g_{NMDA} = g_{NMDA,0} - k_2 f_{post} \quad (2)$$

Here, both g_{NMDA} and τ_{decay}^{NMDA} are negatively regulated by the postsynaptic firing rate f_{post} with the gradients determined by the positive parameters k_1 and $k_2 \cdot g_{NMDA,0}$ and $\tau_{decay,0}^{NMDA}$ denote the NMDAR peak conductance and decay time constant for single NMDAR currents, respectively (i.e., at $f_{post} = 0$).

To incorporate the ADFB mechanism into the learning dynamics of STDP, we used a 4-dimensional STDP map, which is the same as the one used in our previous study [13]. This map is represented by $\Delta w(\Delta t, \tau_{decay}^{NMDA}, g_{NMDA})$ and gives the amount of synaptic weight change Δw corresponding to each pre- and postsynaptic spike pair as a function of its interspike interval (ISI) $\Delta t = t_{post} - t_{pre}$, and τ_{decay}^{NMDA} , and g_{NMDA} . This map was constructed from the simulation of the *in vitro* experiment of STDP (e.g., [9]) as described previously [13, 15]. Briefly, we used our previously proposed spine Ca²⁺-dependent plasticity (CaDP) model [15], which is based on the experimental findings that higher Ca²⁺ increase induces LTP, while a prolonged period of moderate Ca²⁺ increase is required for LTD induction [16, 17]. In this model, the plasticity Δw can be described as follows:

$$\Delta w = f_p([Ca]_{peak}) + f_D([Ca]_{peak}) \cdot f_B(T_{Ca} - \hat{T}([Ca]_{peak})), \quad (3)$$

where $[Ca]_{peak}$ is the peak Ca²⁺ amplitude, T_{Ca} is the time period during which the Ca²⁺ concentration is above a threshold. $f_p(x)$ and $f_D(x)$ are functions responsible for LTP and LTD induction, respectively; $f_p(x) > 0$ for $x > \sigma_p$ and $f_p(x) = 0$, otherwise, while $f_D(x) < 0$ for $x \in (\sigma_D, \sigma_p)$ and $f_D(x) = 0$, otherwise. σ_p and σ_D ($\sigma_D < \sigma_p$) are the threshold Ca²⁺ levels for the LTP and LTD induction, respectively. $f_B(x)$ is assumed to be the Heaviside step function. This function plays a role in blocking LTD induction when T_{Ca} is shorter than the threshold function $\hat{T}([Ca]_{peak})$ and are required to reproduce asymmetric time windows of STDP found in the neocortex. To simulate the pairing protocol, a presynaptic input was paired with a brief current injection into the soma. After calculating the Ca²⁺ time course in a dendritic spine for each ISI Δt , we applied the CaDP model (Eq. 3) and obtained the STDP curve (Fig. 1A). This simulation was performed with different values of τ_{decay}^{NMDA} and g_{NMDA} to obtain a sufficient number of map data (around 280,000 data points). Figure 1B illustrates a 3-dimensional section of the 4-dimensional STDP map.

Throughout all our simulations, the values of synaptic weights were initially set to their upper limit. When pre- or postsynaptic spike occurs, the synaptic weights were

updated stepwise by a small amount Δw given by the 4-dimensional STDP map $\Delta w(\Delta t, \tau_{decay}^{NMDA}, g_{NMDA})$, where the values of τ_{decay}^{NMDA} and g_{NMDA} were dynamically modified by the ADFB mechanism represented by Eqs. 1 and 2.

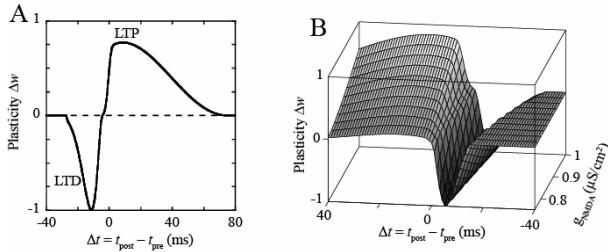


Fig. 1. (A) An example of the STDP window function obtained by the calcium-dependent plasticity model (Eq. 3). (B) A 3-dimensional section of the 4-dimensional STDP map.

3 Results

STDP tends to strengthen correlated inputs that are likely to arrive together within a short time period and contribute to rapidly evoking a postsynaptic spike [11, 14]. To investigate such mechanism of cooperation between correlated inputs in the presence of the ADFB modulation, we examined a case where the neuron receives inputs from 2 groups of synapses, consisting of a correlated and an uncorrelated group, which are activated by the inputs with the same input frequency (Fig. 2).

When the correlation time is relatively shorter ($\tau_c = 5$ ms), the synaptic weights of the correlated group accumulated near the upper limit at the equilibrium state, while those of the uncorrelated group was pushed toward 0, similar to a previous study [14] (Fig. 2A). However, when the correlation time is much more prolonged ($\tau_c = 10,000$ ms), the effects of input correlation was reversed such that the synapses of uncorrelated group accumulated near the upper limit, while those of the correlated group were depressed (Fig. 2B). This result was quantified by measuring the difference in the average weights between the 2 groups normalized by the maximum weight (w_{max}), denoted as $\delta w = (\bar{w}_{corr} - \bar{w}_{uncorr})/w_{max}$, where \bar{w}_{corr} and \bar{w}_{uncorr} are the average weights of the correlated and uncorrelated groups, respectively (Fig. 2C). The figure shows that the input correlation has either facilitatory or suppressive effect on the correlated inputs, depending on whether τ_c is less than or greater than a threshold value (~50 ms), respectively. The effect of a sufficiently longer correlation time ($\tau_c \sim 10000$ ms) to weaken the correlated inputs is around the same in strength as that of a sufficiently shorter correlation time ($\tau_c \sim 5$ ms) to potentiate the inputs (Fig. 2C). This finding strongly contrasts with the result of a previous STDP model [14], in which a longer correlation time either does not affect or slightly weakens the correlated input group as compared to uncorrelated group. Our result suggests that in the presence of

activity-dependent feedback modulation of STDP, a longer correlation time not only eliminates the cooperative effect of input correlation but also reverses this effect, inducing significant depression of a group of correlated inputs.

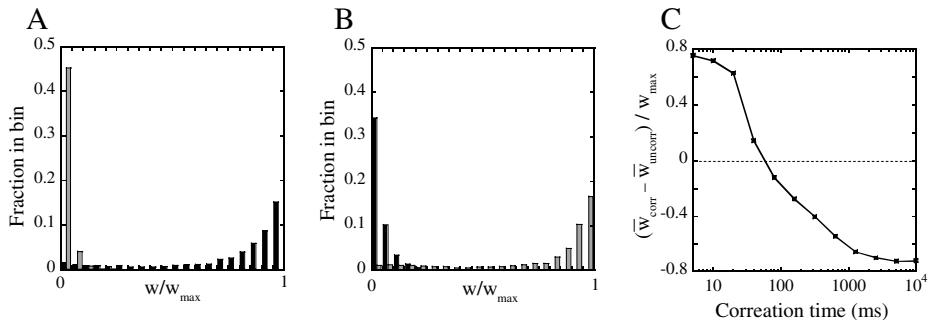


Fig. 2. The effect of changing correlation time (τ_c) on the synaptic distribution at the equilibrium state in the presence of a correlated and an uncorrelated input group. (A and B) The final weight distribution of the correlated (black) and uncorrelated (gray) groups when $\tau_c = 5$ ms (A) and 10,000 ms (B). (C) The difference between the average weights of the two groups as a function of τ_c . In this calculation, both groups were activated by the input frequency of 3 Hz, which corresponds to ~30 Hz of the firing rate of presynaptic neurons, given the low success rate of synaptic transmission (~10%) [18].

We next investigated how the cooperation within the same group influences the competition between the different groups. For this purpose, we used 2 groups of mutually independent correlated inputs that can compete with each other [14]. After the synaptic efficacies converged to the equilibrium state by using the same input frequency for the 2 groups (3 Hz), we decreased the input rate of either one group by 50%, to examine the influence of the bias in the afferent activity on the competition.

As shown in Fig. 3, when the correlation time is sufficiently short ($\tau_c = 5$ ms), which corresponds to the case where the synaptic cooperation has a facilitatory effect on the strengthening of synapses (Fig. 2C), the 2 groups segregated into the ones winning and losing the competition (which group wins is random) ($t < 400,000$ ms in Figs. 3A and 3B) [14]. After the input frequency of either one group was decreased, the group activated by higher frequency inputs becomes the one winning the competition, which agrees with the prediction from the correlation-based plasticity ($t > 400,000$ ms in Figs. 3A and 3B).

However, in the case of sufficiently longer correlation time ($\tau_c = 10,000$ ms; Fig. 4), the average weights of the 2 groups strongly fluctuated around their temporal mean values but did not segregate from each other when both groups are activated by the inputs with an identical frequency ($t < 400,000$ sec in Figs. 4A and 4B). This result can be partly understood from the cooperative mechanism for longer correlation time as follows: if either one group is more potentiated than the other group, the potentiated group tends to control the timing of postsynaptic spiking. This in turn may activate the inhibitory function of synaptic cooperation that weakens correlated inputs

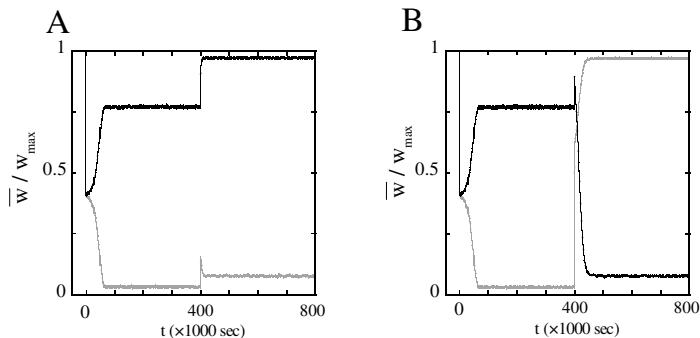


Fig. 3. The time course of the average weights of the 2 groups of correlated inputs when the correlation time is $\tau_c = 5 \text{ ms}$. The input rates of all synapses are 3 Hz for $t < 400,000 \text{ sec}$. Thereafter, the input frequency for either one group, denoted by gray line (A) or black line (B), was decreased by 50 %. In both cases, the group of synapses that are activated by higher frequency inputs (black in (A), gray in (B)) wins the competition.

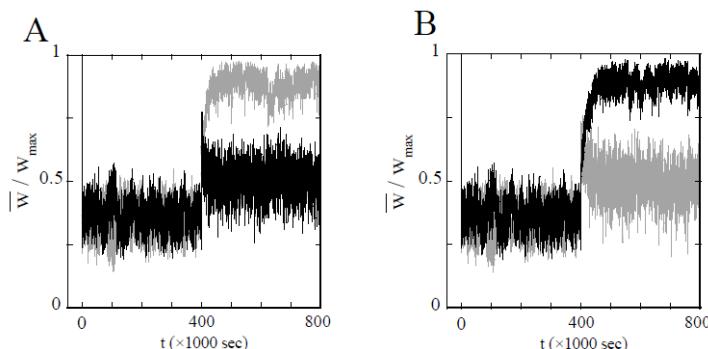


Fig. 4. The time course of the average weights of the 2 groups of correlated inputs when the correlation time is 10,000 ms. The input rates were 3 Hz for all neurons for $t < 400,000 \text{ sec}$, while, for $t > 400,000 \text{ sec}$, the input rate for the group denoted by gray line (A) or black line (B) is decreased by 50 %, similar to Fig. 3. In both cases, the group of synapses that are activated by lower frequency inputs (gray in (A), black in (B)) wins the competition.

(Fig. 2), and thereby inhibit the further potentiation of the same synaptic group. However, as shown in Figs. 4A and 4B ($t > 400,000 \text{ sec}$), if the input frequency of either one group was decreased, the group receiving lower frequency inputs was strengthened more than the one receiving higher frequency inputs. Thus, in the presence of this bias in the input activity, the inhibitory effect of cooperation that weakens the correlated inputs has lead to the emergence of anti-correlation-based competition, in which less frequently activated synapses are potentiated more than frequently activated ones.

4 Conclusions

In this study, we have examined how the cooperation and competition among synapses is regulated by STDP in the presence of the ADFB mechanism. Our results show that the input correlation with a shorter correlation time facilitates the potentiation of a group of synapses, whereas a longer correlation time acts to depress it (Fig. 2). Therefore, the input correlation may have either facilitatory or suppressive effect on the synapses depending on the correlation time. Such cooperation within the same group of correlated inputs may also influence the activity-dependent competition between the different groups. When the cooperation has a facilitatory effect, more frequently activated synapses win the competition, as predicted by the correlation-based rule of plasticity (Fig. 3). However, when the cooperation has a suppressive effect, a group of less frequently activated synapses wins the competition (Fig. 4). These results imply that the plasticity rule governing the synaptic weight dynamics can be switched between correlation-based and anti-correlation-based rules by the input correlation time. The emergence of anti-correlation-based plasticity is reminiscent of the barrel map plasticity in natural environment [6], where the cortical representation of a single whisker is contracted following the removal of all neighboring whiskers. Similar mechanisms might be involved in producing more metabolically efficient cortical representations or smaller, sharper receptive field properties [7].

References

1. Miller, K.D.: Synaptic economics: Competition and cooperation in synaptic plasticity. *Neuron* 17, 371–374 (1996)
2. Shatz, C.J.: Impulse activity and the patterning of connections during CNS development. *Neuron* 5, 745–756 (1990)
3. Kaas, J.H.: Plasticity of sensory and motor maps in adult mammals. *Annu. Rev. Neurosci.* 14, 137–167 (1990)
4. Buonomano, D.V., Merzenich, M.M.: Cortical plasticity: from synapses to maps. *Annu. Rev. Neurosci.* 21, 149–186 (1998)
5. Fox, K.: A critical period for experience-dependent synaptic plasticity in rat barrel cortex. *J. Neurosci.* 12, 1826–1838 (1992)
6. Polley, D.B., Chen-Bee, C.H., Frostig, R.D.: Two direction of plasticity in the sensory-deprived adult cortex. *Neuron* 24, 623–637 (1999)
7. Polley, D.B., Kvasnak, E., Frostig, R.D.: Naturalistic experience transforms sensory maps in the adult cortex of caged animals. *Nature* 429, 67–71 (2004)
8. 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)
9. Feldman, D.E.: Timing-based LTP and LTD at vertical inputs to layer II/III pyramidal cells in rat barrel cortex. *Neuron* 27, 45–56 (2000)
10. Abbott, L.F., Nelson, S.B.: Synaptic plasticity: Taming the beast. *Nat. Neurosci.* 3(suppl.), 1178–1183 (2000)
11. Song, S., Miller, K.D., Abbott, L.F.: Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* 3, 919–926 (2000)

12. Tegnér, J., Képčes, A.: Why neuronal dynamics should control synaptic learning rules. *Adv. Neural Inf. Process. Syst.* 14, 285–292 (2002)
13. Kubota, S., Kitajima, T.: Possible role of cooperative action of NMDA receptor and GABA function in developmental plasticity (submitted)
14. Song, S., Abbott, L.F.: Cortical development and remapping through spike timing-dependent plasticity. *Neuron* 32, 339–350 (2001)
15. Kubota, S., Kitajima, T.: A model for synaptic development regulated by NMDA receptor subunit expression. *J. Comput. Neurosci.* 24, 1–20 (2008)
16. Yang, S.N., Tang, Y.G., Zucker, R.S.: Selective induction of LTP and LTD by postsynaptic $[Ca^{2+}]_i$ elevation. *J. Neurophysiol.* 81, 781–787 (1999)
17. Taniike, N., Lu, Y.F., Tomizawa, K., Matsui, H.: Critical differences in magnitude and duration of N-methyl-D-aspartate (NMDA) receptor activation between long-term potentiation (LTP) and long-term depression (LTD) induction. *Acta Med. Okayama* 62, 21–28 (2008)
18. Koch, C.: Biophysics of computation. Oxford University Press, New York (1999)

An Exemplar-Based Statistical Model for the Dynamics of Neural Synchrony

Justin Dauwels^{1,2}, François Vialatte¹, Theophane Weber²,
and Andrzej Cichocki¹

¹ RIKEN Brain Science Institute, Saitama, Japan

² Massachusetts Institute of Technology, Cambridge, MA

{jdauwels, theo_w}@mit.edu,

{fvialatte, cia}@brain.riken.jp

Abstract. A method is proposed to determine the similarity of a collection of time series. As a first step, one extracts events from the time series, in other words, one converts each time series into a point process (“event sequence”); next one tries to align the events from those different point processes. The better the events can be aligned, the more similar the original time series are considered to be. The proposed method is applied to predict mild cognitive impairment (MCI) from EEG and to investigate the dynamics of oscillatory-event synchrony of steady-state visually evoked potentials (SSVEP).

1 Introduction

The problem of detecting correlations between neural signals (“neural synchrony”) has recently attracted much attention in the neuroscience community (e.g., [1]). For instance, it has frequently been reported that abnormalities in neural synchrony lie at the heart of brain disorders such as Alzheimer’s disease (e.g., [2]).

In this paper, we present a method to quantify dynamical interdependencies between a collection of time series (e.g., spike data, EEG or MRI signals; see Fig. [1]). As a first step, we extract “events” from each time series, in other words, we transform each time series into a point process, i.e., a sequence of events. Next we try to align the events of each of those point processes. The better the point processes can be aligned, the more similar the point processes and hence the original time series are considered to be. In our method, the similarity may vary over time.

Our approach is inspired by the “stochastic event synchrony” measures (SES) of [3], which are also based on event alignment. However, those measures are only applicable to *pairs of signals*, in addition, they are not time-dependent. As a result, they cannot be applied to study the *dynamics* of neural synchrony. The proposed technique can deal with *collections* of signals, moreover, it treats the similarity parameters as stochastic processes; therefore, it may be used to investigate how the synchrony of a collection of neural signals *evolves* over time.

The proposed method makes use of so-called “exemplars”, which are events that serve as representatives of each cluster; it is related to exemplar-based approaches for clustering such as affinity propagation [4] and the convex clustering

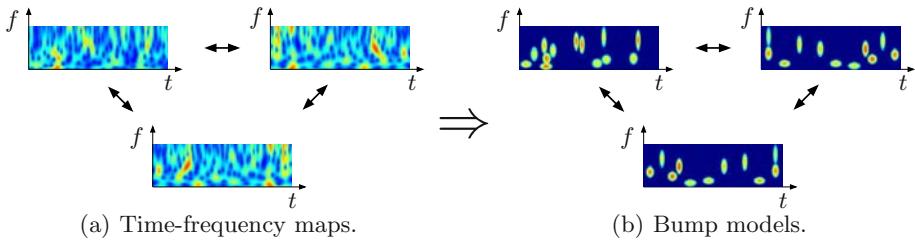


Fig. 1. Similarity of 3 EEG signals ($N = 3$); from their time-frequency transforms (left), one extracts bump models (right), which are then aligned by the proposed algorithm

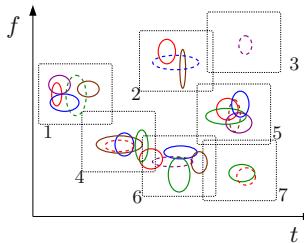


Fig. 2. Five bump models on top of each other ($N = 5$; each model has a different color); the dashed boxes indicate clusters, the dashed ellipses correspond to exemplars

algorithm of [5]. The exemplar-based formulation allows us to extend the pairwise similarity measures of [3] to multivariate similarity measures.

We will use our method to predict mild cognitive impairment (MCI) from EEG. This application was also investigated in [3], however, the method of [3] only considers pairwise synchrony; we will quantify the synchrony of multiple brain areas *simultaneously*, which improves the sensitivity of EEG to diagnose MCI and leads to a more detailed understanding of the abnormalities in EEG synchrony in MCI patients. As a second application, we study the dynamics of oscillatory-event synchrony of steady-state visually evoked potentials (SSVEP).

This paper is organized as follows. In the following section, we outline the exemplar-based statistical model for synchrony; in Section 3 we describe how to perform inference in that model, and we characterize the underlying combinatorial problem. As an illustration, we apply our method to detect MCI induced perturbations in EEG synchrony (Section 4), and to analyze the neural synchrony of steady-state visually evoked potentials (Section 5).

2 Exemplar-Based Statistical Model

We consider N signals S_1, \dots, S_N from which we extract point processes X_1, \dots, X_N by some method. Each point process X_i is a list of n_i points (“events”) in a given multi-dimensional set $\mathcal{S} \subseteq \mathbb{R}^M$, i.e., $X_i = \{X_{i,1}, X_{i,2}, \dots, X_{i,n_i}\}$ with

$X_{i,k} \in \mathcal{S}$ for $k = 1, \dots, n_i$ and $i = 1 \dots N$. Let us consider the example of bump models [6] extracted from the time-frequency maps of EEG signals (see Fig. 1). The time-frequency (“wavelet”) transform of each EEG signal is approximated as a sum of half-ellipsoid basis functions, referred to as “bumps” [6]; each bump is described by five parameters: time T , frequency F , width ΔT , height ΔF , and amplitude W . (In this paper, we use precisely the same procedure and same parameter settings as in [6] to extract bumps.) How similar are the N resulting bump models $X_i = ((T_{i,1}, F_{i,1}, \Delta T_{i,1}, \Delta F_{i,1}, W_{i,1}), \dots, (T_{i,n_i}, F_{i,n_i}, \Delta T_{i,n_i}, \Delta F_{i,n_i}, W_{i,n_i}))$ (with $i = 1, 2, \dots, N$)? Intuitively speaking, N bump models X_i may be considered well-synchronized if bumps appear in all models (or almost all) simultaneously, potentially with some slowly varying offset in time and frequency. In other words, if one overlays N more or less resembling bump models (cf. Fig. 2 with $N = 5$), and removes the potential offsets in time and frequency, bumps naturally appear in clusters that contain precisely one bump from all (or almost all) bump models. In the example of Fig. 2, cluster 1, 5 and 6 contain bumps from all 5 models X_i , cluster 2, 4 and 7 contains bumps from 3, 4, and 2 models respectively, and cluster 3 consists of a single bump.

This intuitive concept of similarity may readily be translated into a generative stochastic model. In that model, the N point processes X_i are treated as independent noisy observations of a hidden “mother” process \tilde{X} . An observed sequence X_i is obtained from \tilde{X} by the following three-step procedure:

1. COPY: generate a copy of the mother bump model \tilde{X} ,
2. DELETION: delete some of the copied mother bumps,
3. PERTURBATION: alter the position and shape of the remaining mother bump copies, amounting to the bump model X_i .

As a result, each sequence X_i consists of “noisy” copies of a non-empty subset of mother bumps. The point processes X_i may be considered well-synchronized if there are only few deletions (cf. Step 2) and if the bumps of X_i are well aligned with the corresponding mother bumps (cf. Step 3), apart from some slowly varying offset in time and frequency. One way to determine the synchrony of given point processes X_i is to first reconstruct the hidden mother process \tilde{X} , and next to align the point processes X_i with the mother process \tilde{X} . Inferring the mother process is a high-dimensional estimation problem, the underlying probability distribution typically has a large number of local extrema. Therefore, we will use an alternative procedure: as in [4], we will assume that each cluster contains one *identical* copy of a mother bump, the other bumps in that cluster are *noisy* copies of that mother bump. The identical copy, referred to as “exemplar”, plays the role of “center” or “representative” of each cluster (see Fig. 2).

The exemplar-based formulation amounts to the following inference problem: given the point processes X_i , we need to identify the bumps that are exemplars and the ones that are noisy copies of some exemplar. Obviously, this inference problem also has potentially many locally optimal solutions, however, in contrast to the original (continuous) inference problem, we can in practice find the global optimum by integer programming (see Section 3).

We now discuss the above exemplar-based statistical model in more detail. The number M of mother events \tilde{X}_m is geometrically distributed with parameter $\lambda \text{vol}(S)$. Each mother event \tilde{X}_m for $m = 1, \dots, M$ is uniformly distributed in S .

The noisy copies are modeled as follows. The number C_m of copies is modeled by a prior $p(c_m|\theta^c)$, parameterized by θ^c , which in turn has a prior $p(\theta^c)$. We consider as prior for C_m a binomial distribution $\text{Bi}(p_s)$ with $N - 1$ trials and probability of success p_s . We adopt a conjugate prior for the parameters p_s , i.e., the beta distribution $B(\kappa, \lambda)$. Conditional on the number C_m of copies, the copies are attributed uniformly at random to other signals X_j , with the constraints of at most one copy per signal and $j \neq i(m)$; since there are $\binom{N-1}{c_m}$ possible attributions $\mathcal{A}_m \subseteq \{1, \dots, i(m) - 1, i(m) + 1, \dots, N\}$ with $|\mathcal{A}_m| = c_m$, the probability mass of an attribution \mathcal{A}_m is $p(\mathcal{A}_m|c_m) = \binom{N-1}{c_m}^{-1}$.

The process of generating a noisy copy $X_{i,r}$ from a mother bump \tilde{X}_m is described by a conditional distribution $p_x(x_{i,r}|\tilde{x}_m; \theta_i^x)$. The vectors θ_i^x may be treated as random vectors with non-trivial priors $p(\theta_i^x)$. In the case of bump models (cf. Fig. 2), we generate copies by shifting the mother bump center while the other parameters (width, height, and amplitude) are drawn from some prior distribution, independently for each copy. The center offset may be modeled as a Gaussian random variable with mean vector $(\delta_{t,i}, \delta_{f,i})$ and diagonal non-isotropic covariance matrix $V_i = \text{diag}(s_{t,i}, s_{f,i})$, and hence, $\theta_i^x = (\delta_{t,i}, \delta_{f,i}, s_{t,i}, s_{f,i})$. The model allows an *average offset* $(\delta_{t,i}, \delta_{f,i})$ between bump models; even if the average offset is zero, there may still be random offsets between the exemplars and their copies (see Fig. 2). We will assume that $s_{t,i} = s_t$ and $s_{f,i} = s_f$ for all i . We adopt improper priors $p(\delta_{t,i}) = 1 = p(\delta_{f,i})$ for $\delta_{t,i}$ and $\delta_{f,i}$ respectively, and conjugate priors for s_t and s_f , i.e. scaled inverse chi-square distributions.

The parameters $\theta = (\theta^c, \theta^x)$ might be constant or time-varying. In the latter case, we make the reasonable assumption that the parameters θ vary smoothly over time. The prior on $\theta(t)$ can then be chosen as:

$$p(\theta(t)) = \frac{1}{Z(\beta)} \prod_j \exp \left[\beta_j \int_0^T \left(\frac{d^2 \theta_j}{dt^2} \right)^2 dt \right], \quad (1)$$

where $Z(\beta)$ is a normalization constant and β_j (for $j=1,2,\dots$) is a real positive number. For later convenience, we will introduce some more notation. The exemplar associated to mother event \tilde{X}_m is denoted by $X_{i(m), k(m)}$, it is the event $k(m)$ in point process $X_{i(m)}$. We denote the set of pairs $(i(m), k(m))$ by \mathcal{I}^{ex} . A noisy copy of \tilde{X}_m is denoted by $X_{j(m), \ell(m)}$, it is the event $\ell(m)$ in point process $X_{j(m)}$ with $j(m) \in \mathcal{A}_m$. We denote the set of all pairs $(j(m), \ell(m))$ associated to \tilde{X}_m by $\mathcal{I}_m^{\text{copy}}$, and furthermore define $\mathcal{I}^{\text{copy}} \triangleq \mathcal{I}_1^{\text{copy}} \cup \dots \cup \mathcal{I}_M^{\text{copy}}$ and $\mathcal{I} = \mathcal{I}^{\text{ex}} \cup \mathcal{I}^{\text{copy}}$; the latter contains the indices of all exemplars and their copies. In this notation, the exemplar-based probabilistic model may be written as:

$$p(\tilde{X}, X, \mathcal{I}, \theta) = p(\theta^c)p(\theta^x)(1 - \lambda \text{vol}(S))\lambda^M N^{-M} \prod_{m=1}^M \delta(x_{i(m), k(m)} - \tilde{x}_m) \\ \cdot p(c_m | \theta^c) \binom{N-1}{c_m}^{-1} \prod_{(i,j) \in \mathcal{I}_m^{\text{copy}}} p_x(x_{i,j} | \tilde{x}_m, \theta^x). \quad (2)$$

If the point processes $X = (X_1, \dots, X_N)$ are well-synchronized, almost all processes X_i contain a copy of each mother bump \tilde{X}_m ; the sets $\mathcal{I}_m^{\text{copy}}$ are either of size $N - 1$ or are slightly smaller. In the case of bump models, the variances s_t and s_f are then small. Note that \mathcal{I} specifies the exemplars and their copies, and as a result, from \mathcal{I} one can deduce various properties of the clusters, e.g., average number of bumps per cluster; the parameters s_t and s_f are part of the parameter vector θ . Therefore, given point processes $X = (X_1, \dots, X_N)$, we wish to infer \mathcal{I} and θ , since those variables contain information about similarity.

3 Statistical Inference

A reasonable approach to infer (\mathcal{I}, θ) is maximum a posteriori (MAP) estimation:

$$(\hat{\mathcal{I}}, \hat{\theta}) = \underset{(\mathcal{I}, \theta)}{\operatorname{argmax}} \log p(\tilde{X}, X, \mathcal{I}, \theta). \quad (3)$$

There is no closed form expression for (3), therefore, we need to resort to numerical methods. A simple technique to try to find (3) is coordinate descent: We first choose initial values $\hat{\theta}^{(0)}$, and then perform the following updates for $r \geq 1$ until convergence:

$$\hat{\mathcal{I}}^{(r)} = \underset{\mathcal{I}}{\operatorname{argmax}} \log p(\tilde{X}, X, \mathcal{I}, \hat{\theta}^{(r-1)}) \quad (4)$$

$$\hat{\theta}^{(r)} = \underset{\theta}{\operatorname{argmax}} \log p(\tilde{X}, X, \hat{\mathcal{I}}^{(r)}, \theta). \quad (5)$$

3.1 Integer Program

We can write the update (4) as an integer program, i.e., a discrete optimization problem with linear objective function and linear (equality and inequality) constraints; we will omit the details here due to space constraints, and will merely discuss some general observations. The update (4) is for $N = 2$ equivalent to *bipartite maximum weighted matching* optimization, a problem that can be solved in polynomial time, for instance by using the LP relaxation of the corresponding IP formulation, or by the max-product message-passing algorithm detailed in [3]. We have shown that for $N > 2$, the combinatorial problem (4) is equivalent to *weighted k-dimensional matching* optimization, an NP-hard problem in the general case. Therefore, the extension from 2 time series to more than 2 is far from trivial. In practice (see the applications of Section 4 and 5), we were often able to solve the corresponding integer program very efficiently (using CPLEX). For integer programs with more than 10'000 variables and 5'000 constraints, the solution of a given was obtained in less than 1 second on a fast processor (3GHz). The total running time of the algorithm (iterations of equations (14) and (15) until convergence) was under 7 seconds on average.

3.2 Parameter Estimation

We now consider the update (5), i.e., estimation of the parameters $\theta = (\theta^x, \theta^c)$. First we treat constant parameters θ . The estimate $\hat{\theta}^{(r+1)} = (\hat{\theta}^{x(r+1)}, \hat{\theta}^{c(r+1)})$ (5) is often available in closed-form. This is in particular the case for the parametrization $\theta_i^x = (\delta_{t,i}, \delta_{f,i}, s_t, s_f)$. The point estimates $\hat{\delta}_{t,i}^{(r+1)}$ and $\hat{\delta}_{f,i}^{(r+1)}$ are the (sample) mean of the timing and frequency offset respectively, computed between all noisy copies in X_i and their associated exemplars. The estimates $\hat{s}_t^{(r)}$ and $\hat{s}_f^{(r)}$ are obtained similarly. The expression for the parameter p_s of the binomial prior for the number of copies C_m is also straightforward. Now we treat time-varying parameters θ (cf. (1)). The estimate $\hat{\theta}^{(r+1)}(t) = (\hat{\theta}^{x(r+1)}(t), \hat{\theta}^{c(r+1)}(t))$ (5) are usually not available in closed-form. Let us again consider the parametrization $\theta_i^x = (\delta_{t,i}, \delta_{f,i}, s_t, s_f)$. The point estimates $\hat{\delta}_{t,i}^{(r+1)}$ are given by:

$$\hat{\delta}_{t,i}^{(r)} = \underset{\delta_{t,i}}{\operatorname{argmin}} \sum_{k,i',k'} \hat{b}_{i,k,i',k'}^{(r)} \frac{(T_{i,k} - T_{i',k'} - \delta_{t,i}(T_{i',k'}))^2}{\hat{s}_t^{(r)}(T_{i',k'})} + \beta_{\delta_t} \int_0^T \left(\frac{d^2 \delta_{t,i}}{dt^2} \right)^2 dt, \quad (6)$$

where $\hat{b}_{i,k,i',k'}^{(r)}$ equals one if, according to $\hat{\mathcal{I}}^{(r)}$ (4), bump $X_{i,k}$ is considered as “noisy” copy of the exemplar $X_{i',k'}$ and zero otherwise. Similarly we have:

$$\begin{aligned} \hat{s}_{t,i}^{(r)} = \underset{s_{t,i}}{\operatorname{argmin}} \sum_{k,i',k'} & \hat{b}_{i,k,i',k'}^{(r)} \left[\log s_t(T_{i',k'}) + \frac{(T_{i,k} - T_{i',k'} - \hat{\delta}_{t,i}^{(r)}(T_{i',k'}))^2}{s_t(T_{i',k'})} \right] \\ & + \beta_{s_t} \int_0^T \left(\frac{d^2 s_{t,i}}{dt^2} \right)^2 dt. \end{aligned} \quad (7)$$

Note that the updates (6) and (7) are coupled. The processes $\hat{\delta}_{t,i}^{(r)}$ and $\hat{s}_{t,i}^{(r)}$ may be determined jointly by coordinate descent: with an initial guess for $\hat{s}_{t,i}^{(r)}$ one determines $\hat{\delta}_{t,i}^{(r)}$ (6), with the resulting estimate $\hat{\delta}_{t,i}^{(r)}$ one updates $\hat{s}_{t,i}^{(r)}$ through (7), and so on, until convergence. A similar procedure can be applied to jointly determine $\hat{\delta}_{f,i}^{(r)}$ and $\hat{s}_{f,i}^{(r)}$. The updates (6)–(7) may be solved by a variational approach, in particular, by solving the associated Euler-Lagrange equations numerically. However, it is well known that the unique solution of optimization problems of the form (6) over the space of twice differentiable functions are smoothing cubic splines [8]. As a result, the expression (6) can be found by cubic-spline smoothing. This does not apply, however, to the updates (7), since it involves non-quadratic terms. One option is to solve this expression by numerically integrating the associated Euler-Lagrange equations, as we pointed out earlier. Alternatively, one may determine the second-order Taylor expansion of the first term in (7). This results in quadratic approximations, known as saddle-point or Laplace approximation, of the non-quadratic terms. The solutions of those relaxed variational problems are smoothing cubic spline. In other words, the expressions $\hat{s}_{t,i}^{(r)}$ and $\hat{s}_{f,i}^{(r)}$ can be found *approximately* by cubic-spline smoothing. The latter is much

faster than numerically solving the associated Euler-Lagrange equations, which is a major advantage. One can derive similar variational problems for the parameter p_s of the binomial prior, which in turn can also be solved practically by cubic-spline smoothing.

4 Diagnosis of MCI from EEG

As a first application, we consider the problem of diagnosing mild cognitive impairment (MCI) from EEG. For the sake of comparison, we used the same EEG data as in [3], i.e., rest eyes-closed EEG data recorded from 21 sites on the scalp based on the 10–20 system [9] with sampling frequency of 200Hz and band-pass filter between 4 and 30Hz. The subjects consist of two groups: (i) 22 patients suffering from mild cognitive impairment (MCI), who subsequently developed mild AD; (ii) a control set of 38 age-matched, healthy subjects who had no memory or other cognitive impairments. Pre-selection was conducted to ensure that the data were of a high quality, as determined by the presence of at least 20s of artifact free data. We aggregated the 21 bump models in five regions (frontal, temporal left and right, central, occipital) by means of the aggregation algorithm described in [6], resulting in a bump model for each of those five regions ($N = 5$).

A large variety of *classical* synchrony measures (more than 30 in total) have in [3] been applied to both data sets with the aim of detecting MCI induced perturbations in EEG synchrony; none of those classical measures except full frequency Directed Transfer Function (ffDTF) [7], which is a Granger causality measure, was able to detect significant loss of EEG synchrony in MCI patients; note that since we considered many methods simultaneously, we need to apply Bonferroni postcorrection: the p-values need to be multiplied by the number of measures. On the other hand, the stochastic-event-synchrony measure ρ_{spur} , proposed in [3], resulted in significant differences between both subject groups ($p = 2.1 \cdot 10^{-4}$). This seems to indicate that there is an increase of unsynchronized activity in MCI patients.

The results from our exemplar-based approach are summarized in Table II: we adopted constant parameters, because time-varying parameters for spontaneous EEG. We studied the following statistics: posterior distribution $p(c_m = i|X) = p_i^c$ of the number of copies of each exemplar c_m , parameterized by $(p_0^c, p_1^c, \dots, p_4^c)$; \bar{c}_m : average number of copies per cluster; s_t : variance in time domain (“time jitter”); s_f : variance in frequency domain (“frequency jitter”); $\Delta\bar{T}$: average width of bumps; $\Delta\bar{F}$: average height of bumps; \bar{F} : average frequency of bumps. We also consider the linear combination h^c of all parameters p_i^c that optimally separates both subject groups. Interestingly, the latter statistic amounts to about the same p -value as the index ρ_{spur} of SES [3]. The posterior $p(c_m|X)$ mostly differs in p_1^c , p_2^c and p_4^c : in MCI patients, the number of clusters of size five (p_4^c) significantly decreases; on the other hand, the number of clusters of size one (p_1^c) and two (p_2^c) significantly increases. This explains and confirms the observed increase of ρ_{spur} in MCI patients [3]. Combining h^c with ffDTF and $\Delta\bar{T}$ allows

to separate the two groups quite well (more than 90% correctly classified); this is far better than what can be achieved by means of classical similarity measures (about 75% correctly classified). Classification rates between 80 and 85% can be obtained by combining two features. (The classification rates were obtained through crossvalidation, i.e., the leave-one-out procedure.)

We also verified that the measures p_i^c , \bar{c}_m , h^c , and s_t are not correlated with other synchrony measures, e.g., Pearson correlation coefficient, magnitude and phase coherence, phase synchrony etc. (Pearson r , $p > 0.10$). In contrast to the classical measures, they quantify the synchrony of oscillatory events instead of more conventional amplitude or phase synchrony, therefore, they provide *complementary* information about EEG synchrony.

Table 1. Sensitivity of multivariate SES for diagnosing MCI (p-values for Mann-Whitney test; * and ** indicate $p < 0.05$ and $p < 0.005$ respectively)

Stat.	p_0^c	p_1^c	p_2^c	p_3^c	p_4^c	\bar{c}_m	h^c	s_t	s_f	$\Delta\bar{T}$	$\Delta\bar{F}$	\bar{F}
p-value	0.016	2.9.10 ^{-4**}	0.089	0.59	0.0054*	1.10 ^{-3**}	1.10 ^{-4**}	0.46	0.28	2.3.10 ^{-4**}	0.023*	2.10 ^{-3**}

5 Dynamics of Oscillatory Event Synchrony in SSVEP

As an application of the exemplar-based statistical model with *time-varying* parameters, we investigate how EEG synchrony evolves in response to a visual stimulus. In particular, human scalp EEG was recorded in a dark room while a subject was exposed to flickering light. The stimulus was a single flashing white square that flickers at 16Hz during 4 sec, and produces a steady-state response of the human visual system at the same frequency, referred to as “steady-state visually evoked potentials” (SSVEP). In total 50 trials were recorded with a baseline period of 3 sec, stimulation during 4 sec, and a post-stimulation period of 3 sec during which EEG resumed to the baseline activity. EEG data was recorded from 64 sites on the scalp, based on the extended 10-20 standard system, with a sampling frequency of 2048Hz; the EEG was high-pass filtered off-line

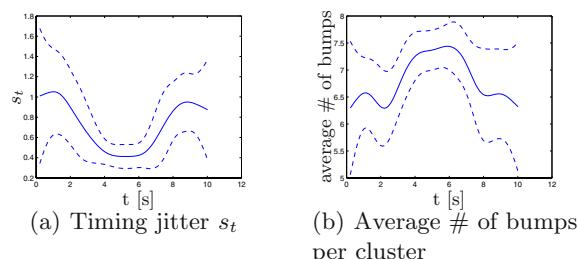


Fig. 3. Time-varying parameters (solid line: average; dashed lines: \pm one standard deviation over the 50 trials)

with cut-off frequency of 3Hz (Butterworth forward and reverse), next it was downsampled by a factor of 8. A Biosemi system with average reference was used. As in Section 4, we aggregated the electrode sites into separate zones (here 9 instead of 5, since we used 64 instead of 21 channels), and extracted a bump model from each zone by means of the algorithms of [6]. The time-frequency maps were restricted to the frequency range of SSVEP, i.e., 15–17Hz.

Fig. 3 shows that the timing jitter significantly decreases during stimulation ($t \in [3s, 7s]$); also the average number of bumps per cluster clearly increases. The parameters β_{δ_t} , β_{δ_f} , β_{s_t} , and β_{s_f} (cf. [6][7]) were determined by cross-validation.

References

1. Pereda, E., Quiroga, R.Q., Bhattacharya, J.: Nonlinear multivariate analysis of neurophysiological signals. *Progress in Neurobiology* 77, 1–37 (2005)
2. Uhlhaas, P., Singer, W.: Neural synchrony in brain disorders: relevance for cognitive dysfunctions and pathophysiology. *Neuron* 52, 155–168 (2006)
3. Dauwels, J., Vialatte, F., Rutkowski, T., Cichocki, A.: Measuring neural synchrony by message passing. In: *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge
4. Frey, B., Dueck, D.: Clustering by passing messages between data points. *Science* 315(5814), 972–976 (2007)
5. Lashkari, D., Golland, P.: Convex clustering with exemplar-based models. In: *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge
6. Vialatte, F., Martin, C., Dubois, R., Haddad, J., Quenot, B., Gervais, R., Dreyfus, G.: A Machine learning approach to the analysis of time-frequency maps, and its application to neural dynamics. *Neural Networks* 20, 194–209 (2007)
7. Kamiński, M., Liang, H.: Causal Influence: Advances in Neurosignal Analysis. *Critical Review in Biomedical Engineering* 33(4), 347–430 (2005)
8. Schoenberg, I.J.: Spline functions and the problem of graduation. *Mathematics* 52, 974–50
9. Nunez, P., Srinivasan, R.: *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press, Oxford (2006)

Towards a Comparative Theory of the Primates' Tool-Use Behavior

Toshisada Mariyama and Hideaki Itoh

Department of Computational Intelligence and System Science,
Tokyo Institute of Technology,
4259, Nagatsuta-cho, Midori-ku, Yokohama 226-8502, Japan
`{mariyama, hideaki}@brn.dis.titech.ac.jp`

Abstract. Primates can use tools effectively, and each of the species exhibits different tool-use behavior. We study what difference in tool-use behavior will be observed if different internal models of the environment are used for planning. We compare two agents with different internal models, each of which consists of an artificial neural network. The first agent (agent 1) has an internal model that predicts the outcome of a given task. The second agent (agent 2) has an internal model that predicts a step-by-step state transition of the environment. By employing a rake-use task, we demonstrate that agent 2 can adapt to a task change more quickly than agent 1. The results suggest a possibility that the internal model available to each primate species can be inferred by investigating the response of the species to adapt to task changes.

1 Introduction

Primates can use objects other than their own bodies as tools. Because non-primate animals rarely use tools, tool-use behavior is considered a landmark of primates' high intelligence and has a long history of research. Köhler [1] studied captive chimpanzees' tool-use behavior, suggesting that the tool-use behavior can be a measure of the primates' intelligence level. Goodall [2] and many other studies [3–6] observed that wild chimpanzees use tools.

It should be noted that each primate species exhibits different tool-use behaviors [4, 5, 7]. Matsuzawa [5] proposed dividing primate species into groups on the basis of the number of tools that they can employ to accomplish a goal. Torigoe et al. [7] categorized 74 species of non-human primates by their object manipulation patterns.

In contrast to the large amount of literature available on observational studies, there are few theoretical studies. A theoretical analysis of the underlying mechanisms of the behavior differences can lead better understanding of the human intelligence as well as contribute toward designing of highly intelligent robots.

As a step towards this goal, we study what difference in tool-use behavior will be observed if different internal models of the environment are used for planning of the behavior.

Internal models are considered to play an important role in primates' brains when performing difficult tasks, including the use of tools [8–11]. For example, Imamizu et al. [10] showed by means of an fMRI study that when human subjects use a new

tool, a new activity pattern is observed in the cerebellum, which is the location in the brain where an internal model is suggested to be acquired. Wolpert et al. [11] proposed the MOSAIC model to solve complex tasks by combining several internal models.

The internal model available to each species may differ from each other. For example, Köhler pointed out that there exists a major difference between humans and chimpanzees in their understanding of the geometric dynamics [1]. Köhler let chimpanzees solve a task in which they had to stack two boxes together in order to obtain a banana hanging from the ceiling. A chimpanzee stacked the boxes into an unsteady position that humans can, at a glance, predict to be unstable and ready to collapse. For several years, the chimpanzee repeated the action, causing the boxes to collapse every time. Thus, it is considered that the chimpanzees' internal model is different from that of humans; if not, the chimpanzee could have predicted the collapse of the boxes and changed its behavior.

For the abovementioned reasons, we study the effect of a difference in the internal model on tool-use behavior. In Sections 2 and 3, we conduct a simulation study to compare two agents, each of which has a different internal model. The first agent has an internal model that predicts the outcome (success or failure) of a given task. The other agent has an internal model that predicts a step-by-step state transition of the environment. By employing a rake-use task, which is a typical example of tool-use behavior [12, 13], we show that the latter agent can adapt to a task change more quickly than the former agent. The results obtained suggest a possibility that the internal model available in each primate species can be inferred by investigating the response of the species to adapt to task changes. In Section 4, we discuss measures to be adopted to take further steps towards formulating a comparative theory of the primates' tool-use behavior.

2 Methods

2.1 Task

The tasks involved in our experiments are rake-use tasks. A *trial* of a task begins when an agent is presented with a right and a left tray, on each of which an L-shaped rake and a food pellet are placed. The agent can pull either one of the two rakes in the vertical direction. The food pellet is accessible by pulling the rake on one of the trays, and not on the other tray.

After the agent has pulled a rake, the agent observes the result of pulling the unselected rake. This is to facilitate the learning process of the agent. The task ends when the agent has pulled a rake and observed the result of pulling the unselected rake.

An agent solves two types of tasks. In task 1, trials with six different configurations of the rakes and the food pellets are assigned to the agent (Fig. 1 (A)). We call a set of such six trials a *session*. In each session, the six trials with the different configurations are assigned to the agent in a random order. The sessions are repeated until each agent learns the rake that should be selected in order to obtain the food pellet for every configuration.

Task 2 is assigned to the agent after it has learned to solve task 1. In task 2, trials with two different configurations of the rakes and the pellets are assigned to the agent (Fig. 1 (B)). We call a set of such two trials a *session* of task 2. It must be noted that the rake head on each tray is flipped horizontally in task 2 when compared with task 1. Thus, we can study by means of task 2 the manner in which the agents adapt to the modification of the task.

These tasks are based on Hauser's study [13]. In his study, tasks 1 and 2 were assigned to *tamarins* in an intermingled order, i.e., task 2 was assigned before they learned task 1 completely. We modified the task used in his study because the difference in the agents' internal models would not lead to a clear behavioral difference.

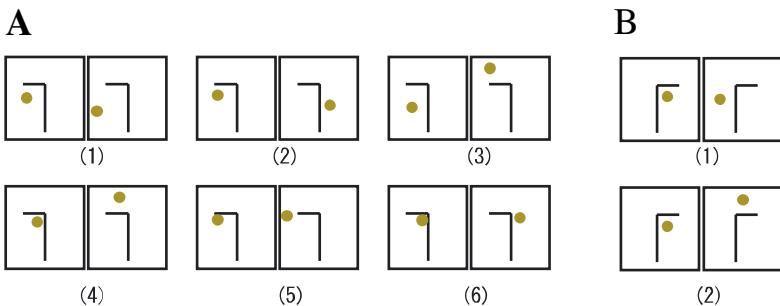


Fig. 1. Task conditions. (A) task 1 and (B) task 2.

2.2 Model

The environment is defined as a finite state machine $E = (E_K, E_\Sigma, E_\delta, E_q, E_f)$.

State space E_K : We define the set of the states of the environment as $E_K = \{(F_X^R, F_Y^R, T_{XR}^R, T_{XL}^R, T_Y^R, F_X^L, F_Y^L, T_{XR}^L, T_{XL}^L, T_Y^L) | -2 < F_X^R, F_X^L, T_{XR}^R, T_{XL}^R, T_{XR}^L, T_{XL}^L < 2, 0 \leq F_Y^R, F_Y^L, T_Y^R, T_Y^L < 4\}$, where each variable indicates the positions of the rake and the food pellet (Fig.2). The superscripts R and L denote the right and the left tray, respectively. F_X^* and F_Y^* , in which

$* \in \{R, L\}$, indicate the position of the food pellet in the Cartesian coordinate system in which the origin $(0, 0)$ is the position of the handgrip of the rake. T_{XR}^* and T_{XL}^* represent the x-coordinates of the right and left ends of the horizontal head part of the rake. T_Y^* represents the y-coordinate of the rake head. For later use, we define $S^* := (F_X^*, F_Y^*, T_{XR}^*, T_{XL}^*, T_Y^*)$ in which $* \in \{R, L\}$.

Alphabet E_Σ : The alphabet is the set of possible inputs from the agent to the environment. We define $E_\Sigma = \{A^R, A^L\}$, where A^R and A^L imply that the agent pulls the right and the left tray, respectively.

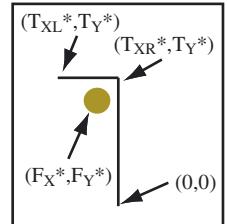


Fig. 2. The coordinates of the rake and the food pellet

State transition function E_δ : E_δ defines the manner in which the environment changes its state after receiving an input from the agent.

We assume discrete time $t = 0, 1, 2, \dots$. Each trial begins with $t = 0$. By adding “(t)” to each state variable, we indicate its value at time t ; for example, $S^R(t)$ denotes the positions of the rake and the food pellet on the right tray at time t .

After the agent selects A^R , the positions of the rake and the food pellet on the right tray vary as follows:

$$T_Y^R(t+1) = \begin{cases} T_Y^R(t)-1 & \text{if } 1 \leq T_Y^R(t), \\ 0 & \text{if } 0 \leq T_Y^R(t) < 1, \end{cases} \quad (1)$$

$$F_Y^R(t+1) = \begin{cases} T_Y^R(t+1) & \text{if } T_{XL}^R(t) \leq F_X^R(t) \leq T_{XR}^R(t) \\ & \quad \text{and } T_Y^R(t)-1 < F_Y^R(t) \leq T_Y^R(t), \\ F_Y^R(t) & \text{otherwise.} \end{cases} \quad (2)$$

The other parameters do not change. If the agent selects A^L , the positions of the rake and the food pellet on the left tray vary in the same manner.

Initial state E_q : The initial state for each trial is given according to the rake and food positions in Fig. 1. We set $T_Y^* = 2$. We set $T_{XL}^* = -1$ (or 0) and $T_{XR}^* = 0$ (or 1) for task 1 (or 2, respectively).

Final states E_f : The final states are the states in which $T_Y^R = 0$ or $T_Y^L = 0$.

2.3 Agent

At time $t = 0$ (i.e., the beginning of each trial), the agent observes the environment’s state and predicts the result of pulling each of the two rakes. Subsequently, it chooses to pull the rake that is more likely to result in the agent gaining the food pellet.

We study two agents with different internal models to predict the future. The first agent, which we call *agent 1*, has an internal model, which we call *model 1*, that predicts the outcome (success or failure to obtain the food) of pulling each rake. The second agent, *agent 2*, has a different internal model, *model 2*, that predicts the step-by-step state transition of the environment.

We employ artificial neural networks [14] for the internal models. The neural networks are exactly the same for both models 1 and 2, except for the teacher signal. We provide the details in the following sections.

2.3.1 Model 1

We employ a three-layer neural network that consists of 5 neurons in the input layer, 5 neurons in the middle layer, and 1 neuron in the output layer. We employ a sigmoid function as the response function of each neuron. The output of each neuron is expressed by $f(x) = 1/\{1 + \exp(-x)\}$, where x denotes the input to the neuron.

We denote the input to the i -th neuron in the input layer by x_i . Let w_{ij} be the synapse weight from the i -th neuron in the input layer to the j -th neuron in the middle layer, and let v_j be the synapse weight from the j -th neuron in the middle layer to the neuron in the output layer. The output neuron's output is $z = f(\sum v_j f(\sum w_{ij} f(x_i)))$.

In model 1, the neural network is trained so that it can predict the outcome of pulling the rake on each tray during each task. Recall that $S^*(0)$ denotes the initial positions of the rake and the food on each tray ($*$ is R or L for the right or the left tray, respectively). Let $O(S^*(0))$ denote the output of the network for input $S^*(0)$. Let O^* represent the outcome of pulling the rake; $O^* = 1$, for success and $O^* = 0$, for failure. After each trial, we train the network by giving it $S^*(0)$ as the input and O^* as the teacher signal, for each of the right and the left trays. We employ the back propagation method [14] to train the network.

At $t = 0$, the neural network predicts the outcome of pulling the rake on each tray by calculating $O(S^*(0))$. Because the network is trained such that its output is 1 when it predicts success to obtain the food, a higher value of $O(S^*(0))$ indicates that success is more likely. Therefore, the agent should choose to pull the rake on the tray with larger $O(S^*(0))$ more often as compared to pulling the tray with smaller $O(S^*(0))$. Although there exists several ways to do this (e.g., softmax and epsilon-greedy), for simplicity, we make the agent select the right tray with the probability of

$$P(R) = O(S^R(0)) / (O(S^R(0)) + O(S^L(0))). \quad (3)$$

2.3.2 Model 2

Model 2 employs the same neural network as that in model 1. The network is trained such that it predicts the step-by-step state transition of the environment. At each instant t , it predicts the configuration of each tray at instant $t + 1$, assuming that the agent pulls the rake on the tray. For simplicity, we make the network predict only whether the food pellet moves together with the rake. From this prediction, the whole configuration can be predicted (which is hand-coded). The input to the network is denoted as $S^*(t)$, and the teacher signal $O^*(t)$ is 1 if $T_{XL}^*(t) \leq F_x^*(t) \leq T_{XR}^*(t)$ and $T_Y^*(t)-1 < F_Y^*(t) \leq T_Y^*(t)$, and 0 otherwise.

In each trial, the network makes predictions and learns the correct answers twice, at $t = 0$ and $t = 1$ for the following reason. Recall that the initial y-coordinate of the rake's head, $T_Y^*(0)$, is 2, and each time the agent pulls the rake, 1 is subtracted from T_Y^* (Eq. (1)). Thus, the rake head reaches the agent's position at time $t = 2$. This means that the network should make predictions twice, at $t = 0$ and $t = 1$. For each tray during a trial, the network is trained twice, at $t = 0$ and $t = 1$, with $S^*(t)$ and $O^*(t)$ as the input and the teacher signals, respectively.

At instant $t = 0$, the network predicts the outcome of pulling the rake on each tray by calculating $\Pi_{t=0,1} O(S^*(t))$, $* \in \{R, L\}$, in which $S^*(0)$ indicates the current configuration that is given from the environment, and $S^*(1)$ indicates the predicted configuration that can be calculated by using the network's output $O(S^*(0))$. Thus, if $\Pi_{t=0,1} O(S^*(t))$ is a large value, it would indicate that the agent is likely to obtain the food along with the rake. The agent should choose to pull the rake on the tray with

larger $\prod_{t=0,1} O(S^*(t))$ more often. We make the agent select the right tray with the probability of

$$P(R) = \prod_{t=0}^1 O(S^R(t)) / (\prod_{t=0}^1 O(S^R(t)) + \prod_{t=0}^1 O(S^L(t))). \quad (4)$$

3 Results

Initially, task 1 was assigned to agents 1 and 2. Fig.3A shows the learning curves. Each curve indicates the average of five runs of simulation. The error bars indicate the standard deviation.

We define *a learning time* as that at which the neural network learns a target function from a given pair of an input and a teacher signal. In each trial, there exists one learning time for agent 1 and two learning times for agent 2.

To learn to solve task 1, both the agents required almost the same amount of learning times. For the success rate to exceed 0.9, agent 1 required 3,830 sessions, which amounts to 22,980 learning times, and agent 2 required 1,848 sessions, which amounts to 22,176 learning times (Fig. 3A).

After 20,000 sessions of task 1 (by which the agent learned to solve the task without errors), each agent was assigned task 2.

A considerably low amount of time was required for agent 2 to adapt to task 2 as compared to that required by agent 1 (Fig. 3B). For the success rate to exceed 0.9, agent 1 required 2,222 sessions (4,444 learning times) and agent 2 required only 20 sessions (80 learning times).

To be fair, we conducted another simulation experiment. In the experiment above, the number of experienced sessions of task 1 before proceeding to task 2 was the same (20,000) for both the agents. However, the learning times of task 1 were different (120,000 for agent 1 and 240,000 for agent 2). To ensure that the difference in the learning times would not be the reason for agent 2's quicker adaptation to task 2 as compared to that of agent 1, we assigned twice as many sessions of task 1 to agent 1. No significant change was found in the results (not shown).

We examined the reason why agent 2 was quicker when adapting to the task change as compared to the response of agent 1 by looking into the synapse weights and their changes during the tasks. In task 1, agent 2 was assigned a larger variety of (input, teacher) pairs as compared to those assigned to agent 1, because agent 2 learned the step-by-step change in the environment's state while agent 1 learned only the outcome in each trial (Section 2.3). Hence, the internal model of agent 2 became almost the same function as the relevant part of the state transition function E_δ , whereas the internal model of agent 1 became a function that was not easily understood (although the prediction of the outcome was perfectly correct for the training set). When task 2 was assigned, a significant difference was observed in the changes of the synapse weights. For agent 2, the task change caused only a small modification to the weights of the synapses related to the x-coordinates of the rake and the food, because E_δ remained the same in tasks 1 and 2 (the small modifications were necessary because novel values of the x-coordinates were assigned to the agent in task 2). For agent 1, the new function adopted to learn was

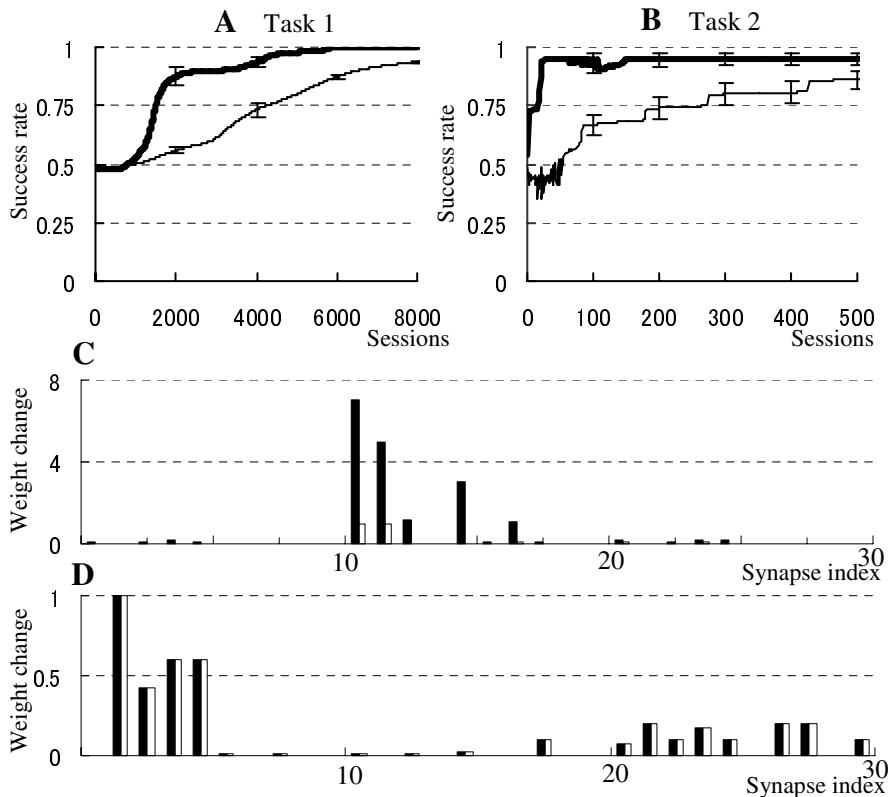


Fig. 3. (A) and (B): Learning curves of the agents during (A) task 1 and (B) task 2. The thin and thick lines show the results obtained for agents 1 and 2, respectively. The abscissa indicates the number of experienced sessions. The ordinate shows the rate of obtaining the correct answer. (C) and (D): The weight change in each synapse of (C) model 1 and (D) model 2 during the initial 30 sessions of task 2. Each black and white bar indicates the sum of the weight changes at each learning time and the sum of the weight changes at each learning time of each synapse, respectively (note that there are 30 synapses in each model), where $|\cdot|$ denotes the absolute value. The synapse weights oscillated in model 1 and not in model 2.

significantly different from the learned function. In fact, some synapse weights oscillated for a long time, indicating that it was difficult for model 1 to assign the responsibility of the prediction error to proper synapses.

Fig. 3 (C) and (D) show the weight change of each synapse during the initial 30 sessions of task 2, for model 1 and 2, respectively. In Fig. 3(C), the black bars were significantly taller than the white bars, indicating that the weights oscillated in model 1. In Fig. 3(D), the black bars were almost as tall as the white bars, indicating that the weights did not oscillate in model 2. In summary, agent 2 was quicker to adapt because the dynamics of the environment that model 2 learned in task 1 was common in task 2 and hence only a small and straightforward modification of the synapse

weights was required. Agent 1 was slower because it had to re-learn a new function and finding the responsible synapses took time.

4 Discussion

In this study, two agents with different internal models were compared. The first agent had an internal model (model 1) that predicted the outcome (success or failure) of a given task. The second agent had a different internal model (model 2) that predicted a step-by-step state transition of the environment. After each agent completely learned to solve a task, it was assigned a slightly modified task. The agent with model 2 was much quicker to adapt to solving the new task as compared to the agent with model 1.

Thus, this study provides a concrete example of how a particular difference in the internal models leads to a large difference in tool-use behavior. It is suggested that the generalization (i.e., adaptation to task change) capability of tool-use behavior of each primate species may possibly depend on the internal model available to the species. The species who predict a step-by-step state transition of the environment may be more adaptive than those who predict only the outcome.

Further study is necessary in several directions.

First, we must analyze the agents mathematically. In Section 3, we have examined the reason for the different adaptation speeds of the two internal models in a particular case. By means of more extensive mathematical analyses, we will be able to determine the sufficient and necessary conditions for this difference to occur in more general cases. It must be noted that although we have employed a three-layered neural network model with the back propagation method in this study, we do not intend to argue that a particular neural network and the learning method are important for us to produce the different adaptation speeds. Currently, we cannot deny the possibility that the network structure is causally related to the results obtained. It should be proven that such specific models and methods are not crucial for us to produce the results.

Second, we must investigate other types of internal models. The two models employed in this study are extreme examples of internal models. For example, in model 1, we assumed that the agent could not use any information regarding the environment's dynamics. However, even monkeys are able to recognize the dynamics to some extent [4]. Hence, the question is the extent of their abilities. It would be interesting to study internal models whose capability of learning dynamics can be modulated by a parameter.

Finally, we must compare the simulation results with the actual behavior of the primates. For example, Santos et al. [12] studied the adaptation capability of *tamarins* by employing some tasks similar to those in our study. After learning the simple rake-use tasks, they were asked to solve a slightly different task in which the rake was well connected on one tray while the rake was poorly connected (the rake grip was separated from the body) on the other tray. For humans, it is easy to predict that the food pellet is not accessible by the poorly connected rake. However, tamarins made a large number of errors before they correctly selected the well-connected rake. Thus, it is probable that they could not understand the geographic dynamics of the rake, unlike humans who could understand them. These results support our findings. Currently, we are conducting a simulation study by employing their task in order to make a direct comparison.

By means of further theoretical studies, the mappings from differences in possible underlying functions in the brain to differences in observable tool-use behavior should be clearly understood. These mappings could provide us with important insight into the actual brain functions that each primate species employs. It is extremely difficult to investigate the underlying brain functions of non-human primates by means of experimental methods because their languages are very poor and also because invasive experiments are limited. Therefore, theoretical studies should significantly contribute to understanding primates' intelligence.

Acknowledgments

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

References

1. Köhler, W.: *The mentality of apes*. Brace and Company, New York (1925)
2. Goodall, J.: *The Chimpanzees of Gombe: Patterns of Behavior*. Harvard Univ. Press, Cambridge (1986)
3. Whiten, A., Goodall, J., McGrew, W.C., Nishida, T., Reynolds, V., Sugiyama, Y., Tutin, C.E.G., Wrangham, R.W., Boesch, C.: Cultures in chimpanzees. *Nature* 399, 682–685 (1999)
4. Beck, B.B.: *Animal tool behavior: the use and manufacture of tools by animals*. Garland STPM Press, New York (1980)
5. Matsuzawa, T.: *Primate origins of human cognition and behavior*. Springer, Heidelberg (2001)
6. Sumita, K., Kitahara-Frisch, J., Norikoshi, K.: The acquisition of stone tool use in captive chimpanzees. *Primates* 26, 168–181 (1985)
7. Torigoe, T.: Comparison of object manipulation among 74 species of Non-human primates. *Primates* 26(2), 182–194 (1985)
8. Wolpert, D.M., Ghahramani, Z., Jordan, M.I.: An internal model for sesorimotor investigation. *Science* 269, 1880–1882 (1995)
9. Shadmehr, R., Mussa-Ivaldi, F.A.: Adaptive Representation of Dynamics during learning of a motor task. *J. Neurosci.* 12(5), 3208–3224 (1994)
10. Imamizu, H., Miyauchi, S., Tamada, T., Sasaki, Y., Takino, R., Puetz, B., Yoshioka, T., Kawato, M.: Human cerebellar activity reflecting an acquired internal model of a novel tool. *Nature* 403, 192–195 (2000)
11. Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. *Neural Networks* 11, 1317–1329 (1998)
12. Santos, L.R., Rosati, A., Sproul, C., Spaulding, B., Hauser, M.D.: Means-means-end tool choice in cotton-top tamarins (*Saguinus oedipus*): finding the limits on primates' knowledge of tools. *Anim. Cogn.* 8, 236–246 (2005)
13. Hauser, M.D.: Artifactual kinds and functional design features: what a primate understands without language. *Cognition* 64, 285–308 (1997)
14. Rumelhart, D.M., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.M., McClelland, J.L. (eds.) *Parallel distributed processing: exploration in the microstructure of cognition. Foundations*, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)

Artifact Removal Using Simultaneous Current Estimation of Noise and Cortical Sources

Ken-ichi Morishige¹, Dai Kawawaki², Taku Yoshioka², Masa-aki Sato²,
and Mitsuo Kawato²

¹ Department of Intelligent Systems Design Engineering,
Toyama Prefectural University, Toyama, Japan

² ATR Computational Neuroscience Laboratories, Kyoto, Japan
kmorishi@pu-toyama.ac.jp,
{dai-kaw,taku-y,masa-aki,kawato}@atr.jp

Abstract. The measurement of magnetoencephalographic (MEG) signals is contaminated by large magnetic artifacts, such as heart beats, eye movements, and muscle activities, and so on. These artifacts can be orders of magnitude larger than the signal from the brain, thus making cortical current estimation extremely difficult. This paper proposes a novel method to remove the effects of artifacts by simultaneously estimating the cortical and artifactual dipole currents. By using proper prior information, we show that this method can estimate the currents of artifacts and cortical activities simultaneously, and the estimated cortical currents are more reasonable in comparison to those of previous methods.

1 Introduction

Neuronal currents induce weak transient magnetic fields that can be measured by magnetoencephalography (MEG). This measurement is contaminated by various kinds of artifacts, such as heart beats, eye movements, muscle activities, and so on. These artifacts can be orders of magnitude larger than the signal from the brain, thus making cortical current estimation extremely difficult.

It is now widely accepted that the statistical methods such as independent component analysis (ICA) and principal component analysis (PCA) are useful tools for isolating artifacts and cortical activities from MEG data. These statistical methods could be efficient techniques. However, they might mistakenly remove the effect of brain activities if the statistical property resembles the artifact.

Berg et al. proposed a new method that put current dipoles as the source of eye artifacts, and removed the estimated MEG artifacts from observed MEG [1]. This method might remove the components of brain activities because the observed MEG data contains not only the effects of artifacts but also brain activities.

To overcome these problems, Fujiwara et al. tried simultaneously estimating cortical currents and eye artifacts [2]. Actually, the observed data were contaminated not only by single sources of an artifact but also by such multiple sources as heart beats, eye movements, muscle activities, and so on. If the observed MEG data were

contaminated by multiple artifacts, it should be more appropriate to simultaneously estimate the brain activities and multiple artifacts from the observed MEG data.

This paper proposes a novel method to remove the effects of artifacts by simultaneously estimating the cortical and artifactual dipole currents. By using proper prior information, we show that this method can estimate the currents of artifacts and cortical activities simultaneously, and the estimated cortical currents are more reasonable in comparison to those of previous methods.

2 Simultaneous Estimation of Artifactual and Cortical Currents

2.1 Current Dipole Model

The cortical neuronal sources are modeled as L_{brain} current dipoles. The artifact sources are modeled as $L_{artifacts}$ current dipoles. The relationship between the amplitudes of the current dipoles and the observed magnetic fields at time point τ is given by

$$\mathbf{B}(\tau) = \mathbf{G}_{brain} \cdot \mathbf{J}_{brain}(\tau) + \mathbf{G}_{artifacts} \cdot \mathbf{J}_{artifacts}(\tau) + \mathcal{E}(\tau), \quad (1)$$

where $\mathbf{B}(\tau)$ is an N -element vector for the observed magnetic field, $\mathbf{J}_{brain}(\tau)$ is an L_{brain} -element vector for the cortical current, and $\mathbf{J}_{artifacts}(\tau)$ is a $L_{artifacts}$ -element vector for the artifact current. \mathbf{G}_{brain} is an N -by- L_{brain} matrix, and $\mathbf{G}_{artifacts}$ is an N -by- $L_{artifacts}$ matrix. \mathbf{G}_{brain} and $\mathbf{G}_{artifacts}$ are referred to as the lead field. Lead field \mathbf{G}_{brain} is calculated by using the Sarvas equation that is derived for a single spherical shell model. The cortical dipole current directions are assumed to be perpendicular to the cortical surface. A positive current was defined as one directed toward the inside of the cortex. The $\mathbf{G}_{artifacts}$ is the lead field matrix for the artifact sources. Current dipoles were located at the artifactual sources, and estimated the amplitudes and the directions. The $\mathbf{G}_{artifacts}$ is calculated by the Biot-Savart equation. Observation noise $\mathcal{E}(\tau)$ is assumed to obey a Gaussian distribution with zero mean.

2.2 Artifactual Sources

Our approach places dipoles not only on cortical surfaces but also on artifactual sources. The eyeball noise source can be described as a fixed dipole with a positive pole at the cornea and a negative pole at the retina. Other factors, such as the heart beat and muscle activations, can cause artifacts. This paper assumes that these areas are also artifactual sources. The artifact current dipoles were located in the center of right and left eyeballs, the heart, and the carotid arteries. The cortical and artifact currents were estimated simultaneously by using Hierarchical Bayesian estimation (VBMEG) [3],[4].

2.3 Hierarchical Bayes Estimation

This paper used the Hierarchical Bayes estimation method for dipole current estimation [3],[4]. The probability distribution, that the magnetic field \mathbf{B} is observed for given currents \mathbf{J} , is given by

$$P(\mathbf{B} | \mathbf{J}) \propto \exp\left[-\frac{\beta}{2}\|\mathbf{B} - \mathbf{G} \cdot \mathbf{J}\|^2\right], \quad (2)$$

where β denotes the inverse of the noise variance. In the Bayesian method, a prior assumption on source currents \mathbf{J} is imposed as a prior probability distribution $P_0(\mathbf{J})$. The Bayesian method calculates the posterior distribution for source current, $P(\mathbf{J} | \mathbf{B})$, which represents the probability that the source current is \mathbf{J} under observed MEG data \mathbf{B} and the prior information:

$$P(\mathbf{J} | \mathbf{B}) = \frac{P(\mathbf{B} | \mathbf{J})P_0(\mathbf{J})}{P(\mathbf{B})}. \quad (3)$$

The hierarchical Bayesian method assumes a hierarchical prior distribution for the current as follows:

$$\begin{aligned} P_0(\mathbf{J} | \boldsymbol{\alpha}, \beta) &\propto \exp\left[-\frac{\beta}{2}\mathbf{J}^T \cdot \mathbf{A} \cdot \mathbf{J}\right] \\ P_0(\boldsymbol{\alpha}) &= \prod_i \Gamma(\alpha_i | \bar{\alpha}_{0i}, \gamma_{0i}) \\ \Gamma(\alpha | \bar{\alpha}, \gamma) &\equiv \alpha^{-1}(\alpha\gamma/\bar{\alpha})^\gamma \Gamma(\gamma)^{-1} e^{-\alpha\gamma/\bar{\alpha}} \\ P_0(\beta) &\propto \beta^{-1}, \end{aligned} \quad (4)$$

where i is the dipole index, \mathbf{A} is the diagonal matrix whose diagonal elements are $\boldsymbol{\alpha} = \{\alpha_i | i = 1, \dots, (L + K)\}$, α_i^{-1} means the prior current variance of J_i . $\mathbf{G} = [\mathbf{G}_{brain} \ \mathbf{G}_{artifacts}]$, $\mathbf{J} = [\mathbf{J}_{brain} \ \mathbf{J}_{artifacts}]^T$, and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_{brain} \ \boldsymbol{\alpha}_{artifacts}]$.

The mean and degree of the freedom of the Gamma distribution $\bar{\alpha}_{0i}$ and γ_{0i} , respectively, are referred to as hyperparameters. In this study, these values were determined from fMRI information, a variance magnification parameter m_0 , and confidence parameter γ_0 . γ_{0i} was set to γ_0 for all dipoles, while $\bar{\alpha}_{0i}$ was set based on the fMRI t-value at each dipole and baseline variance $V_{0,base}$.

$$\bar{V}_{0i} = V_{0,base} + (m_0 - 1) \cdot V_{0,base} \cdot \hat{t}_i^2, \quad (5)$$

where $\bar{V}_{0i} \equiv \alpha_{0i}^{-1}$ and \hat{t}_i are normalized fMRI dipole t-values with a maximum value of 1.

A larger m_0 means a higher probability of current variance. γ_0 controls the reliability of the hierarchical prior distribution. A larger γ_0 means larger reliability of the hierarchical prior distribution.

2.4 Hyperparameters

For estimation of cortical dipole currents, the fMRI information can be imposed as prior information on the variance distribution. The hyperparameter values $m_0 = 500$ and $\gamma_0 = 500$ were employed for the cortical dipoles to estimate the prior current variance.

Unlike the cortical dipoles, less knowledge about the current variance of artifacts was obtained a priori. Since the signal magnitudes of the cortical and artifact sources markedly differed with several orders of magnitudes, our method set a different prior current variance for each dipole source, thus giving our approach the ability to simultaneously estimate the proper cortical and artifact dipole currents. These values of prior current variances and the hyperparameters for artifactual sources were estimated based on the two criteria: (1) the prior and posterior variances should be the same and (2) high correlation coefficients between the spatial patterns of fMRI activities and that of the MEG estimation obtained as the magnitude of the estimated mean current variance. With varying hyperparameters for artifactual sources, we calculated the two indices and took the best values of the hyperparameters (table 1). We set the rough information about the prior current variance using these hyperparameters, and estimated the dipole current.

Table 1. Hyperparameters for artifactual sources

Artifactual source	direction	curr. var. [nAm] ²	γ_{0i}
left eye	x	10^2	$10^{4.5}$
	y	10^2	$10^{4.5}$
	z	10^2	$10^{4.6}$
right eye	x	10^2	$10^{4.3}$
	y	10^2	$10^{4.2}$
	z	10^2	$10^{4.5}$
Heart	x	10^4	$10^{4.2}$
	y	$10^{3.9}$	10^3
	z	$10^{4.2}$	$10^{3.6}$
Left carotid artery	x	10^2	10^3
	y	$10^{2.6}$	$10^{4.7}$
	z	10^2	$10^{4.3}$
Right carotid artery	x	$10^{2.5}$	10^4
	y	10^2	$10^{4.3}$
	z	$10^{3.9}$	$10^{4.9}$

3 Experiments

3.1 Experimental Settings

Three subjects participated in the experiments, which were approved by ATR's Human Subjects Review Committee. All subjects gave informed consent and had normal or corrected-to-normal visual activity.

A whole-head 208-channel MEG system (PQ1400RM, Yokogawa Co., Japan) was used for the MEG recordings. The sampling frequency was 1000 Hz. Electrocorticograms (EOGs) were simultaneously recorded.

Subjects were instructed to pursue covertly the target motion while orienting their attention to the fixation point. Three patterns of the target motion were prepared. These target motions were basically sinusoidal, and the trial frequency was changed when the target velocity was zero. The mean trial frequency was 0.5 Hz, and the amplitude was 4°. Target motions were randomly presented. One session consisted of thirty repetitions of a trial for three target motions. Each subject carried out 270 trials (30 trials × 3 target motions × 3 sessions = 270 trials).

Structural and functional MR images were obtained with a 1.5-T MR scanner (Shimadzu Marconi, MAGNEX ECLIPSE 1.5T Power Drive 250). A block design was used for the fMRI experiment. One session consisted of four repetitions of a task block and a rest block. In the task block, subjects covertly pursued the moving target for four seconds. After this period, they were allowed a one-second rest. These periods were repeated alternately three times (15 seconds). In the rest block, the target was presented in the center of the screen for 15 seconds. Five sessions were conducted. A target motion was generated in the same way as in the MEG experiment.

3.2 Data Analysis

The drift components of the observed MEG data were removed by a reference sensor. The baselines of the MEG signals were corrected by subtracting the mean of the pre-stimulus intervals from averaged signals. Trials containing eye blinks were discarded. If a subject fell asleep during a session, the session's observed data were rejected. Then remaining trials of subjects A, B, and C were 62.6%, 93.0%, and 98.9%, respectively. These remaining data were used for this analysis.

The fMRI data were preprocessed and analyzed using SPM5 (The Wellcome Department of Cognitive Neurology) to obtain prior information of variances of brain current sources.

4 Results

To investigate the validity of our approach, we applied our denoising method to the MEG data obtained by subjects who were instructed to covertly pursue a target motion.

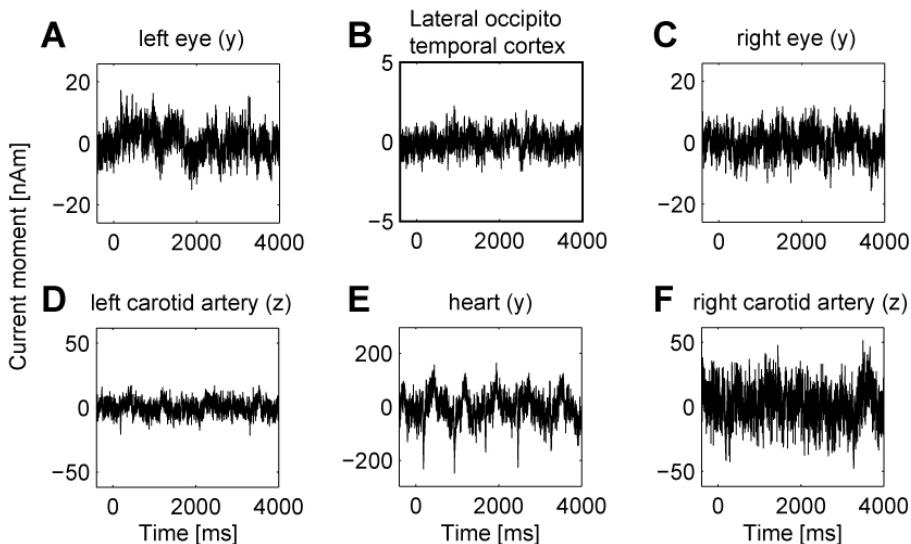


Fig. 1. Results of estimated dipole currents. **B** shows the waveform of peak dipole current of Lateral occipito temporal cortex. **A, C, D, E, and F** show the estimated current waveform of left eye, right eye, left carotid artery, heart, and right carotid artery, respectively.

Figure 1 shows examples of estimated currents. It is known that heart beats have a characteristic to have electrical impulse patterns in cycles, and amplitude was larger than those of others. When the heart gets smaller in size, it makes an electrical impulse. Estimated current at the heart can express the character of electrical impulse in cycles (Fig. 1E).

Next, we investigated the validity of spatial pattern of estimated cortical currents. If well isolating the brain and artifactual activities, it is expected that the estimate of cortical current is more accurate. Since the true brain activity is unknown, we assumed that the temporal average of the true brain activity is proportional to the fMRI signals. Given this assumption, the correlation coefficient between the fMRI signals and the spatial patterns of the dipole current intensity was used as criteria for the estimation results. This correlation between them was calculated by using the following equation:

$$J_{n,i} = \frac{1}{T} \sum_{t=1}^T |J_{i,n}(t)| \quad (6)$$

$$Corr_n = \frac{\sum_{i=1}^I J_{n,i} \cdot f_i}{\sqrt{\sum_{i=1}^I J_{n,i}^2} \sqrt{\sum_{i=1}^I f_i^2}} \quad (7)$$

where J is estimated dipole current, f_i is fMRI signals (t -values), I is the number of cortical current dipoles, T is a sampling number, and N is a trial number. First,

temporal average of current intensity was calculated (Eq.(6)). Second, correlation coefficient was calculated for each single trial (Eq. (7)).

To compare to the other denoising methods, four types of spatial pattern were calculated. First method is “VBMEG”, that only located the dipoles on the cortical surface and estimate the dipole currents. Second one is “PCA-pruned”, that estimated dipole currents from MEG data which artifact components were removed by using PCA. Third one is “ICA-pruned”, that estimated from MEG data which artifact components were removed by using ICA. Last one is “VBMEG with extra-dipole”, that places dipoles not only on cortical surfaces but also on artifact sources, and simultaneously estimate the dipole currents.

We investigated which estimated current intensity was most similar to the spatial pattern of fMRI signals. A randomized block ANOVA found a significant statistical differences among them ($F(3,687)=6.85$, $P<0.05$, Fig. 2). Turkey’s HSD comparison showed that the correlation coefficient of VBMEG-with-extra-dipole was significantly larger than others. The spatial pattern of estimated current intensity using our proposed method was most similar to the spatial patterns of fMRI signals.

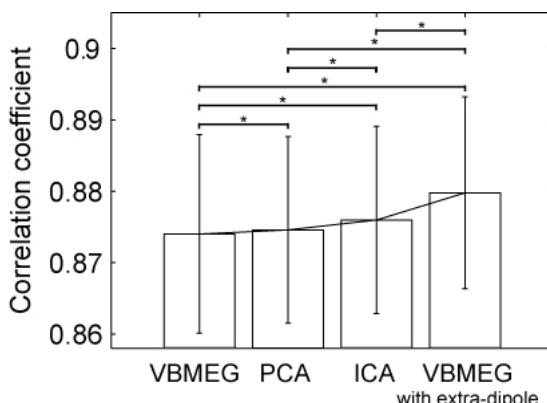


Fig. 2. Comparison of correlation coefficients. Data represent means \pm SD. A randomized block ANOVA found a significant statistical differences among them ($F(3,686)=625.41$, $P<0.001$). Turkey’s HSD comparison showed that the correlation coefficient of VBMEG-with-extra-dipole was significantly larger than others. * denotes a significant level of $P<0.01$.

5 Conclusion

This paper proposed a novel method to remove the effect of artifacts by simultaneously estimating the cortical and artifactual dipole currents. Since the signal magnitudes of the cortical and artifact sources differed, our method set a different prior current variance for each dipole source, thus giving our approach the ability to simultaneously estimate the proper cortical and artifact dipole currents.

To examine the performance of the proposed approach, we tried removing the artifacts from real MEG data. The estimation results consist with known physiological findings and express the spatial and temporal characters well. Our results indicate that

this approach was more reasonable and effective than existing methods for removing artifacts contaminating MEG signals.

Acknowledgment

This research was supported by a contract with the National Institute of Information and Communications Technology entitled, 'Multimodal integration for brain imaging measurements'.

References

1. Berg, P., Scherg, M.: Dipole models of eye movements and blinks. *Electroencephalogr. Clin. Neurophysiol.* 79, 36–44 (1991)
2. Fujiwara, Y., Yamashita, O., Kawakami, D., Doya, K., Kawato, M., Toyama, K., Sato, M.: A hierarchical Bayesian method to resolve an inverse problem of MEG contaminated with eye movement artifacts. *NeuroImage* (2008) (to appear)
3. Sato, M., Yoshioka, T., Kajiwara, S., Toyama, K., Goda, N., Doya, K., Kawato, M.: Hierarchical Bayesian estimation for MEG inverse problem. *Neuroimage* 23(3), 806–826 (2004)
4. Yoshioka, T., Toyama, K., Kawato, M., Yamashita, O., Nishina, S., Yamagishi, N., Sato, M.A.: Evaluation of hierarchical Bayesian method through retinotopic brain activities reconstruction from fMRI and MEG signals. *Neuroimage* 42(4), 1397–1413 (2008)

Significance for Hippocampal Memory of Context-Like Information Generated in Hippocampal CA3c

Toshikazu Samura¹, Motonobu Hattori², Shinichi Kikuchi¹, and Shun Ishizaki¹

¹ Graduate School of Media and Governance, Keio University
5322 Endo, Fujisawa-shi, Kanagawa, 252–8520, Japan
{samura,kikuchi,ishizaki}@sfc.keio.ac.jp

² Interdisciplinary Graduate School of Medicine and Engineering,
University of Yamanashi
4-3-11 Takeda, Kofu-shi, Yamanashi, 400–8511, Japan
m-hattori@yamanashi.ac.jp

Abstract. It has been suggested that the cooperation between spatial and temporal selectivities of hippocampal CA3 contributes to sequence disambiguation, which is one of essential functions of the hippocampus. We demonstrate the validity of the suggestion by using sequences assumed from a modified T-maze in which rats perform a task. It should be noted here that CA3c, which is one of the CA3 subregions, generates context-like information that contributes to sequence disambiguation. Finally, we discuss the significance of the context-like information generated in CA3c from a structural and functional viewpoint and suggest that the feedforward and feedback of the context-like information play an essential role in the memory formation of the hippocampus by affecting spatial and temporal information processing in hippocampal dentate gyrus and CA1, respectively.

1 Introduction

Eichenbaum suggested that daily episodes are memorized as a relational network in the hippocampus [1]. In the relational network, an episode is expressed by a sequence of primitive elements of memory. Episodes are associated with each other by events shared among them. For example, let us consider that the hippocampus memorized two episodes, one composed of events A, B, and C ($A \rightarrow B \rightarrow C$), the other composed of events: X, B, and Y ($X \rightarrow B \rightarrow Y$), where event B associates the two episodes. It is then difficult to decide which pattern should be retrieved from event B. Such ambiguity of sequences becomes a problem for retrieval. Thus, sequence disambiguation is an essential function for retrieving the original episodes. Levy suggested the critical microscopic properties that allow the network to create suitable codes for solving sequential problems including sequence disambiguation [2]. The hippocampal CA3 region has such properties; therefore, they demonstrated its sequence disambiguation ability by a simplified CA3 model. The CA3 region has unique recursive axons that are

called recurrent collaterals (RCs). Its uniqueness led many researchers to focus on it, producing many computational models of CA3. Meanwhile, recent studies showed considerable detailed properties at the hippocampal subregional level. We focused on location-dependencies of CA3 subregions (CA3a, CA3b, and CA3c) and suggested that CA3 can be divided into autoassociative and heteroassociative memory [3]. We also demonstrated that the functional division of CA3 contributes to sequence disambiguation [4]. Moreover, we suggested that CA3 can be divided into two regions that show spatial or temporal selectivity rather than autoassociative or heteroassociative memory and demonstrated by computer simulations [5] that the divided CA3 contributes to sequence disambiguation. In this study, we demonstrate the validity of the suggestion in more plausible conditions using sequences assumed from a modified T-maze [6] that rats perform in the real world. Finally, we discuss the significance of the context-like information generated in CA3c for hippocampal memory formation from a structural and functional viewpoint.

2 Location-Dependencies in Hippocampal CA3

CA3 is a unique region in the hippocampus in that it has recurrent collaterals (RCs) that recursively connect CA3 neurons to other CA3 neurons, as shown in Fig. II. Each CA3 subregion differs from others in the projections of RCs. Fig. II(a) shows the relationship between the location of a neuron and the projection of its RCs [7]. First, the RCs of CA3c neurons are limited to the area surrounding them. Second, the RCs of CA3b neurons are widely spread. Finally, the RCs of CA3a neurons are limited to CA3a and CA3b. Moreover, the dendritic locations of RCs depend on the positions of a postsynaptic neuron [7]. As shown in Fig. II(b), CA3a and CA3b neurons tend to receive RCs on dendrites near a soma (proximal dendrites), whereas CA3c neurons tend to receive them on dendrites distant from a soma (distal dendrites). Spike-timing dependent plasticity (STDP) is a rule for changing synaptic weights in the hippocampus. A recent study suggested that STDP turns an asymmetric profile into a symmetric one depending on the density of inhibitory interneurons [8]. A symmetric profile STDP (SSTDP) was observed from a high-density area near a soma, whereas an asymmetric profile STDP (ASTDP) was observed from a low-density area distant from a soma. Furthermore, Xu *et al.* suggested that long-term potentiation (LTP) induction at distal dendrites increases the linearity of summation only for inputs arriving within a narrow time window; on the other hand, LTP induction at proximal dendrites increases the linearity of summation for inputs arriving within a long time window [9]. Thus, a distal dendrite shows spatial summation and a proximal one shows temporal summation after repeated LTP induction.

3 Spatial and Temporal Selectivity in Hippocampal CA3

In CA3a and CA3b where neurons receive RCs at proximal dendrites, temporal summation and SSTDP coexist. Under temporal summation, neurons can fire

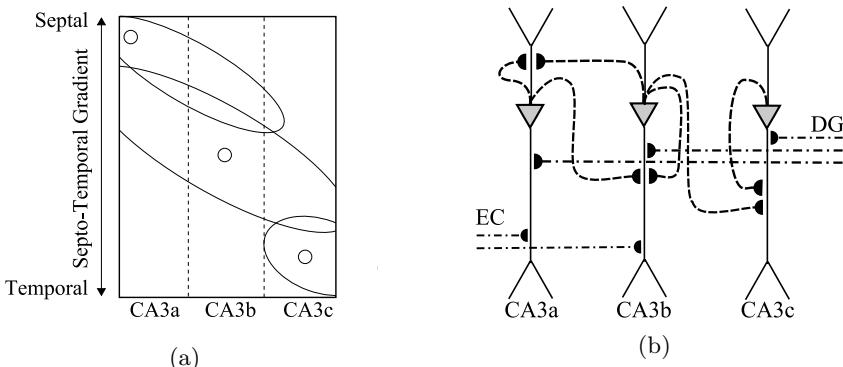


Fig. 1. Structure of CA3. (a) Projections of RCs (circle: source neuron, ellipse: projection of circled neuron in it). (b) Connections within CA3 and its dendritic locations (inverse triangle: soma, forked line: dendrite, dashed line: RCs, chain line: connections from external regions).

even if they receive inputs discontinuously. Moreover, under SSTDP, simultaneous firing leads to potentiation of synaptic weights and time lag leads to their depression. If neurons simultaneously fire, then the relationship between the neurons is mapped onto synaptic weights as that of firing patterns of the network. Thus, when a postsynaptic neuron receives inputs from neurons that compose the same firing pattern, it is likely to be activated regardless of the firing order of these neurons. Consequently, CA3a and CA3b are sensitive only to spatial information of inputs: a neuron in CA3a and CA3b can fire when it receives inputs from neurons that compose the same firing pattern, regardless of the timing and order of inputs. Spatial summation and ASTDP coexist in CA3c where neurons receive RCs at distal dendrites. Under spatial summation, neurons can fire when they receive inputs within a narrow time window. Moreover, ASTDP potentiates synapses when postsynaptic neurons fire after presynaptic firings. Conversely, synapses between them are depressed if the order of their firing is reversed. As a result, synaptic weights can reflect the order of firing. Therefore, when a neuron receives inputs through potentiated synapses from presynaptic neurons in the memorized order, the neuron is likely to be activated. Consequently, CA3c is sensitive to temporal information of inputs in that neurons in CA3c can fire when they receive inputs from a memorized set of neurons in a memorized order within a short time window.

4 Computer Simulations

4.1 Conditions

In this simulation, we used the same model as that described in reference [5], where its parameters were changed to improve its biological validity. The hippocampal CA3 is represented by Izhikevich's neuron model [10], and the neurons

are interconnected by recurrent collaterals as illustrated in Fig. 1. Proximal dendrite and distal one differ each other in the STDP profile and the time constant of EPSP. The details of the neurons, anatomical properties, and learning rules are described by Equations (1)–(13) in reference [5]. Its parameters in the equations are listed in Table 1. We evaluated the CA3 model memorizing sequences assumed from a T-maze modified by Wood *et al.* [6]. In the modified T-maze, rats perform a continuous task in which they traverse the central stem of the maze and then decide on a left or right turn at a T junction (Fig. 2). Following the choice, they return to the base of the stem via each arm, and then traverse the central stem again. In each choice, rats are required to go in the opposite direction to the last choice. Here, we defined the looped sequence ($\rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow A \rightarrow$) by assigning seven patterns (A–G) to each point on the route that rats pass, as shown in Fig. 2. Each pattern was represented by the activation of 90 neurons with no overlap among them. In this sequence, ambiguity arises at pattern C. Therefore, the model had to determine what patterns needed to precede pattern C to solve the ambiguity. In this simulation, the model memorized the sequence by applying it twice (learning phase). Following the phase, we confirmed that the model could output information that leads to disambiguation (retrieving phase). The connections from EC to CA3 are required for retrieval [11], and EC connects only to CA3a and CA3b [12]. Thus, inputs are limited to them during the retrieving phase. During both phases, input sequences were applied to the model by the conventional theta-phase procedure [4].

Table 1. Parameters for simulation

W	21	H	35	k	1.75	v_r	-55.0	v_t	-40.0	C	80
a	0.021	b	-1.7	c	-38.0	d	190.0	v^{peak}	10.0	w_{\min}	1.0×10^{-7}
w_{\max}	7.0	$w^{\text{DG-EC}}$	6.0	θ	0.25π	ι	30	κ	6	λ_{\min}	0.5
λ_{\max}	2.5	η_{CA3a}	0	η_{CA3b}	0	η_{CA3c}	10	δ_{\max}	10	β	0.08
α_{proximal}	66.0	τ_{proximal}	5.0	α_{distal}	72.5	τ_{distal}	1.5				

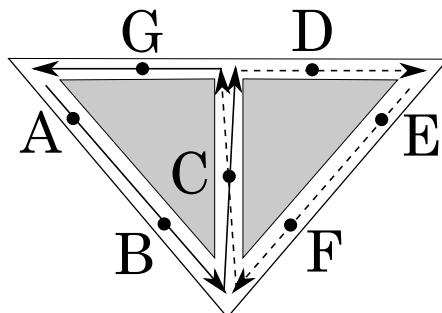


Fig. 2. Relationship between patterns and locations in modified T-maze

4.2 Results

Figs. 3(a) and (b) show a similarity given by the direction cosine between the model output and each pattern when a part of the looped sequence (subsequence α : G→A→B→C, subsequence β : D→E→F→C) was applied to the model during the retrieving phase. The first pattern (G or D) was inputted to CA3a and CA3b at the beginning. After that, subsequent patterns were applied to them in order of patterns of each subsequence every 10 unit times. Although each pattern was applied in each subsequence only once, they showed periodic activation of patterns in CA3a and CA3b as shown in these figures. Then, we compared the similarities of CA3c output in two subsequences. When pattern C of subsequence α was applied to the model, CA3c outputted pattern D (Fig. 3(a), time 3560). When pattern C of subsequence β was applied, CA3c outputted pattern H (Fig. 3(b), time 3720). These results mean that the model generated different activities according to the differences between the subsequences in spite of the same pattern C.

5 Discussion

In this simulation, previously inputted patterns were periodically retrieved in CA3a and CA3b. Thus, they could buffer the differences between sequences. The information in the buffer was transmitted to CA3c through the connections that are sensitive to temporal information. Therefore, the temporal differences in the buffer caused the difference in the retrieved pattern of CA3c. This CA3c output is a code that dissociates the same pattern according to previous inputs. In this study, we showed the generation of the code due to the selectivities of the hippocampus CA3 even where the sequence that may occur in a modified T-maze [6] was used. Consequently, we have validated our suggestion that the cooperation between spatial and temporal CA3 selectivities contributes to sequence disambiguation. Levy *et al.* suggested that the network must create a code for a small subsequence of patterns in order to solve the disambiguation problem and showed that their simplified CA3 model is able to do it [13]. They called the neuron representing such a code a local context neuron. In this study, we showed that CA3c can react to subsequences stored in CA3a and CA3b. Thus, we regard a neuron of CA3c as a local context neuron. Consequently, the cooperation between the selectivities of CA3 suggested a mechanism for generating context-like information. In turn, this suggests that context-like information is generated in the hippocampal CA3c. Moreover, Levy derived the significance of context-like information in order to solve sequential problems including the disambiguation from the computer simulations [2]. However, if the context-like information is generated in CA3c, CA3 cannot use the information because there are no pathways from CA3c to CA3a and CA3b (Fig. 1(b)) [7]. The context-like information is important for solving the problems, but where is the information used? In general, information is processed in the hippocampus in the order: DG→CA3→CA1. In particular, CA3c has strong projection to CA1 as compared

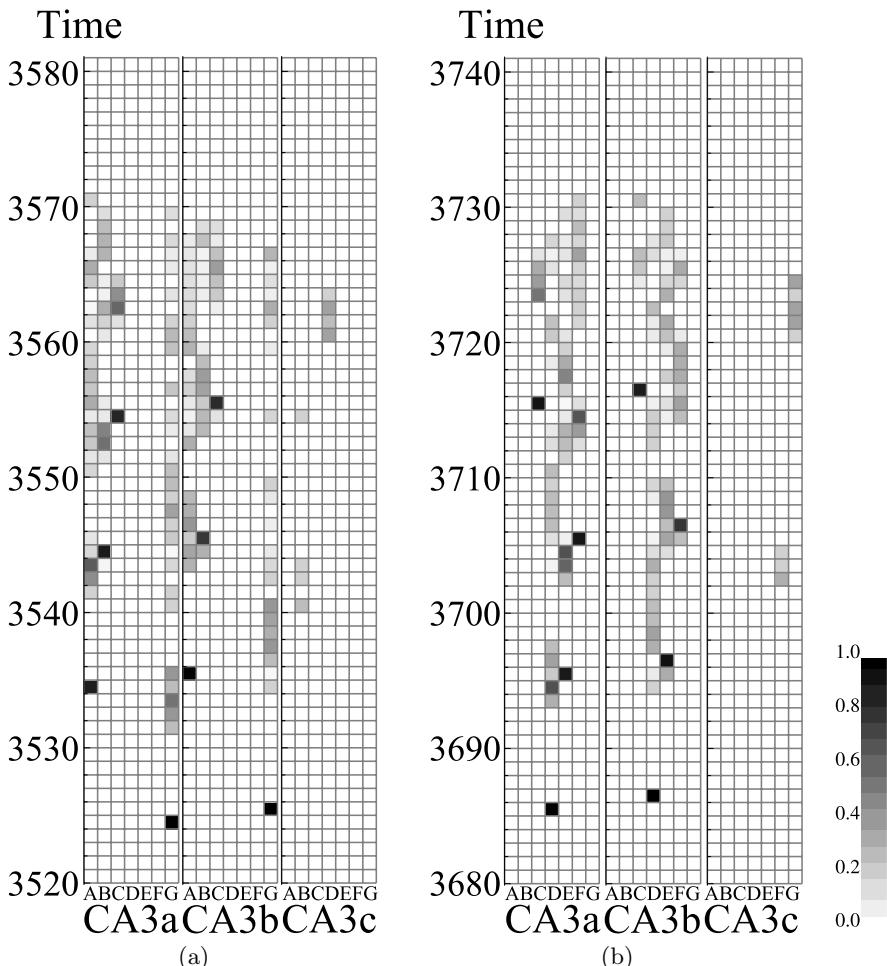


Fig. 3. Similarity in each subregion. Gray level of each cell indicates similarity between the output and each pattern. (a) Similarity in subsequence α . (b) Similarity in subsequence β .

with other CA3 subregions [4]. Moreover, a recent study suggested strong feedforward pathways crossing the hippocampal region from CA3c to DG [4]. Thus, it is clear that the context-like information generated in CA3c is transmitted to CA1 and DG through the feedforward and feedback pathways, respectively. Kesner *et al.* reviewed the functions from experimental results [15]. In their review, DG-CA3 cooperation works as spatial pattern separation that separates spatial information, so that spatial locations are separated from each other and spatial interference is reduced. In addition, Miyazawa *et al.* suggested that DG regulates the memory formation of CA3 on the basis of selective desensitization by which context-like information is used [16]. In the artificial neural networks, selective desensitization can create distributed representation that retains the

generalization capability. Thus, DG–CA3 cooperation creates a representation that retains the generalization capability and reduces spatial interference to improve hippocampal memory performance. It is highly possible that the context-like information transmitted to DG through the feedback pathway is used for the spatial memory formation. On the other hand, CA3–CA1 cooperation works as temporal pattern separation, which is important in sequential processing of information [5] and leads to sequence disambiguation. In the Wood *et al.* experiments, neurons in the CA1 region showed different activities despite the same location according to the trial type [6]. Further, by using computer simulation, Yoshida *et al.* demonstrated that sequence sensitivity would emerge in CA1, which reflects context-like information generated in CA3 [7], although our suggestion differs from their demonstration in that the context-like information is generated in CA3c. Thus, it is highly possible that the context-like information transmitted through the feedforward pathway is used for temporal information processing including sequence disambiguation.

6 Conclusions

In this study, we have demonstrated the validity of the suggestion that context-like information is generated by the cooperation between the selectivities of hippocampal CA3 even where the sequences that may occur in a modified T-maze [6] are used. Here it should be noted that CA3c generates the context-like information. CA3c has the important structural characteristic of feedforward and feedback connections. Conventionally, DG is important as the entry point of the hippocampus on the general signal pathway: DG→CA3→CA1. However, Scharfman suggested that CA3 may be as much of an entry point to the hippocampus as DG because of its wiring spreading the information to DG and CA1 [14]. In this discussion, we have emphasized that CA3c generates context-like information and spreads it to DG and CA1. Moreover, on the basis of the functional relationship between CA3c and other hippocampal regions (DG and CA1), it is highly possible that the feedforward and feedback of the context-like information play an essential role in the memory formation of the hippocampus by affecting spatial and temporal information processing in DG and CA1, respectively. Consequently, we have proposed the significance for the hippocampal memory formation of the context-like information generated in CA3c.

References

1. Eichenbaum, H.: Hippocampus: cognitive processes and neural representations that underlie declarative memory. *Neuron* 44, 109–120 (2004)
2. Levy, W.B.: A sequence predicting CA3 is a flexible associator that learns and uses context to solve hippocampal-like tasks. *Hippocampus* 6, 579–590 (1996)
3. Samura, T., Hattori, M., Ishizaki, S.: Autoassociative and heteroassociative hippocampal CA3 model based on physiological findings. *Abstracts of BrainIT2006* 62 (2006)

4. Samura, T., Hattori, M., Ishizaki, S.: Sequence disambiguation by functionally divided hippocampal CA3 model. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4232, pp. 117–126. Springer, Heidelberg (2006)
5. Samura, T., Hattori, M., Ishizaki, S.: Spatial and temporal selectivity of hippocampal CA3 and its contribution to sequence disambiguation. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4669, pp. 49–58. Springer, Heidelberg (2007)
6. Wood, E.R., Dudchenko, P.A., Robitsek, R.J., Eichenbaum, H.: Hippocampal neurons encode information about different types of memory episodes occurring in the same location. *Neuron* 27, 623–633 (2000)
7. Ishizuka, N., Weber, J., Amaral, D.G.: Organization of intrahippocampal projections originating from CA3 pyramidal cells in the rat. *J. Comp. Neurol.* 295, 580–623 (1990)
8. Tsukada, M., Aihara, T., Kobayashi, Y., Shimazaki, H.: Spatial analysis of spike-timing-dependent ltp and ltd in the ca1 area of hippocampal slices using optical imaging. *Hippocampus* 15(1), 104–109 (2005)
9. Xu, N., Ye, C., Poot, M., Zhang, X.: Coincidence detection of synaptic inputs is facilitated at the distal dendrites after long-term potentiation induction. *J. Neurosci.* 26(11), 3002–3009 (2006)
10. Izhikevich, E.M.: *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. MIT Press, Cambridge (2007)
11. Treves, A., Rolls, E.T.: Computational Constraints Suggest the Need for Two Distinct Input Systems to the Hippocampal CA3 Network. *Hippocampus* 2, 189–200 (1992)
12. Ishizuka, N., Maxwell, W., Amaral, D.G.: A quantitative analysis of the dendritic organization of pyramidal cells in the rat hippocampus. *J. Comp. Neurol.* 362, 17–45 (1995)
13. Levy, W.B., Wu, X., Baxter, R.A.: Unification of hippocampal function via computational/encoding considerations. *IJNS* 6, 71–80 (1995)
14. Scharfman, H.E.: The CA3 "backprojection" to the dentate gyrus. *Prog. Brain Res.* 163, 627–637 (2007)
15. Kesner, R.P., Lee, I., Gilbert, P.: A behavioral assessment of hippocampal function based on a subregional analysis. *Reviews in the Neurosciences* 15, 333–351 (2004)
16. Miyazawa, Y., Suemitsu, A., Morita, M.: Modeling the activity of the hippocampal neurons based on the theory of selective desensitization. *The Brain and Neural Networks* 14(1), 3–12 (2007) (in Japanese)
17. Yoshida, M., Hayashi, H.: Emergence of sequence sensitivity in a hippocampal CA3-CA1 model. *Neural Networks* 20, 653–667 (2007)

Bio-signal Integration for Humanoid Operation: Gesture and Brain Signal Recognition by HMM/SVM-Embedded BN

Yasuo Matsuyama, Fumiya Matsushima, Youichi Nishida,
Takashi Hatakeyama, Koji Sawada, and Takatoshi Kato

Department of Computer Science and Engineering,
Waseda University, Tokyo 169-8555, Japan

{yasuo,f-matsushima,y-nishida,rdsg,sd-kj0910,tkato}@wiz.cs.waseda.ac.jp
<http://www.wiz.cs.waseda.ac.jp/index-e.html>

Abstract. Joint recognition of bio-signals emanated from human(s) is discussed. The bio-signals in this paper include camera-captured gestures and brain signals of hemoglobin change $\Delta O_2 H_b$. The recognition of the integrated data is applied to the operation of a biped humanoid. Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) undertake the first stage recognition of individual signal. These subsystems are regarded as soft command issuers. Then, such low-level commands are integrated by a Bayesian Network (BN). Therefore, the total system is a novel HMM/SVM-embedded BN. Using this new recognition system, human operators can control the biped humanoid through the network by realizing more motion classes than methods of HMM-alone, SVM-alone and BN-alone.

1 Introduction

The biped humanoid, or the walking PC, has been an intriguing subject under examination from various directions including necessary functions, desirable operation methods, human interfaces and suitable areas of applications. Until a close consensus is reached, we need a pile of evidences more. Towards such a progress of the biped humanoid¹, this paper addresses the following problems.

- (1) Methods to integrate and recognize signals emanated from human(s) are presented. Recognized results are used for the humanoid operation. Data to be integrated include human gestures and brain signals. Gestures, or human motions, are captured by a camera. Brain signals are non-invadingly measured as the difference of hemoglobin levels ($\Delta O_2 H_b$).
- (2) A set of new data integration and recognition methods is given. Therein, the total system is a Bayesian Network (BN) which includes Hidden Markov Models (HMMs) and a Support Vector Machine (SVM). Therefore, the overall system is an HMM/SVM-embedded BN. This method realizes more control and motion classes than existing ones [1], [2].

¹ Hereafter, *biped* is omitted unless this emphasis is necessary.

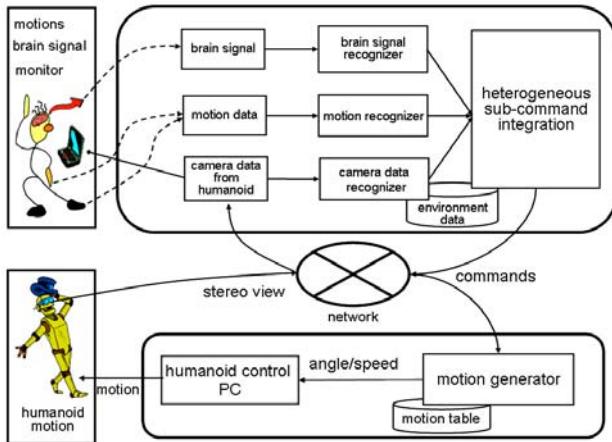


Fig. 1. Bio-signal driven humanoid control

2 Humanoid Operation by Bio-signals

The humanoid and its human operator are regarded as intelligent nodes in the network. Figure 1 illustrates the total system. The information flow is bidirectional through the network.

Human→humanoid: A human operator generates two types of signals; body motions and brain signals. Dashed lines in Figure 1 mean that these paths are actually through the network. Such signals are integrated and recognized so that appropriate commands are generated. Commands are converted to humanoid motions as angles and speeds. Then, a humanoid control PC undertakes the task of physical movements.

Humanoid→human: The humanoid has two eyes (cameras). A stereo view is sent to the monitor for the human operator through the network. Thus, the human operator and the humanoid share the same visual information although they are located distantly.

3 Major Sub-components

The total system of Figure 1 has several major sub-components. Each has enough capability to recognize incoming signals. Some methods *per se* are new by this paper.

3.1 Visual Motion Capturing and Recognition

In this paper, a video motion capturing by an off-the-shelf camera is adopted; not by dedicated motion capturing systems [1], [2]. The human operator does not wear any target sensors. On the data richness, the ordering is

optical motion capture system \approx magnetic motion capture system \gg
camera capturing with target markings > video images.

Rich data sources are desirable, however, dedicated motion capturing systems require special environments such as a metal-free space [1], [2]. On the other hand, video images are easily captured. Therefore, this paper's stance on the visual motion capturing is to make the following clear:

- (1) To establish a motion recognition system applied to the humanoid control by using video data alone.
- (2) To evaluate the possible granularity of recognizable human motions by this system.
- (3) To obtain the trade-off between the performance increase and the system sophistication. Here, the performance includes the recognition correctness and the response speed. The system sophistication mainly indicates the necessity of additional hardware. *The performance of this system is regarded as the baseline.*

The motion capture camera is fixed. This means that the background image of the human operator of Figure 1 is fixed. Operator's positions and motions are free. The flow of image preprocessing and motion recognition is as follows.

[Motion Recognition]

Step 1: The Region Of Interest (ROI) for the human shape is extracted by eliminating the background. Threshold operations are used.

Step 2: The human region is refined by using expansion and shrinking. The expansion is to set a pixel value 1 if there is a pixel of 1 in eight surrounding pixels. The shrinking is to set a pixel 0 if there is a pixel of 0 in eight surrounding pixels. This process identifies ROI.

Step 3: ROI is normalized to be $x \times y = 128 \times 256$ binary pixels. This region is regarded as a 32-dimensional vector from each mesh [4].

Step 4: HMMs are learned on this data.

Recognizable tasks and the structure of the HMM were decided by preliminary experiments.

Tasks for Recognition: For the motion recognition, three types of recognizers are used to give evidences to the Bayesian network described in Section 4.

- (1) Pararell HMMs for *standing still, right hand up, right hand down, left hand up, left hand down, right leg up, right leg down, left leg up, left leg down*. Each HMM is a left-to-right model with 8 states.
- (2) An SVM for *walking*.
- (3) A sub-BN for *squatting* by the x-y ratio.

3.2 Brain Signal Detection and Recognition

For non-invading measurements of brain states, current tools are still limited. If we address real time applications, possible measurement systems are further limited. The system which we use optically measures the level of hemoglobin ($\Delta O_2 H_b$) by probes [3]. A scene of using this machine will be found in experiments of Section 5 (Figure 6). Brain signals are generated and classified as follows.

[Brain Signal Generation and Classification]

Step 1: A human operator tries a specific task.

Step 2: Corresponding brain signal is transformed by FFT. Trends, or specific noise frequencies, are filtered out.

Step 3: An SVM is learned by using a set of teacher signals reflecting the assigned task.

So far, the granularity of the extractable information from the brain signal by the non-invading method is much less than the motion recognition of Section 3.1. But, the resolution is rapidly refined by the hardware. Besides, there is a constant necessity of the brain machine interface (BMI). The BMI integration with heterogeneous signals such as motions is novel.

3.3 Camera Data Recognizer

The camera data recognizer in Figure 1 processes images sent from two eyes of the humanoid. Since there is a constant parallax, stereo images can be obtained. Those stereo images can be utilized for the object recognition and the distance estimation to a cue. Since this recognition system is apart from the HMM/SVM-embedded BN of this paper, we refer this subject to [2].

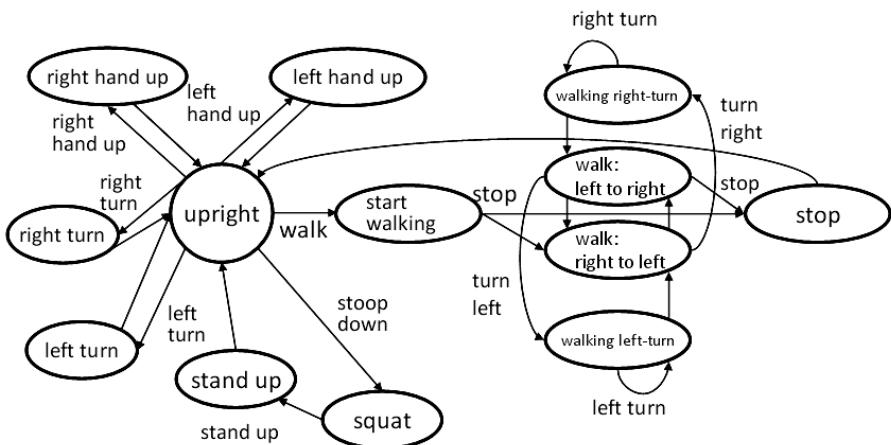


Fig. 2. Bio-signal driven humanoid control

3.4 Motion Generation at Humanoid

The target machine used in this paper is HOAP-2 [5]. It has two arms with five degrees of freedom, two legs with six degrees of freedom, and one head with one degree of freedom. There are twenty five joints. Any humanoid's flexibility does not exceed the human. Therefore, we prepared a table for individual and transition motions of the humanoid. This is to satisfy the basic requirement for the biped humanoid: Falling down by losing the balance is never allowed. Thus, the two classes of short motions were designed.

- (a) Element movements corresponding the operator's motions are prepared as a motion table illustrated in Figure 1.
- (b) Transition states are prepared between two consecutive motions which may cause falling down. Figure 2 illustrates the transition diagram for this. As can be found, there are artificially inserted states, e.g., {Stop, Walking Left Turn, Walking Right Turn}.

3.5 Network

The network illustrated in Figure 1 is not a mere transmission line. It contributes to stable humanoid motions. The command integrator issues a set of decisions based on the recognition. This result is written on a blackboard. The humanoid reads this event to understand what the command is. The command is read out from the blackboard. Upon receiving the completion signal, the next motion is generated [6].

4 Individual Recognition Methods and Integration

4.1 HMM Array for Motion Recognition

Source data for the HMM array are not sensor positions such as [1], [7]. Instead, image data after preprocessing are fed to the HMM array in this paper. The structure of the HMM array, a set of left-to-right models placed in parallel, is the same. The procedure of the image preprocessing is as follows.

- (a) Background portions are removed.
- (b) ROI is extracted by removing the background image. An extended ROI where the operator can change positions is specified.
- (c) ROI is divided into subregions.

Figure 3 illustrates an example obtained by steps (a) ~ (c). The internal square is the ROI. Note that the x-y ratio of the ROI helps to identify “squatting.”

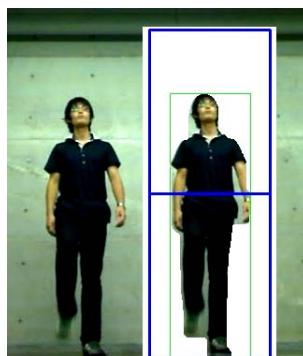


Fig. 3. Region of interest and its subregions

4.2 SVM for Brain $\Delta O_2 H_b$

The brain signal used for the humanoid operation is the oxygenated hemoglobin change $\Delta O_2 H_b$ $\mu\text{mol/L}$ [6]. This quantity is obtained by the spectroscopy. A sequence of digital outputs are given by RS232C to the network with the interval of 1/6 sec. Figure 4 (left) illustrates separation levels of $\Delta O_2 H_b$ from the left brain. Task 0 stands for the quiescent state. Task 1 is to draw a simple figure. In task 2, a subject draws a complex figure. Figure 4 (right) is the response of the left brain when a subject controls his/her middle of the forehead by knitting brows. Thus, considerable amounts of level differences appear on $\Delta O_2 H_b$.

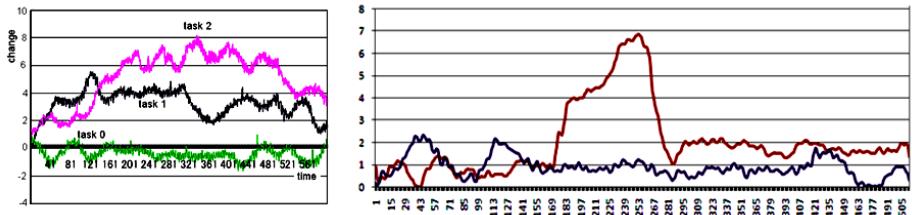


Fig. 4. Brain signal patterns

4.3 Integration by BN

An integrated recognizer is necessary since heterogeneous data are input to operate the humanoid. The Bayesian network fulfills the role of this integration. It is known that every HMM can be represented by a BN [8]. But, compact structures and learning methods of HMM help to avoid an overly huge BN.

Figure 5 is the integration BN which accepts evidences from HMM and SVM. Nodes correspond to the classification of human motions and brain signals. This structure was decided as follows.

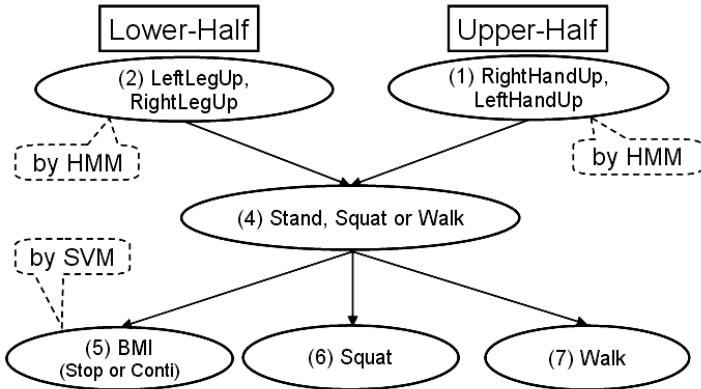
- (1) The number of possible nodes is decided by considering that of recognizable humanoid motions. This number is subject to repeated adjustments.
- (2) Several possible DAGs (directed acyclic graphs) are chosen and tested.
- (3) CPT (conditional probability table) is prepared for each DAG.
- (4) The best structure among the tested ones is chosen.

Here, three checkpoints were jointly considered.

CPT: For the CPT estimation, a beta distribution is considered.

$$P(X_i|\pi(X_i)) \approx \theta(1-\theta)^{N-n}/B(n+1, n-N+1)$$

Link Generation: For the link generation, the greedy search is used. That is, nodes are separated at the start. Then, each node's parents are decided by choosing the heaviest link.

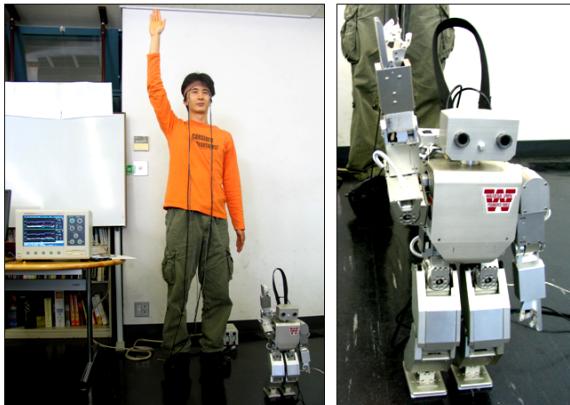
**Fig. 5.** Integration by BN

Model Selection Penalty: The BN structure is learned by posing a penalty for overly complex structure. We used the MDL (minimum description length) [9] on numbers of parent and child states.

Cross Validation: As the final step to adopt the learned model, the cross validation is tried. This process gives the final tuning.

5 Experiments

The overall experiments contain (i) the motion recognition by a 2.00GHz PC, (ii) the brain signal recognition by a 2.66GHz PC, (iii) the network communication, (iv) the humanoid motion control by a 2.40GHz PC, and (v) the visual feedback

**Fig. 6.** Human operator, brain signal processor and a humanoid

from the humanoid. The total path can be identified from Figure 1. Items (i) and (ii) are performed by the HMM/SVM-embedded BN presented by this paper. The recognition frame is twice per second (0.5 sec/recognition-refresh).

Figure 6 illustrates a scene of experiments. In the center of the left figure, a human operator raises up his right hand. This motion is reproduced by the humanoid. The right figure is an expanded illustration of the humanoid. It is important to note here that this is the recognized result of his gesture. Therefore, shoulder angles are different between the operator and the humanoid. The height of the knee is also different during the walk so that the humanoid does not fall down. On the table in the left figure, the brain signal detector is placed. There, cables from the operator's forehead (under the hair band) are connected. Brain signals of Figure 4 (right) were used to stop the walking this time. Thus, all the motions specified in section 3.1 were realized.

6 Conclusion

A system to operate biped humanoids by bio-signals was presented. The bio-signals in this paper include human motions and brain signals. Humanoid operation commands are generated by the recognition engine of the HMM/SVM-embedded Bayesian network. In this paper, dedicated motion capturing systems were not used unlike [1], [7]. Instead, an off-the-shelf camera was used for the motion capturing. On the motion recognition, therefore, we consider this paper's method to be the baseline.

Brain signals from humans need to be measured in a non-invading way so far. Therefore, the response time is still slower than human motions. However, there are provisions that the measurement system becomes smaller and the response time becomes faster. By considering this trend as well as from the theoretical interest, we designed the HMM/SVM-embedded Bayesian network for the integration of all signals. This system was successfully applied to the humanoid operation.

References

1. Matsuyama, Y., Yoshinaga, S., Okuda, H., et al.: Towards the unification of human movement, animation and humanoid in the network. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 1135–1141. Springer, Heidelberg (2004)
2. Kato, J., Takahashi, N., Ueda, Y., Sugihara, Y., Matsuyama, Y.: Networked remote operation of humanoid via motion interpretation and image recognition. In: Proc. The Third International Conference on Autonomous Robots and Agents, pp. 51–56 (2006)
3. Hamamatsu Photonics, K.K.: NIRO 200 document (2003)
4. Umeda, M.: Recognition of multi-font printed Chinese characters. In: Proc. 6th ICPR, pp. 793–796 (1982)
5. Fujitsu Automation Co.: HOAP-2 Reference Manual (2003)

6. Matsuyama, Y., Shiga, T., et al.: Network communication strategies for cooperative physical agents. In: Proc. Asia-Pacific Symposium on Information and Telecomm. Tech., pp. 148–153 (2005)
7. Matsuyama, Y., Nishida, Y.: HMM-embedded Bayesian network for heterogeneous command integration: Applications to biped humanoid operation over the network. In: Proc. Int. Conf. on Soft Computing as Transdisciplinary Sci. and Tech., pp. 138–145. ACM Press, New York (2008) The Best Paper Award
8. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall, Englewood Cliffs (2003)
9. Rissanen, J.: Modeling by shortest data description. Automatica 14, 465–471 (1978)

Interpreting Dopamine Activities in Stochastic Reward Tasks

Akiyo Asahina^{1,2}, Jun-ichiro Hirayama², and Shin Ishii^{1,2}

¹ Graduate School of Information Science, Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara 630-0192 Japan

² Graduate School of Informatics, Kyoto University,
Gokasho, Uji, Kyoto 611-0011, Japan

akiyo-a@is.naist.jp, junich-h@sys.i.kyoto-u.ac.jp, ishii@i.kyoto-u.ac.jp

Abstract. Phasic activities of dopamine (DA) neurons in the primate midbrain have been considered as representing temporal difference (TD) errors from a computational perspective. Recently, several studies have reported that, in stochastic reward tasks, the DA activities gradually increase before receiving actual rewards, which is not well explained by the simple TD model. In this study, we propose an alternative model based on a probabilistic formulation of the stochastic reward task. In simulation experiments, expectation errors, defined by the probabilistic modeling, well described the gradually increasing DA activities during a wait period even in a single trial.

1 Introduction

In order for animals to behave effectively in their surrounding environments, which could be uncertain, it is essential to predict future outcomes as accurately as possible. From neurophysiological observations that the activities of dopaminergic (DA) neurons in the primate midbrain resemble the temporal difference (TD) errors, it has been hypothesized that DA neurons are involved in reinforcement learning (RL), which is an autonomous learning scheme to allow the animals to adapt to their environments [1]. Recently, however, several works have discussed that the DA responses when performing stochastic reward tasks, in which the reward is provided in a probabilistic manner, cannot be explained by the simple TD model. It has been reported that, in a stochastic reward task, the DA activities gradually increase between the onset of a conditioned stimulus (CS) and the reward delivery time [2]. Niv et al. [4] proposed a computational model in which such a gradual increase (GI) phenomenon can be seen as an artifact when the TD errors are back-propagated along the time and averaged over trials. On the other hand, Fiorillo et al. [5] suggested that the GI occurs even in a single trial and the averaging TD errors cannot explain the DA behavior by itself. Accordingly, the GI activities of DA neurons are under controversy, though they are related to the uncertainty of the animal's situation, intuitively. In this study, we formulate the stochastic reward tasks by using a simple probabilistic model and show by simulation experiments that the difference in the

reward prediction can exhibit GI-like behaviors, due mainly to the psychological uncertainty of the reward delivery time.

2 DA Activities in Stochastic Reward Tasks

In neurophysiological studies in which macaque monkeys were performing delayed reward tasks, Schultz et al. [1] showed that the DA activities in the monkey midbrain (areas A8 and A10) look like the TD errors; the DA neurons responded not to rewards but to reward-associated stimuli. When the associated reward was not given, the DA firing rate became smaller than the baseline rate around the time when the reward would have been given. These results were obtained when the monkeys were performing deterministic tasks, in which the rewards were always given, while Fiorillo et al. and Tobler et al. [2,3] have recently reported DA activities when performing stochastic reward tasks. In the task by Fiorillo et al., two monkeys were conditioned in a Pavlovian procedure with distinct five visual stimuli indicating the probability ($p = 0, 0.25, 0.5, 0.75, 1.0$) of a liquid reward that is delivered after a delay of 2 s. In the task by Tobler et al., the reward probability was fixed at $p = 0.5$, but three distinct visual stimuli were associated with three different volumes of a liquid reward: $r = 0.05, 0.15, 0.50$ ml. In the former stochastic reward task, the uncertainty of the reward delivery (or the entropy of the forthcoming probabilistic event) is maximized at $p = 0.5$ and minimized at the two extremes ($p = 0$ and 1); such an uncertainty is critical in assessing the actual outcome against the prediction induced by the visual cue. The GI of DA neurons may occur because the DA signal encodes the uncertainty of the reward prediction as well as errors in the reward prediction [2,3]. A later computational study [4] suggested that the GI activities are artifacts when averaging the back-propagated TD errors over trials, assuming the single DA firing has an asymmetric nature. DA cells are known to have the low baseline firing rate; positive prediction errors are represented by firing rates of about 270 % of the baseline, whereas negative errors induce a decrease of only about 55 % below the baseline [4].

Contrary to this computational consideration, Fiorillo et al. [5] argued that there is a strong and sustained activation between the onset of CS and the reward delivery even in individual trials so that the DA signals are highly probably associated with the uncertainty about the probabilistic reward outcome.

3 The Proposed Model

The GI activities of DA neurons should represent some sort of uncertainty, but it cannot be directly included in the simplest TD model which is based on the scheme of stochastic approximation. We present here a hypothesis that the dopaminergic system performs the value estimation under the consideration of the uncertainty of the reward delivery, in a probabilistic inference manner, and the DA activities would correspond to the temporal difference in the value estimation; this is an extension of the simple TD theory, but does not stay in the

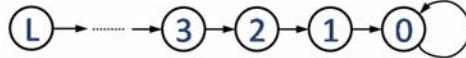


Fig. 1. State transition in the stochastic reward task

conventional TD scheme because the system should identify the environment with consideration of its stochastic nature. With the formulation of value estimation in the stochastic reward task, we will demonstrate that the GI of DA neurons is attributed to the uncertainty in the estimation of the reward time.

First, we present a probabilistic formulation of the stochastic reward task. $t \in \{1, 2, \dots, t_{\max}\}$ indexes discretized timings in an episode, where $t = 1$ and $t = t_{\max}$ are the onset of the visual stimulus and the end of one episode, respectively. A reward is given as r_t at time t , but in the monkeys experiment it may be provided only at a fixed reward time $t = t_{\text{reward}}$.

It is assumed that macaque monkeys estimate the following expected return based on the reward sequences \mathcal{H} experienced in the past stochastic reward tasks:

$$v_t = \mathbb{E} \left[\sum_{u=0}^{t_{\max}-t} \gamma^u r_{t+u} \mid \mathcal{H} \right], \quad (1)$$

where $\gamma \in [0, 1]$ is the discount rate.

The stochastic reward task is modeled as the following dynamical system model, in which the reward time t_{reward} is probabilistic, reflecting that the monkeys would recognize the reward time as fluctuating in their cognition even if it is actually fixed. It should be noted that this dynamical system model is in the cognitive level of the subject monkeys. To model the (cognitive) state transition within an episode, we define an absorbing Markov chain as in Fig. II, in which the state variable takes $z_t \in \{0, 1, 2, \dots, L\}$; a reward is provided probabilistically at the state $z_t = 1$, and $z_t = 0$ represents the absorbing state in which transition to other states does not occur. The state changes from $z_{t-1} = l$ ($1 \leq l \leq L$) to $z_t = l-1$ with probability 1. The initial state is generated according to an initial distribution $\phi(l)$ ($\phi(0) = 0$). Then, the state $z_t \in \{1, 2, \dots, L\}$ corresponds to the remaining steps to a reward event and $z_t = 0$ to the state after the reward event (even though the reward is not delivered). Although transitions between these states occur deterministically, the initial state arises probabilistically; then the reward time t_{reward} is also distributed probabilistically. The observation process, i.e., the conditional distribution for the observation r_t at state z_t , is defined by

$$p(r_t | z_t) = \begin{cases} \mathcal{N}(r_t | \mu, \sigma^2) & (z_t = 1) \\ \mathcal{N}(r_t | 0, \epsilon^2) & (z_t \neq 1), \end{cases} \quad (2)$$

where the model parameters, μ and σ^2 , represent the mean and the variance of the reward amount, respectively, and ϵ^2 is a small constant. The second Gaussian is an approximation of the Dirac delta function with the center of zero.

We here assume that these model parameters have already been obtained. Then, (II) is calculated as:

$$v_t = \sum_{z_t} p(z_t | \mathbf{r}_{1:t}) V(z_t). \quad (3)$$

where $V(\cdot)$ is the state-value function, defined as

$$V(z_t) = \sum_{\mathbf{z}_t^+} p(\mathbf{z}_t^+ | z_t) \sum_{u=0}^{t_{\max}-t} \gamma^u m(z_{t+u}), \quad (4)$$

In the equation above, we have used the following notations: $\mathbf{z}_t^+ = (z_{t+1}, \dots, z_{t_{\max}})$ and $m(z_t) = \mu$ ($z_t = 1$), $m(z_t) = 0$ ($z_t \neq 1$).

Furthermore, since the state transition is deterministic, the value function can be rewritten simply as

$$V(l) = \begin{cases} 0 & (z_t = 0) \\ \gamma^{l-1} \mu & (z_t \neq 0). \end{cases} \quad (5)$$

The model thus defines the estimation way of the expected return v_t at each timing t . After this estimation, we can calculate the difference in the return expectation between two consecutive timings, $\delta_t = v_t - v_{t-1}$; this newly defined quantity (temporal difference) has a similar meaning to the TD error in the RL.

In this study, we assume that the DA signals in the primate midbrain represent this temporal difference in the return expectation and present simulations illustrating its behaviors under several conditions.

Our newly-defined temporal difference δ_t shows GI-like behaviors, as can be seen in a later simulation experiment, when the discount rate γ is set smaller than one. When γ equals one, i.e., there is no discount, $V(l)$ is consistently μ for any $z_t \neq 0$. Since the (deterministic) state transition in our model progresses only toward the absorbing state ($z = 0$), the posterior probability $p(z_t \neq 0 | \mathbf{r}_{1:t})$ is non-increasing as the time goes, implying that the expectation $v_t = p(z_t \neq 0 | \mathbf{r}_{1:t}) \cdot \mu + p(z_t = 0 | \mathbf{r}_{1:t}) \cdot 0 = p(z_t \neq 0 | \mathbf{r}_{1:t})\mu$ is also non-increasing. The temporal difference, δ_t , thus never becomes positive in this case except for $t = 1$ (where the previous value v_0 is normally set at zero reflecting the lack of subjective knowledge about the onset of each episode) and no GI-like activity occurs. In contrast, if γ takes an appropriately smaller value than one, the state value, $V(l) = \gamma^{l-1} \mu$, exponentially increases from $l = L$ to $l = 1$. Then, the expectation v_t of the state value also comes to increase in a super-linearly fashion so that the difference δ_t exhibits GI-like activities.

On the other hand, setting γ appropriately is important for a RL agent to behave effectively in stochastic and even non-stationary environments, though our model does not consider the way of action selection because the monkeys task is essentially in a classical conditioning procedure. In many simple applications of RL, γ has been fixed at a hand-tuned value appropriately for the current environment, but biological agents like animals should adjust it in each specific

situation with which the agents are facing. In this study, we introduce the “meta-learning” scheme proposed by Schweighofer and Doya [6] in the context of RL, to our stochastic reward task (which is not essentially RL).

At the n -th episode in the repetition of the stochastic reward task, γ_n is generated probabilistically by

$$\gamma_n = 1 - \exp(-y_n). \quad (6)$$

y_n is updated as

$$y_{n+1} = y_n + \eta (\bar{r}_n - \bar{\bar{r}}_n), \quad (7)$$

where η is the learning rate, \bar{r}_n and $\bar{\bar{r}}_n$ are short-term and long-term running averages of the reward, respectively. Then, given the immediate reward value r_n , they are also updated at each step by

$$\bar{r}_{n+1} = \bar{r}_n + \frac{1}{\tau_1} (r_n - \bar{r}_n), \quad (8a)$$

$$\bar{\bar{r}}_{n+1} = \bar{\bar{r}}_n + \frac{1}{\tau_2} (\bar{r}_n - \bar{\bar{r}}_n). \quad (8b)$$

The previous study [6] empirically confirmed that these update rules properly adjust γ_n according to the balance between the short-term and long-term reward averages.

4 Computer Experiments

According to the neurophysiological studies, it is known that (i) the DA activities at the onset of visual stimulus are proportional to the reward amount and to the reward probability, which can be explained by the simple TD model; and (ii) the GI activities occur between the onset of the visual stimulus and the reward delivery time. They are (ii-a) strongest when the reward is highly probabilistic ($p = 0.5$) when changing the reward probability, and (ii-b) proportional to the reward amount when varying the reward amount. The characters (i) and (ii-b)

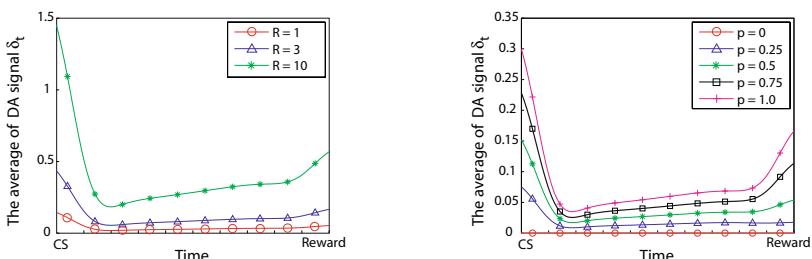


Fig. 2. DA activities including GI (averaged over episodes). The parameters were $\gamma = 0.9$, $L = 20$ and $t_{\max} = 50$. Left: when the reward amount was varied, $R \in \{1, 3, 10\}$, but the probability was fixed at $p = 0.5$. Right: when the reward probability was varied, $p \in \{0, 0.25, 0.5, 0.75, 1.0\}$ with a fixed reward amount.

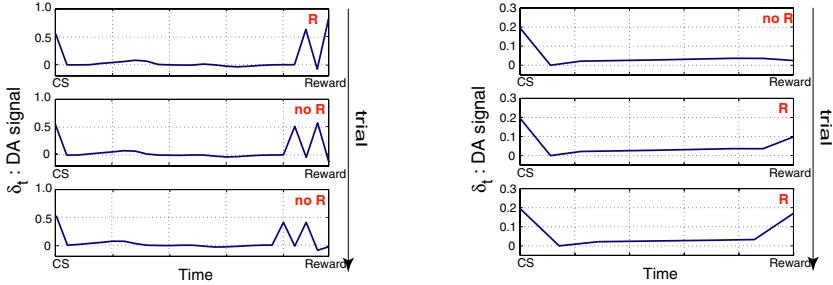


Fig. 3. DA activities in individual episodes. The parameters were $p = 0.5$, $\gamma = 0.9$ and $R = 1$. Left: the DA model by Niv et al. Right: our model.

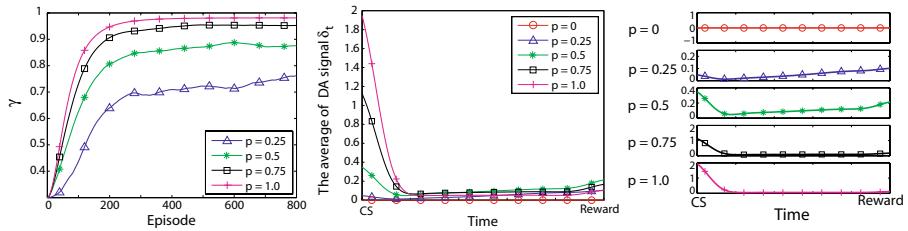


Fig. 4. The meta-parameter algorithm [6] was applied to automatically adjust the meta-parameter γ . Left: Meta-learning process of γ . Middle: GI activities (averaged over episodes) after the discount factor γ was adjusted. Right: GI activities with different reward probability is varied in each panel. The parameters were $\tau_1 = 100$, $\tau_2 = 100$, $\eta = 0.036$, $\bar{r}_1 = 0$, $\bar{r}_2 = 0$, $\gamma_1 = 0.3$, $r_n \in \{0, R\}$, $R = 3$. Note the differences in scales.

above were well reproduced by the temporal difference δ_t in our model, when the reward amount was varied (Fig. 2, left panel) and when the reward probability was varied (Fig. 2, right panel). The character (ii-a), however, could not be well explained because the discount rate γ was constant regardless of the mean value μ . Although the reward delivery time is fixed in the physiological experiments, we set it, in these simulation experiments, in a probabilistic manner, in order to simulate the cognitive fluctuation of the reward time. More concretely, we produced sequences of the true state $z_1, z_2, \dots, z_{t_{\max}}$ according to the dynamical system model described in the previous section, and then, the reward observation is given by $r_t = 0$ when $z_t \neq 1$, and by either of $r_t = R$ with probability p or $r_t = 0$ with probability $1 - p$ when $z_t = 1$. This observation process corresponds to the real situation. We repeated the stochastic reward tasks, and the reward probability p and the reward amount R are both fixed during the repetition.

Since our model is based on the probabilistic estimation of the environment assuming the dynamical system model, the uncertainty of the reward time is involved into the estimation of the value function. Then, the temporal difference δ_t becomes the averaged one over variety of possible reward sequences. Therefore, even in an individual episode, the difference can represent the uncertainty like

the GI activity (Fig. 3, right panel), which cannot be reproduced by a simple TD model (Fig. 3, left panel). If the GI activity represents the uncertainty, it should be the consequence of some kind of probabilistic estimation, because the uncertainty reflects the possible variety which does not directly correspond to the situation in the currently performing task. When the reward probability is varied, the meta-learning of the discount factor γ [6] worked well to adaptively reproduce the GI-like activities. According to our implementation of the meta-learning, the discount factor γ was determined after obtaining the average reward over past episodes instead of over past trials, because in our stochastic reward task the reward is provided probabilistically only once at $t = t_{\text{reward}}$. In this implementation, the parameters, τ_1, τ_2 , were set at the same as in [6], whereas the learning rate η and the initial value of y_1 were set adequately.

The left panel in Fig. 4 shows meta-learning processes of the discount factor γ when the reward probability is varied. It should be noted that this meta-learning algorithm is not effective when the reward probability is zero (then the expected return is necessarily zero). As to reflect the probabilistic nature of reward delivery events, the meta-parameter γ was automatically adjusted; e.g., it was set at $\gamma = 1$ when the reward was certainly provided ($p = 1$). After the discount factor obtained by this meta-learning procedure was plugged in, we examined the behaviors of the temporal difference we defined. As the result, the temporal difference well reproduced the phasic activity at the onset of the visual cue and the GI activities during the wait period (Fig. 4, middle panel); the latter GI activities were largest when $p = 0.5$, whose character was not reproduced without the meta-learning (Fig. 2, right panel). More specifically, the gradient of the GI activities along the time steps was maximal when $p = 0.5$, which was also seen (Fig. 4, right panel).

5 Concluding Remarks

In this study, we presented a stochastic dynamical system model which formulates the stochastic reward task. The temporal difference δ_t in the estimated value function well demonstrated the GI activities which were observed in the midbrain DA neurons but their meaning has still been unknown. According to our hypothesis, the DA neurons monitor the changes of the situation goodness in terms of temporal difference in the value estimation. In a simple deterministic situation, the DA system would easily predict the future outcome. In a more realistic situation, however, to appropriately evaluate the current situation needs to compare it with possible sequences within the variety. Such a comparison should require probabilistic estimation with the identification of the variety; this is nothing but the evaluation of uncertainty of the environment. In our probabilistic model, the estimation of the value function is performed after the identification of the probabilistic nature of the environment. Then, the temporal difference could include the uncertainty of possible reward sequences, leading to GI activities during the wait period. Although in this study we did not directly deal with parameter estimation of the environment, the introduced meta-learning of

the discount factor γ actually worked as the identification of the probabilistic nature of environments; the automatic tuning of the meta-parameter γ was effective in well reproducing the GI activities when the reward probability was varied. In a probabilistic environment, the temporal difference in a single task sequence should be normalized by the average reward over repetition of the same tasks; for example, even a small amount of reward is very favorable if the reward probability is small. Such a normalization based on comparison between the short-term and long-term averages of reward was implemented as the meta-learning of γ in this study. Doya [8] formerly suggested the relationship between the meta-parameter γ in RL and the global regulation by neuromodulator serotonin. The normalization of the temporal difference by the discount factor γ might correspond to the crosstalk between the dopamine and serotonin within the low-dimensional decision making system in the brain.

Based on an extension of our formulation of the stochastic reward task, we will be able to examine DA activities when the reward is delivered earlier or later than usual timing after learning, which was discussed in a former computational study [7]. Although we have assumed the DA neurons represent the temporal difference δ_t in this study, we did not clarify how it is used in the learning or decision making. One intuitive idea is that the temporal difference with the consideration of the environmental uncertainty is useful when animals are performing actions under the balance between the exploration and exploitation. Since the monkeys task is in a classical conditioning framework which does not require any action selection model, we will, in our future work, extend the present formulation as to incorporate an action selection model and make the role of our temporal difference clear.

Animals confronting the uncertain world should predict future outcomes under the consideration of the probabilistic nature. DA activities should be involved in decision making and adaptation to such an uncertain world.

References

1. Schultz, W., Dayan, P., Montague, P.R.: A neural substrate of prediction and reward. *Science* 275, 1593–1599 (1997)
2. Fiorillo, C.D., Tobler, P.N., Schultz, W.: Discrete coding of reward probability and uncertainty by dopamine neurons. *Science* 299, 1898–1902 (2003)
3. Tobler, P.N., Fiorillo, C.D., Schultz, W.: Adaptive coding of reward value by dopamine neurons. *Science* 307, 1642–1645 (2005)
4. Niv, Y., Duff, M.O., Dayan, P.: Dopamine, uncertainty and TD learning. *Behav. Brain Funct.* 1, 6 (2005)
5. Fiorillo, C.D., Tobler, P.N., Schultz, W.: Evidence that the delay-period activity of dopamine neurons corresponds to reward uncertainty rather than backpropagating TD errors. *Behav. Brain Funct.* 1, 7 (2005)
6. Schweighofer, N., Doya, K.: Meta-learning in Reinforcement Learning. *Neural Netw.* 16, 5–9 (2003)
7. Daw, N.D., Courville, A.C., Touretzky, D.S.: Representation and Timing in Theories of the Dopamine System. *Neural Comput.* 18, 1637–1677 (2006)
8. Doya, K.: Metalearning and neuromodulation. *Neural Netw.* 15, 495–506 (2002)

Epileptogenic ECoG Monitoring and Brain Stimulation Using a Multifunctional Microprobe for Minimally Invasive Brain Microsurgery

Toshitaka Yamakawa^{1,*}, Takeshi Yamakawa²,
Michiyasu Suzuki³, and Masami Fujii⁴

¹ Department of Electrical and Electronics Engineering,

Faculty of Engineering, Shizuoka University,

3-5-1 Johoku, Naka-ku, Hamamatsu-shi, 432-8561, Japan

ttyamak@ipc.shizuoka.ac.jp

² Department of Brain Science and Engineering,
Graduate School of Life Science and Systems Engineering,

Kyushu Institute of Technology,

2-4 Hibikino, Wakamatsu, Kitakyushu, Fukuoka 808-0196, Japan

³ Department of Neurosurgery, Graduate School of Medicine,

Yamaguchi University, 1-1-1 Minami-Kogushi, Ube, Yamaguchi 755-8505, Japan

⁴ Department of Neurosurgery, Yamaguchi University Hospital,
1-1-1 Minami-Kogushi, Ube, Yamaguchi 755-8505, Japan

Abstract. A microprobing system which has the functions of measuring the intracranial EEG(IC-EEG)/temperature, providing the brain stimulation current, and freezing brain tissue is proposed for the minimally invasive brain microsurgery of intractable epilepsy treatment.

1 Introduction

About 50 million people worldwide have epilepsy at any one time. The lifetime prevalence of epilepsy (i.e. the number of people presently in the world who have epilepsy now or have had it in the past or will experience it in the future) is approximately 100 million people. The mean prevalence of active epilepsy which causes continuing seizures and needs treatments is 0.82% of the general population around the world[1]. Seizures in 70% of the epilepsy patients can be controlled with medications, even though it can not be cured. However, up to 30% of the patients do not respond to the medications even with the best (strongest) available medicines. In that case, surgeries are applied to remove the brain tissue (:cortex, usually) of the epileptogenic focus. The removing area of the brain tissue is determined by the detected epileptogenic focus. However, the detection accuracy is up to several centimeters. Thus the risk of the side and after effects cannot be avoided.

* This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Specially Promoted Research, No.20001008, 2008.

In this paper, a multifunctional microprobe, intracranial EEG(IC-EEG)/temperature sensing interfaces, and a current source for brain stimulation are proposed to be compatible with the minimally invasive brain microsurgery. The proposed microprobe system has the functions of measuring the IC-EEG/temperature, providing brain stimulation current, and freezing the brain tissue around the microprobe's tip for few millimeters.

2 System Configuration

Figure 1 shows the block diagram of the proposed minimally invasive multifunctional microprobe system. This system mainly consists of 4 blocks: the multifunctional microprobe, the EEG instrumentation amplifier, the brain stimulation current source, and the thermocouple amplifier.

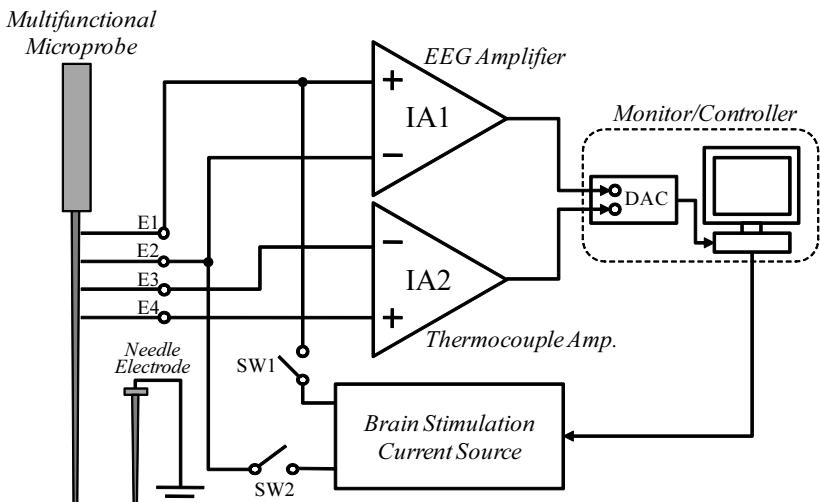


Fig. 1. Proposed minimally invasive multifunctional microprobe system

The system operation is briefly explained in below.

- 1) The multifunctional microprobe is inserted to the pre-detected epileptogenic focus which is the break-out area of the epileptic burst.
- 2) In order to confirm that the probe is inserted to the correct location (i.e. the epileptogenic focus), it is stimulated by the current (enough for activating epileptogenic focus and less than activating normal neurons) through the multifunctional microprobe with SW1 and SW2 closed, and after that the voltage response of IC-EEG is measured by the EEG Amplifier with SW1 and SW2 opened. If the epileptic burst is observed, the probe inserted position is determined to be an epileptogenic focus.

- 3) The refrigerant gas flows into the probe through the inner pipe, then the temperature of the probe's tip falls. Hence the brain tissue around the tip is frozen.
 4) The brain stimulation current is applied in order to confirm that the brain tissue of the epileptogenic focus is adequately neutralized. When the epileptic burst does not occur after the stimulation, the epileptogenic focus no longer exists.

The details of the each block are explained in the following subsections.

2.1 Multifunctional Microprobe

The multifunctional microprobe and its cross sectional view are shown in Fig. 2.

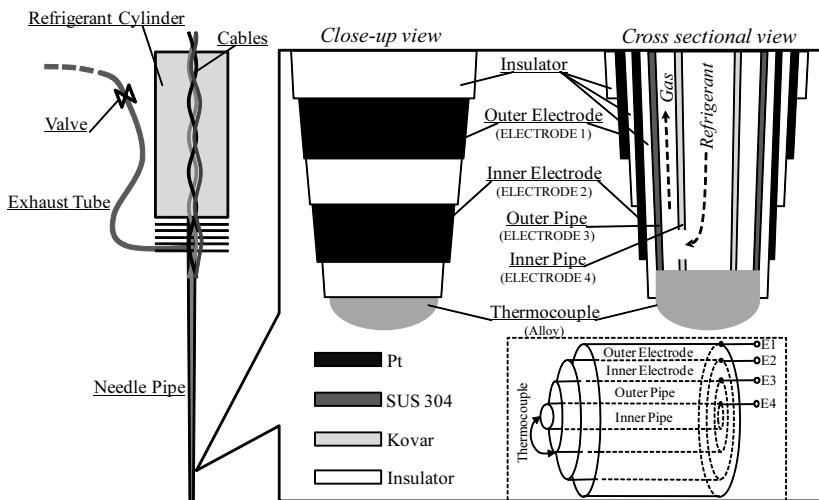


Fig. 2. The proposed multifunctional microprobe and its cross sectional view

The needle pipe consists of electrodes and coaxial pipes. The electrodes are made of iridium[3] or platinum and acts as stimulating/measuring electrodes (:ELECTRODE 1, ELECTRODE 2). The outer pipe is made of stainless steel SUS304 and acts as a negative thermocouple electrode (:ELECTRODE 3). The inner pipe is made of Kovar (Fe54%-Ni29%-Co17%) and acts as a positive thermocouple electrode (:ELECTRODE 4) and a refrigerant guide to evaporate it. Thus the tip of the needle pipe acts as rapid cooling/freezing probe and also acts as temperature measuring probe. The diameters of the tip and the root of the needle pipe are 200 μ m and 1mm, respectively.

The refrigerant cylinder is filled with the refrigerant gas, and is connected to the inner pipe of the needle pipe. Here, HFC (hidrofluorocarbon) is used as the refrigerant gas, since it has relatively high boiling point (-80~ -20°C). During the valve is opened, the refrigerant gas flows through the hole made on the outer

pipe as shown in the cross sectional view of Fig. 2. The freezing ability of the microprobe has already been confirmed in the experiments with saline solidified by gelatin [2].

2.2 Instrumentation Amplifiers

Two instrumentation amplifiers (IA1 and IA2) are used in the proposed system; the EEG amplifier and the thermocouple amplifier. The same circuit design is adopted for both instrumentation amplifiers except the value of the gain-controlling resistor R_G . Figure 3 shows the circuit diagram of the instrumentation amplifier.

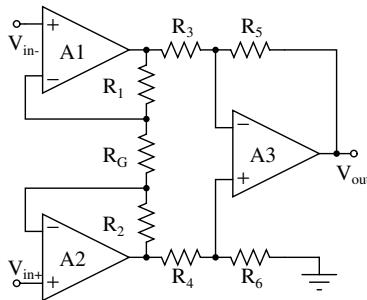


Fig. 3. The circuit diagram of the instrumentation amplifier

The instrumentation amplifier adopts the 3-Op-amp architecture which is commonly used for the high CMR (Common Mode Rejection) amplifiers [4]. In order to obtain high CMR, the resistor values need to strictly be matched as follows: $R_1 = R_2$, $R_3 = R_4$, $R_5 = R_6$. In addition, the carefully matched layout design is required for the Op amps and the switches to avoid CMR degradation caused by the device mismatch. The output voltage V_{out} in Fig. 3 can be given by the following equation (1) when the above conditions are satisfied.

$$V_{out} = (V_{in+} - V_{in-}) \left(1 + \frac{2R_1}{R_G} \right) \frac{R_5}{R_3} \quad (1)$$

In the ECoG measurement using a subdural electrodes array (electrodes distance: 10mm, electrode size: 4mm ϕ), 1~1.5mV of the epileptic burst can be measured. On the other hand, 100~300 μ V of the epileptic burst can be measured in the depth EEG measurement using depth electrodes (the electrodes are equipped on a 1mm-diameter catheter, electrodes distance: 5mm~10mm) [5]. Hence, the differential voltage between the electrodes in Fig. 2 can be estimated to be less than few hundreds of microvolts. Therefore, the gain is set to 60~80dB in the proposed EEG amplifier. The OFF resistance R_{OFF} of the switches must be sufficiently larger than the equivalent output impedance of the EEG source (\geq hundreds of kilo ohms), in order to avoid the loading effect of the brain stimulation current source.

The thermocouple amplifier is designed to have a gain of 80dB, since the Seebeck coefficient of the thermocouple which appears at the probe's tip was $470\mu\text{V/K}$.

For the proposed instrumentation amplifiers, a bipolar supply voltage is necessary. The element values used in the circuits are shown in Sect. 3.

2.3 Brain Stimulation Current Source

A variable current source is used as the brain stimulation current source. The current source is required to provide the square current pulses into the brain tissue which is less than 800Ω . The required specifications for the current source (defined in 5) is shown in Table II.

Table 1. The required specification for the current source

Pulse height	0.5~2mA
Pulse width	0.5~2ms
Frequency	40~60Hz

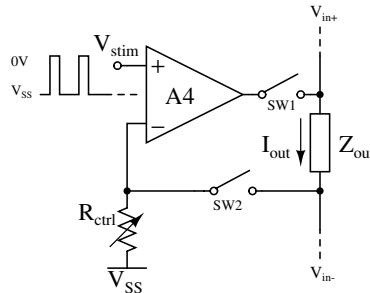


Fig. 4. The brain stimulation current source

Figure 4 shows the circuit diagram of the current source used for the brain stimulation. The circuit consists of a simple current-output Op amp with the feedback through the load. Analog switches SW1, and SW2 are inserted as shown in Figs. II and 4 in order to decouple the current source from the measurement system during when the EEG signal and the temperature of the probe's tip are measured. When the input pulse signal V_{stim} becomes high ($=0\text{V}$) and the switches are turned ON, the output current I_{out} can be given by eq. (2).

$$I_{out} = \frac{-V_{SS}}{R_{ctrl}} \quad (2)$$

The ON resistance R_{ON} must be sufficiently low to be compatible with the output voltage capability of the circuit.

3 Simulation Results

HSPICE simulations were performed to confirm the operation of the proposed circuits. The bipolar supply voltage of $\pm 1.5V$ are applied. The Op amp in Fig. 5 is used as the Op amp in the circuits in Figs. 3 and 4. The bias voltage V_{bias} of the Op amp is set to $-0.5V$. The element values used in the simulations are mentioned in Table 2.

Figure 6 shows the epileptogenic ECoG signal measured by a subdural electrodes array. In Fig. 6, after the normal ECoG signal measured for 2 seconds

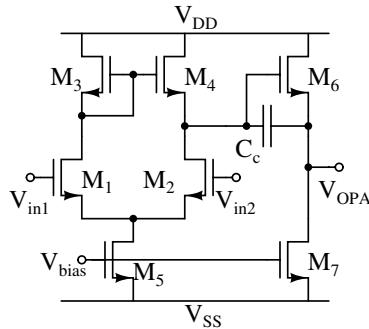


Fig. 5. Op amp used in the simulations

Table 2. Element values used in the simulations

R_1, R_2, R_5, R_6	$500k\Omega$
R_G, R_3, R_4	$10k\Omega$
Analog switches	$R_{ON} = 10\Omega @ V_{stim} = 3V$
SW1, SW2	$R_{OFF} = 10M\Omega @ V_{stim} = 0V$

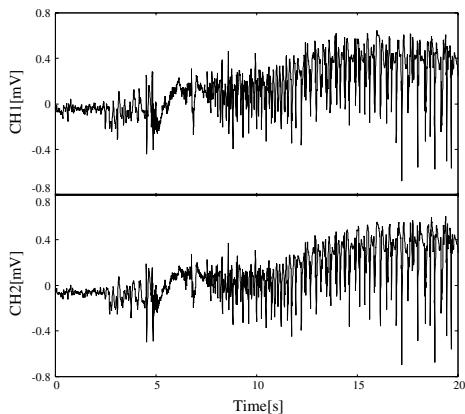


Fig. 6. The epileptogenic ECoG signal used in the simulations for the instrumentation amplifier

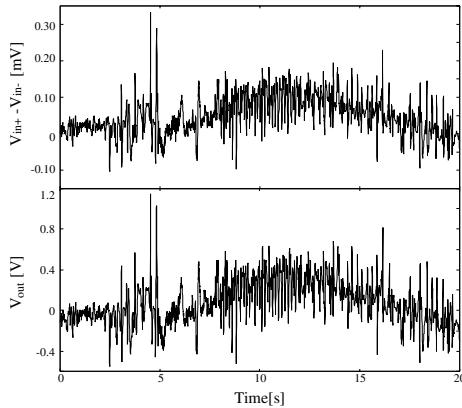


Fig. 7. The differential input (upper) and the simulated output signal (lower) of the circuit in Fig. 3

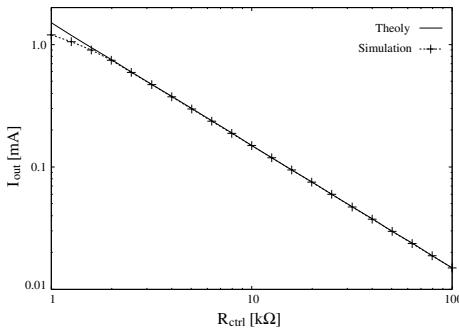


Fig. 8. The provided current I_{out} when R_{ctrl} is changed

from the beginning, the high amplitude epileptic burst appears and lasts for more than 15 seconds. The voltages of CH1 and CH2 are applied to $V_{\text{in}+}$ and to $V_{\text{in}-}$ in Fig. 3, respectively. The epileptogenic ECoG signal is successfully amplified as shown in the lower part of Fig. 7

HSPICE simulations of the current source in Fig. 4 were performed with the Op amp in Fig. 5. The theoretical output amplitude given by eq. (2) and the simulated output amplitude are plotted in Fig. 8. The pulse current is provided to the load Z_{out} and its amplitude is controlled by the R_{ctrl} as shown in Fig. 8. Here, Z_{out} which models the load impedance (including brain tissue and the electrodes) is simulated as a resistor of 800Ω .

The error between the theoretical plot and the simulated plot is increased when the output current is larger than 1mA, because of the output capability of the Op amp. Therefore the maximum output current is 1mA. Even though the specification in Table II is not satisfied, this value is sufficiently large for the proposed application, since the burst threshold of the nerves in epileptogenic focus is relatively low.

Since the output voltage of the Op amp A4 is limited by its output capability, the contact area of the electrodes must be sufficiently large in order to reduce the contact impedance and the ON resistance of the switches must be sufficiently low.

4 Concluding Remarks

A system for intracranial EEG measurement, brain stimulation, and for minimally invasive treatment of intractable epilepsy was proposed in this paper. The freezing function of the microprobe confirmed with the experiments using saline solution solidified by gelatin. The operation of the measurement/stimulation circuits was confirmed by the HSPICE simulations.

References

1. World Health Organization, Epilepsy: aetiology, epidemiology and prognosis (2001), <http://www.who.int/mediacentre/factsheets/fs165/en/>
2. Yamakawa, T., Grigorievich, Z.L., L'vovich, Z.Y.: Micro Freezing Probe and Its Control, Japanese patent, Application No.TOKUGAN2008-102974 (2007)
3. Robblee, L.S., Lefko, J.L., Brummer, S.B.: Activated Iridium: An Electrode Suitable for Reversible Charge Injection in Saline solution. *J. Electrochem. Soc.* 130, 731–733 (1983)
4. Kitchin, C., Counts, L.: A Designer's Guide to Instrumentation Amplifiers, 3rd edn. Analog Devices, Inc., Norwood (2006)
5. Niedermeyer, E., Silva, F.: *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*, 5th edn. Lippincott Williams & Wilkins, Philadelphia (2004)

Generalization in Linguistic Category Formation Based on the Prototype Effect

Tamami Sudo and Ken Mogi

¹ Interdisciplinary Graduate School of Science and Engineering,

Tokyo Institute of Technology

sudo@brn.dis.titech.ac.jp

² Sony Computer Science Laboratories, Inc.

kenmogi@qualia-manifesto.com

Abstract. The development of cognition in humans is tightly coupled with that of language and conceptualization. It's known that children as early as one year after birth are able to recognize the various objects as belonging to a single category. Here we report a series of experiments investigating how humans perceive the environment in terms of categorization based on the "prototype effect", studied in the literature as a characteristic feature of categorization. We focused on two parameters ("typicality" and "similarity") in the process of perceptual categorization. Using a direct rating paradigm in categorical judgment, we derived continuous parameter measures for the "typicality" and "similarity" of objects based on the subject's estimation. The results suggest some cognitive constraints on the development of language, as it relates to the perception of sensory and motor information in the interaction with the environment.

1 Introduction

Categorization, the cognitive process of finding a significant relationship among objects in natural scene, is a fundamental aspect of effective human communication. It is also tightly coupled with language and conceptualization. For example, we recognize an apple and a banana to be in the same category (food) even though they are dissimilar in appearance, and we consider an apple and a red ball to be in different categories even though they are similar in shape and color. It is known that infants are aware of the category of objects in daily life before explicit vocabulary acquisition starts. Infants only a few months old are able to separate complex visual stimuli into generic object categories. Even though the various attributes such as color, texture, size are not the same for real objects, pictures and photos, they are able to perceive them as belonging to a certain category. Several studies have shown that infants from at least 3 months old can discriminate between basic object categories [1, 2]. In another primate research, monkeys were able to categorize computer-generated stimuli as "cats" and "dogs", and they defined the category boundary [3]. Thus the ability of categorization plays an essential role in the cognition preceding language acquisition. A familiar object such as an apple can take various forms, when it is cut and dressed. Children one year after birth are apparently able to recognize these physically different entities as belonging to the same category [4]. It has been

suggested that the ability of categorization is a meta-knowledge tightly coupled with an infant's language acquisition, and is likely to play an essential role in its overall cognitive development.

Historically, various models of categorization have been suggested. The prototype model and the exemplar model are two important models suggested. Prototype model is a mode of graded categorization in which an observer's category representation consists of a single summary abstraction that is the central tendency of the individual members of the category [5]. Classification decisions are then based on the similarity of an individual test item to the category prototypes [6]. For example, prototype effect would consider a category like bird as consisting of different elements which have unequal status - e.g. a robin is more prototypical of a bird than, say a penguin. In contrast, in the exemplar model individual members of a category are stored and classification decisions are based on the separate similarities of a test item to each of the stored items [6]. According to the exemplar model, learning involves the storage of instances in memory. The exemplar model does not assume that learning involves the computation of summary representations for categories or other groups of stimuli as is presumed in prototype model. Instead, it is assumed that each encounter with a stimulus leaves a separate trace in memory, and that subsequent categorization, identification or recognition depends on the retrieval of these specific memory traces. Early theories of categorization assumed that either prototypes, or exemplars, or the others were exclusively used to mentally represent categories of objects. More recently, various hybrid theories of categorization have been proposed that involve mixtures of these different forms of category representation [8, 9]. Prototype theory and its surrounding models also play a central role in language acquisition, as part of the mapping from phonological structure to semantics.

Here we conducted a series of experiments using familiar objects with the same set of stimuli as our previously reported research on infant pre-linguistic cognitive development using the preferential looking paradigm by [4] in order to understand the categorization process based on the prototype effect. Specifically, we focused on two major parameters, "typicality" (how typical an example of category is) and "similarity" (how similar a prototypical member of category is) in perceptual categorization. Using the direct rating paradigm in categorical judgment for adult subjects, we derived a continuous parameter measures for the "typicality" and "similarity" of objects based on the subject's estimation.

2 Experiment

In the former study [4], we investigated the infants' ability to use linguistic cues to associate objects with appropriate categories at one year after birth [4]. From these results, the infants were shown based on a preferential looking paradigm to be successful in categorizing each object with the aid of linguistic labels and to perceive physically different objects as belonging to a single category even before explicit verbal ability emerged. In addition, in terms of the infants' mean looking time,

significant effects of physical shape of the object were observed in certain categories. In order to clarify the terms behind these differences, we conducted a series of experiments for adult subjects using two different experimental paradigms, measuring reaction time in categorical judgment task and direct rating score in questionnaire as previously mentioned. Finally, we analyzed the adult data in comparison with the infants' data.

2.1 Categorical Judgment Task

Participants. Seven subjects (four males, three females) with normal or corrected-to-normal vision participated in the study. The participants' mean age was 26.6 years old.

Stimuli. Three colored stimuli (pictures or photos) are used for each category, resulting in nine stimuli in total (given in Figure 1). These stimuli are identical to the ones used in former study[4]. Subjects were seated at 45 cm away from the laptop screen.

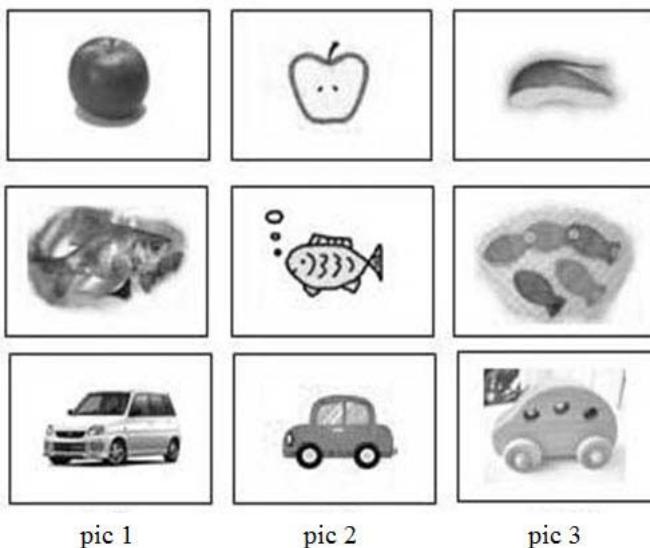


Fig. 1. Stimulus used in the study

Procedure. Each trial consisted of the presentation of a category name for 0.5 s and a continuous presentation of a pair of pictures for 0.5 s (Figure 2). The subjects were instructed to indicate the appropriate stimulus matching the category name by pressing one of the two assigned keys as soon as the two pictures in different categories were shown on the monitor. The reaction time was measured and analyzed. The order of stimulus presentation in each test block was counterbalanced. Test blocks consisted of nine trials. Subjects conducted six different test blocks 10 times each. There were 540 trials in total.

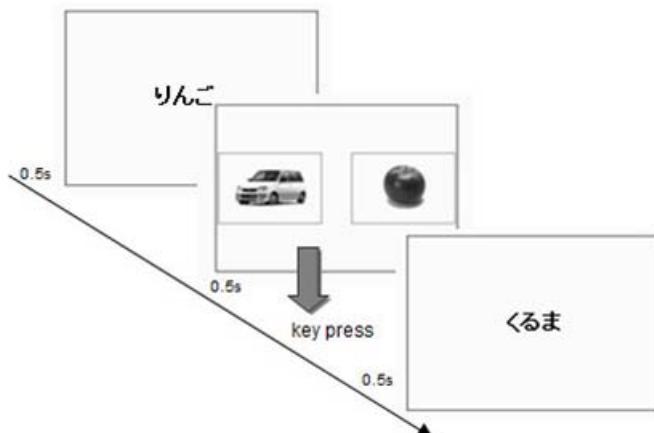


Fig. 2. The stimulus presentation procedure

2.2 Direct Rating in Questionnaire

In conjunction with the categorical judgment tasks, we conducted a survey in the form of a questionnaire on the visual stimuli used in the experiment. We focused on two major parameters, "typicality" (how typical an example of category is) and "similarity" (how similar a prototypical member of category is) in perceptual categorization. Subjects were asked to rate, on a scale from one (least typical in typicality and similar in similarity) to seven (most typical in typicality, and similar in similarity). 230 undergraduates were surveyed. The average age of the people surveyed was 20.0 years old.

3 Results

3.1 Reaction Time in Categorical Judgment

A repeated-measure ANOVA with pairing objects (target vs. other) revealed significant differences [food: $F(1, 72) = 67.21$, ($p < 0.001$), creature: $F(1, 66) = 64.46$, ($p < 0.001$), artifact: $F(1, 72) = 83.28$ ($p < 0.001$), total: $F(1, 214) = 205.9$, ($p < 0.001$)]. This result is consistent with the interpretation that the infant is able to recognize the physically different forms of object as belonging to the single category, even before explicit verbal ability emerges. In addition, significant effects of the physical shape of the object were observed; except category-car (artifact) [food: $F(2, 31) = 5.690$, $p = 0.008$, creature: $F(2, 30) = 4.116$, $p = 0.026$, artifact: $F(2, 34) = 1.561$, $p = 0.225$ (n.s.), total: $F(2, 105) = 7.734$, ($p < 0.001$)]. According to Tukey post-hoc test, there are significant differences except between the picture 2 and 3 of apples ($p = 0.101$), between the picture 1 and 2 of fish ($p = 0.093$) and between the picture 1 and 2 of cars ($p = 0.372$). These data were put under further evaluations by converted standard score. There is a significant difference [$F(2, 1257) = 3.00$, ($p < 0.01$)], which,

following the Tukey post-hoc test, shows virtually the same tendency of results as the ones from the reaction time with a few exceptions as previously mentioned ; i.e., between the picture 1 and 2 of cars ($p = 0.129$). However, significant differences were newly revealed with apples (picture 2 and 3) ($p = 0.026$) and fish (picture 1 and 2) ($p = 0.004$).

3.2 Comparison with the Questionnaire

Direct rating score about the typicality and similarity of each visual stimulus are shown in table 1. Note that the similarity for one of the most typical stimuli among each category was scored as 7 automatically. These results include the results of categorical judgment for line chart in Fig.3.

Comparison of the reaction time of the experimental data in the categorical judgment with the questionnaire revealed a certain type of tendencies. The reaction time for stimuli which got high score in the questionnaire (more representative member in the category) was shorter than for the ones which got low score (less representative member in the category). Each category exhibited a negative correlation between the typicality and the reaction time (apple: $r = 0.17$, fish: $r = 0.26$, car: $r = 0.15$). These results are consistent with the former studies on categorization based on the prototype model [5], and suggest that the more representative members in the category are easier to invoke the prototypical member in the inner representation, indicating that these stimuli have taxonomic hierarchies among each category.

Table 1. Average scores of typicality and similarity for visual stimuli

	typicality	similarity
Category-apple		
pic1: Whole apple photo	6.9	7.0
pic2: Half apple picture	4.2	3.1
pic3: Rabbit-shaped apple photo	4.6	1.6
Category-fish		
pic1: Line drawing fish	5.8	1.9
pic2: Goldfish photo	6.0	7.0
pic3: Cloth fish photo	3.3	2.7
Category-car		
pic1: White car photo	6.7	7.0
pic2: Red car picture	5.8	1.9
pic3: Woody toy car photo	3.4	1.8

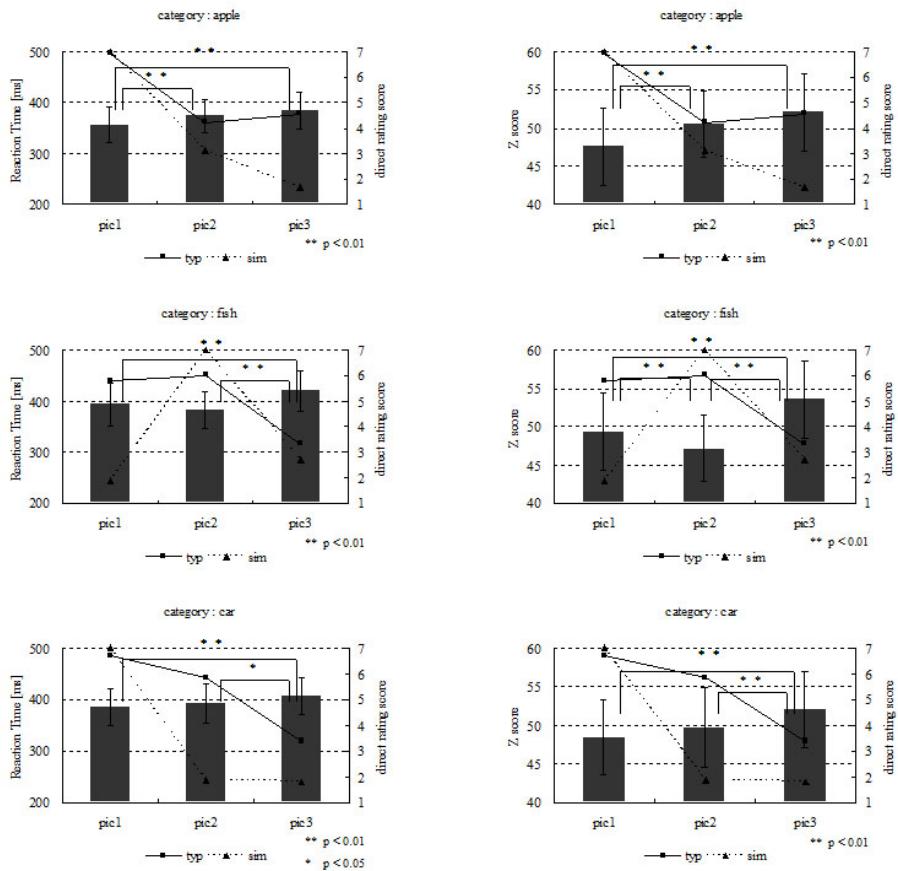


Fig. 3. Average values of reaction time, converted standard scores and direct rating scores of each object from questionnaire (left: reaction time, right: the standard score). The bar charts show the reaction times for each target object. The line charts show the direct rating score from the questionnaire (solid line : typicality, dashed line : similarity).

4 Discussion

In the infants' categorical judgment task using preferential looking paradigm, we have reported earlier that the relatively more representative members in the category are easier to invoke the prototypical member in the inner representation [4].

In this study, we considered the two following factors. First, we have investigated whether the objects which resulted in a longer preferential looking time in the infants' experiments are actually the prototypical members among the category. Second, we investigated whether these taxonomic hierarchies in categorical representation from the adults' experiments exhibit a similar tendency as in the experiments in infants. From our results, we conclude that the reaction time of categorical judgments was negatively correlated with the direct rating score of prototypical judgments from the

adult questionnaire. This result suggests a coherent tendency underlying linguistic and pre-linguistic judgments in both infants and adults, and is consistent with the hypothesis that humans have certain types of taxonomic hierarchies in categorical representation related to the prototype effects.

Acknowledgments. This work was supported by grant from the Matsushita International Foundation, Japan.

References

1. Eimas, P.D., Quinn, P.C.: Studies on the formation of perceptually based basic-level categories in young infants. *Child Development* 65, 903–917 (1994)
2. Quinn, P.C., Eimas, P.D.: Perceptual cues that permit categorical differentiation of animal species by infants. *Journal of Experimental Child Psychology* 63, 189–211 (1996)
3. Freedman, D.J., Riesenhuber, M., Poggio, T., Miller, E.K.: Categorical representation of visual stimuli in the primate prefrontal cortex. *Science* 291, 312–316 (2001)
4. Sudo, T., Mogi, K.: Mechanism of perceptual categorization in the pre-linguistic period. *IEICE Transactions on Information and Systems* E91-D(6), 1649–1655 (2008)
5. Lakoff, G.: *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. The University of Chicago Press, Chicago (1987)
6. Rosch, E., Mervis, C.B.: Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology* 7(4), 573–605 (1975)
7. Nosofsky, R.M.: Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General* 115(1), 39–57 (1986)
8. Minda, J.P., Smith, J.D.: Prototypes in category learning: The effects of category size, category structure, and stimulus complexity. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 27(3), 775–799 (2001)
9. Johansen, M.K., Palmeri, T.J.: Are there representational shifts during category learning? *Cognitive Psychology* 45(4), 482–553 (2002)

Model of the Activity of Hippocampal Neurons Based on the Theory of Selective Desensitization

Atsuo Suemitsu¹, Yasuhiro Miyazawa², and Masahiko Morita³

¹ School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa, Japan

² Hitachi Consulting Co., Ltd., 2-16-4 Konan, Minato-ku, Tokyo, Japan

³ Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1 Tennodai, Tsukuba, Ibaraki, Japan

sue@jaist.ac.jp, y Miyazawa@hitachiconsulting.co.jp,
mor@bcl.esys.tsukuba.ac.jp

Abstract. The integration of different kinds of information is thought to be a key function of the hippocampus. On the basis of a computational theory, we hypothesize that the hippocampus performs information integration using selective desensitization of CA3 neurons, and construct a model of the hippocampal trisynaptic network on the basis of this hypothesis. This model can reproduce the results of a physiological experiment in which rat hippocampal place cells in various environments were recorded, whereas the conventional layered model cannot. This result, together with some other physiological evidence, supports our hypothesis.

1 Introduction

The hippocampus is mainly composed of the dentate gyrus (DG), the CA3 region (CA3) and the CA1 region (CA1). A major input pathway to these regions is the perforant path from the entorhinal cortex (EC), which receives input from all sensory areas (Fig. 1). The hippocampus is known to be involved in various functions, such as spatial memory and cognition.

Among them, it is thought that the hippocampus plays a key role in the integration of different kinds of information. For example, the place cells [1] responding to a restricted location change their activity depending on the environment or the task, even in the same location [2]. Furthermore, Lee et al. [3] observed neuronal activities related to information integration in the rat hippocampus by rotating two kinds of cues relative to each other. Their study is very important in providing detailed data on place cells of CA3 as well as CA1. This result, however, is difficult to explain using conventional neural network models.

In this paper, we hypothesize that the selective desensitization method is used for information integration in the hippocampus. We also construct a model based on this hypothesis, and show that this model can reproduce the empirical data presented by Lee et al. [3].

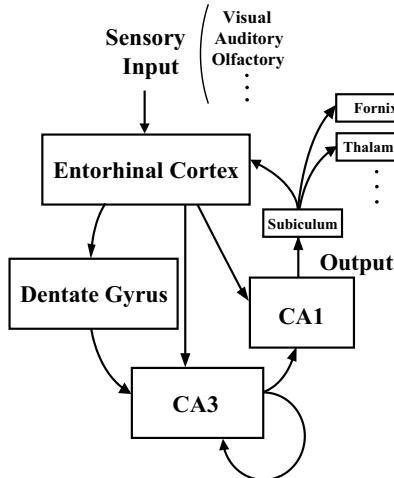


Fig. 1. Major pathways of the hippocampus

2 Background

2.1 Selective Desensitization Theory

Let us consider a simplified task of information integration in a neural network. Specifically, the target pattern $T^{\mu,\nu}$ to be recalled depends on the combination of input patterns S^μ ($\mu = 1, \dots, p$) and C^ν ($\nu = 1, \dots, q$), where these patterns are n -dimensional binary vectors and are randomly generated, and where n is a sufficiently large value. A conventional two-layer network consisting of the input layer with $2n$ elements and the output layer with n elements cannot learn and perform this task unless pq is very small because of a serious computational difficulty, which we call the problem of connection averaging.

To explain this problem, consider the case in which $\mu = \nu = 2$, $T^{1,1} = T^{2,2} = T^0$ and $T^{1,2} = T^{2,1} = T^1$ (generalized XOR task). This task cannot be solved using the two-layer network, since S^μ and C^ν are associated equally with both T^0 and T^1 , and the signal vector from the input layer to the output layer is an intermediate one between T^0 and T^1 regardless of what input patterns S^μ and C^ν are given. More generally, as individual input patterns correspond to multiple target patterns, the connections from the input layer to the output layer become less significant because of a similar averaging effect.

This problem is not resolved by introducing middle layers or recurrent connections, or by improving the learning algorithm. Although the use of a three-layer network with a middle layer can solve the above task, it requires local or orthogonal representations corresponding one-to-one to the combinations of S^μ and C^ν in the middle layer to avoid the one-to-many correspondence causing connection averaging. However, such representations greatly impair the generalization ability. Consequently, conventional neural networks, in which the output of elements is determined by the weighted sum of the inputs, cannot cope with both the learning ability for many samples and the generalization ability for unlearned data.

The simplest method of resolving the above problem is probably the selective desensitization method [4], which desensitizes a part of the elements representing $S = (s_1, \dots, s_n)$ depending on $C = (c_1, \dots, c_n)$. Here, “desensitize” means to fix the output at a neutral value. In mathematical terms, the output x_i of the i -th element is given by

$$x_i = g_i(s_i - \bar{x}_i) + \bar{x}_i, \quad (1)$$

where \bar{x}_i denotes the average of x_i and g_i is the output gain which usually takes 1 but becomes 0 when the element is desensitized. We will deal with the simplest case where $g_i = 1 - c_i$, that is, the element is desensitized when $c_i = 1$, and denote by $S(C)$ the output pattern modified in this manner.

By applying this method to the input layer of the two-layer network, the generalized XOR task can be easily solved because pattern sets $\{S^1(C^1), S^2(C^2)\}$ and $\{S^1(C^2), S^2(C^1)\}$ are linearly separable. The limitation on the numbers of p and q in the information integration task is also removed, since S^μ is not directly associated with multiple target patterns but $S^\mu(C^\nu)$ is associated with a single target pattern so that the problem of connection averaging is avoided. Moreover, the generalization ability, which is a crucial merit of distributed representations, is not lost because similarity relations between input patterns are preserved, that is, if S^1 and S^2 or C^1 and C^2 are similar, $S^1(C^\nu)$ and $S^2(C^\nu)$ or $S^\mu(C^1)$ and $S^\mu(C^2)$ are similar.

2.2 Physiological Evidence

There exists some physiological evidence suggesting that selective desensitization actually occurs in hippocampal CA3.

First, mossy fibers of DG establish unique giant synapses on the CA3 pyramidal cell. These synapses are thought to strongly influence the activity of CA3 neurons because they form multiple synaptic contacts close to the cell body. Also, these synapses contain Zn^{2+} in large amounts and release Zn^{2+} with the activity of DG neurons [5], then the Zn^{2+} inhibits NMDA receptors. Thus, this makes it possible to decrease the gain of CA3 neurons selectively.

Second, most mossy fibers are connected with inhibitory interneurons interposed between DG and CA3 [6]. These neurons may function as presynaptic inhibitors.

The other piece of evidence is the phenomenon called “rate remapping”, in which the locations of place fields remain unchanged, but most CA3 neurons change their firing rates when the color or shape of the recording chamber are changed [7]. This phenomenon can be interpreted to indicate that CA3 neurons are selectively desensitized depending on the environment.

3 Proposed Model

3.1 Target Physiological Data

Lee et al. [3] trained rats to run clockwise on a circular track in a curtained environment, as shown in Fig. 2, where the circular track was composed of four

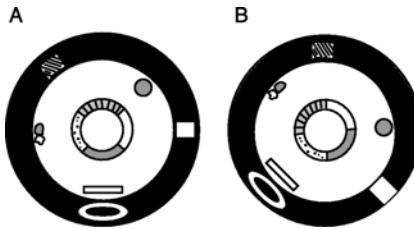


Fig. 2. Apparatus (reproduced from Ref.[3])

different textured surfaces as the local cue, and six objects were present as the distal cue around the curtain. During training, a constant combination of the local and distal cues was used as the standard environment (STD, Fig. 2A).

They then recorded neuronal activities in CA3 and CA1 while the trained rats were running clockwise on the circular track in STD, and in the mismatch environment (MIS), in which the local and distal cues were rotated by the same angle anticlockwise and clockwise, respectively (Fig. 2B). Mismatch angles between the local and distal cues were set to 45° , 90° , 135° , and 180° . Subsequently, for each neuron exhibiting a place field response, they calculated the firing rate for each 1° bin of the track and constructed a population firing rate vector for each of the 360 bins, and then produced 360×360 correlation matrices by calculating the correlation coefficient between these vectors for STD and MIS.

As a result, they found that with an increase in the mismatch angle, the band of high correlation for CA3 shifted right-downward, while that for CA1 disappeared, as shown later in Fig. 4(a). This indicates that the place fields of many cells change with a rotation of the cues.

3.2 Model Construction

Taking into account the experimental design and results described above, we constructed a model based on the selective desensitization theory and the following assumptions: the rat hippocampus calculates its absolute position in the curtained environment by integrating information on the local and distal cues; such information is represented separately in EC, in which information on the local cue is directly fed to CA3, and information on the distal cue modifies the activity in CA3 through DG.

Specifically, information on the local and distal cues at position θ on the track in STD is represented by activity patterns L^θ and D^θ in layers EC-L and EC-D, respectively. In the following, let us consider $\theta = (0, 1, \dots, 359)$ and cell number $n = 360$ in each layer. The i -th components of L^θ and D^θ take a value of 1 when $\theta = i - 1$ and decrease exponentially with distance from position θ . In mathematical terms,

$$l_i^\theta = e^{-\alpha_L \phi(\theta-i+1)}, \quad (2)$$

$$d_i^\theta = e^{-\alpha_D \phi(\theta-i+1)}, \quad (3)$$

$$\phi(u) = \begin{cases} |u| & (|u| < 180) \\ 360 - |u| & (|u| \geq 180), \end{cases} \quad (4)$$

where α_L and α_D are positive constants.

The architecture of the constructed model (referred to as the selective desensitization model), each layer of which is described below, is shown in Fig. 3(a).

EC. The output of EC-D is D^θ , but the output s_i of the i -th cell in EC-L is the i -th component of $S^\theta = (l_{\sigma(1)}^\theta, \dots, l_{\sigma(360)}^\theta)$, which is generated by randomizing the order of components of $L^\theta = (l_1^\theta, l_2^\theta, \dots, l_{360}^\theta)$ to eliminate the high correlation between L^θ and D^θ . Here, $\sigma(i)$ is a permutation function and is independent of θ .

DG. The output of EC-D is sent to DG, and is then transformed into a binary vector. In other words, the output of the i -th cell in DG is given by

$$c_i = h(d_i - \beta). \quad (5)$$

Here, $h(u)$ is the Heaviside function, where $h(u) = 1$ for $u \geq 0$ and 0 for $u < 0$, and β is a threshold which is set to a small value so that c_i may have a value of 1 over a relatively wide range of θ .

CA3. The i -th cell in CA3 receives the output s_i from EC-L, and is desensitized when $c_i = 1$ in DG. In mathematical terms, the output x_i of CA3 is given by

$$x_i = (1 - c_i)(s_i - \bar{x}_i) + \bar{x}_i, \quad (6)$$

where \bar{x}_i is the mean level of x_i . However, since the mean level \bar{x}_i is very low in this model, we regard \bar{x}_i as zero for simplicity. Accordingly, the output x_i can be expressed as

$$x_i = (1 - c_i)s_i. \quad (7)$$

CA1. Each cell in CA1 receives the output of all cells in CA3 and emits the pattern representing position θ . Specifically, the i -th cell acts according to

$$y_i = f \left(\sum_j w_{ij} x_j \right), \quad (8)$$

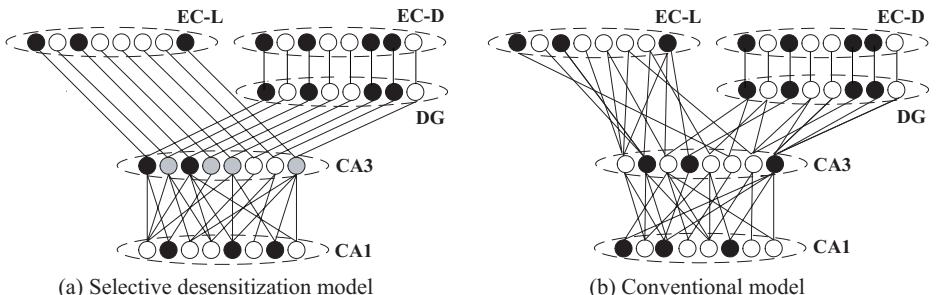


Fig. 3. Architecture of the hippocampal models

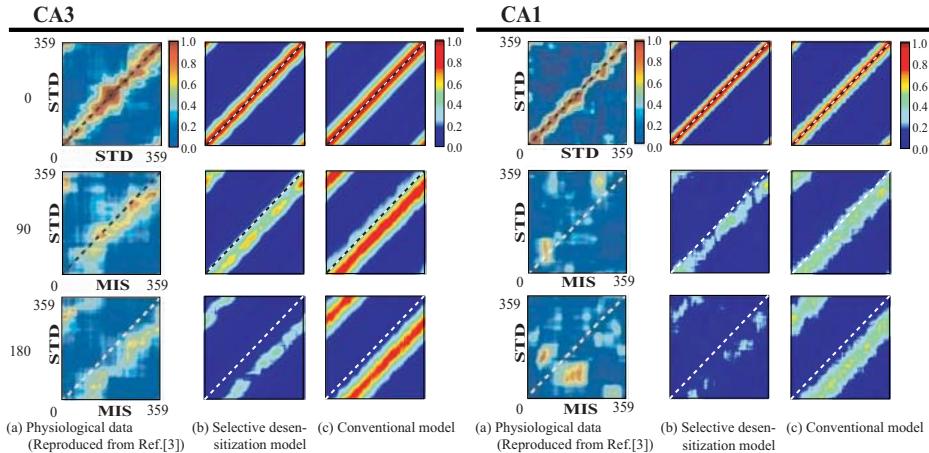


Fig. 4. Correlation matrices between population activity vectors

where w_{ij} is the synaptic weight from the j -th cell in CA3 and $f(u)$ is a monotonic sigmoid function given by $f(u) = 1/(1 + e^{-u})$. Learning of the synaptic weight is performed using the target pattern $T^\theta = (t_1^\theta, \dots, t_n^\theta)$ as a teacher signal according to

$$\Delta w_{ij} = \varepsilon(t_i^\theta - y_i)x_j, \quad (9)$$

where ε is a positive constant and the i -th component of T^θ is defined as

$$t_i^\theta = e^{-\alpha_T \phi(\theta-i+1)}. \quad (10)$$

For comparison, we also constructed another model (referred to as the conventional model) without selective desensitization. As shown in Fig. 3(b), this model is a kind of simple multilayer network, in which EC-L and DG are the input layers, CA3 the association layer, and CA1 the output layer. The cells in EC-L and DG are fully connected to the cells in CA3 through synaptic weights w_{ij}^S and w_{ij}^C , respectively. These synaptic weights are determined by learning with an appropriate teacher signal. The output of CA3 is given by

$$x_i = f \left(\sum_j w_{ij}^S s_j + \sum_j w_{ij}^C c_j \right). \quad (11)$$

CA1 of the conventional model is the same as that of the selective desensitization model.

3.3 Computer Simulation

We performed a computer simulation using the selective desensitization model and the conventional model. First, we prepared 360 pairs of patterns (S^θ, D^θ) and trained both models by feeding these pairs into EC. Training was repeated 20

times for each pair. The parameters were common to both models: $\alpha_L = 0.048$, $\alpha_D = 0.032$, $\alpha_T = 0.062$ and $\varepsilon = 0.15$. Threshold β for the selective desensitization model and the conventional model was set to 0.1 and 0.75, respectively.

After training, we examined the response of the cells in CA3 and CA1 in STD and MIS, where we input pairs of patterns $(S^{\theta-\Delta\theta}, D^{\theta+\Delta\theta})$ for each environment (mismatch angle $2\Delta\theta = 0, 45, 90, 135, 180$). To further analyze this simulation data in the same way as Lee et al. [3], we calculated the correlation matrix between the population activity vectors for STD and those for MIS according to

$$R_{ij} = \frac{\sum_{k=1}^n FR_{STD_{i,k}} FR_{MIS_{j,k}}}{\sqrt{\left(\sum_{k=1}^n FR_{STD_{i,k}}^2\right) \left(\sum_{k=1}^n FR_{MIS_{j,k}}^2\right)}}, \quad (12)$$

where $FR_{STD_{i,k}}$ and $FR_{MIS_{j,k}}$ are the output of the k -th cell at position i in STD and at position j in MIS, respectively. Fig. 4 shows the correlation matrices in STD (top row), 90° MIS (middle row), and 180° MIS (bottom row). We can see that the graphs for the selective desensitization model are more similar to the empirical data than are those for the conventional model. The conventional model reproduces the shift of the band of high correlation in CA3, but not its disappearance in CA1, while the selective desensitization model reproduces both phenomena.

This result strongly suggests that selective desensitization actually occurs in CA3, since the essential difference between the two models is whether or not layer CA3 is based on selective desensitization.

4 Concluding Remarks

On the basis of a computational theory of information integration in the neural network, we hypothesized that the hippocampus performs information integration using selective desensitization of CA3 neurons. We also constructed a model of the hippocampal trisynaptic network on the basis of this hypothesis, and demonstrated that this model can reproduce the activity of rat hippocampal place cells in a dynamically changing environment.

To verify our hypothesis, however, further physiological and theoretical investigations are required. The development of a model for dynamic processing is also a subject of future study.

Acknowledgments. This work was supported by a Grant-in-Aid for Scientific Research from JSPS and a Grant-in-Aid for Scientific Research on Priority Areas from MEXT.

References

1. O'keefe, J., Nadel, L.: The Hippocampus as a Cognitive Map. Clarendon Press, Oxford (1978)
2. Markus, E.J., Qin, Y.L., Leonard, B., Skaggs, W.E., McNaughton, B.L., Barnes, C.A.: Interactions between Location and Task Affect the Spatial and Directional Firing of Hippocampal Neurons. *Journal of Neuroscience* 15, 7079–7094 (1995)
3. Lee, I., Yoganarasimha, D., Rao, G., Knierim, J.J.: Comparison of Population Coherence of Place Cells in Hippocampal Subfields CA1 and CA3. *Nature* 430, 456–459 (2004)
4. Morita, M., Murata, K., Morokami, S., Suemitsu, A.: Information Integration Ability of Layered Neural Networks with the Selective Desensitization Method. IEICE Transactions on Information and Systems (Japanese edn.) J87-D-II, 2242–2252 (2004)
5. Ueno, S., Tsukamoto, M., Hirano, T., Kikuchi, K., Yamada, M.K., Nishiyama, N., Nagano, T., Matsuki, N., Ikegaya, Y.: Mossy Fiber Zn²⁺ Spillover Modulates Heterosynaptic N-methyl-D-aspartate Receptor Activity in Hippocampal CA3 Circuits. *Journal of Cell Biology* 158, 215–220 (2002)
6. Acsady, L., Kamondi, A., Sik, A., Freund, T., Buzsaki, G.: GABAergic Cells Are the Major Postsynaptic Targets of Mossy Fibers in the Rat Hippocampus. *Journal of Neuroscience* 18, 3386–3403 (1998)
7. Leutgeb, S., Leutgeb, J.K., Barnes, C.A., Moser, E.I., McNaughton, B.L., Moser, M.B.: Independent Codes for Spatial Memory and Episodic Memory on Hippocampal Neuronal Ensembles. *Science* 309, 619–623 (2005)

EEG-Based Classification of Brain Activity for Brightness Stimuli

Qi Zhang

Sony Computer Science Laboratories, Inc.

Takanawa Muse Bldg., 3-14-13, Higashigotanda, Shinagawa-ku, Tokyo, 141-0022, Japan
zhangq@csl.sony.co.jp

Abstract. Brain-computer interface (BCI) is providing a new channel for human to interact with computers and devices. Researches have been conducted on motor imaginary detection to aid individuals who cannot use any motor system for communication, to operate the computer and other devices by prosthesis. In this paper, we developed a classifier that can distinguish EEG signals responding to visual stimuli. Brightness difference was used as a preliminary case study of this approach as it is easily controllable and provides stable stimuli. The brain activities were measured by an electroencephalogram (EEG) system, and an adaptive auto-regressive (AAR) model was utilized to extract the features of different brain states corresponding to stimuli in different brightness. A minimum distance analysis (MDA) classifier was created to discriminate different brightness perception. By this means, different perceptions of four brightness conditions were decoded successfully based on the brain activities of single trials.

1 Introduction

A brain-computer interface (BCI) is a system which includes a means for measuring neural signals from the brain, a method/algorithm for decoding these signals and a methodology for mapping this decoding to a behavior or action [1], [2]. One of the primary rationales for developing BCI systems is to provide a means of communication for individuals with severe neurological disease and/or injury, who have lost the volitional control of their muscles, but keep their cognitive capacities unaffected. Thus decoding of the direct measurement of neural activity from their brains (part of the central nervous system) can be used to bypass their malfunctioning peripheral nervous systems. In addition to clinical applications, BCI systems potentially provide a new type of interface with which people who have normal neurological function, can interact with a computer/machine.

Besides of patients, BCI systems also can be used by healthy people as a further communication path. By detecting human feeling based on the brain activity, the environment can be adjusted automatically to make the people more comfortable and enjoyable bypassing any consciously involvement. Such BCI systems to healthy people can improve the quality of their lives.

BCI systems are typically divided in invasive and non-invasive systems, based on how the neural signals are collected. Invasive systems relying on implanted arrays of

electrodes, are common in experiments involving rodents and nonhuman primates [3], and are well suited for decoding activity in the cerebral cortex. Non-invasive systems are better suited for situations in which a surgical implementation is not possible or warranted and thus have a much wider field of application. Electroencephalography (EEG) is the most commonly used measurement modality for noninvasive recordings.

In BCI researches based on the EEG measurement, BCIs based on evoked potentials like SSVEP [4] and P3 [5], [6], or BCIs based on unstimulated brain signals like ERD/ERS of oscillatory features or ERPs of slow cortical potentials [7]-[9] have been the major directions. In these BCI researches, two different approaches exist. The first one starts with presenting general bio-feedback and adapting the system [7], where the subject is confronted with feedback based on a fixed setup such he is able to find how he can control the system. The other one uses machine learning techniques based on one initial training session to attain an individually optimal setup [8], [9].

So far, most of the BCI systems aim the motor disability persons. Motor imagery of the body or cursor movement for motor prosthesis has been the hottest topic [7]-[9]. However, the majority of people have normal or correctable visual ability, to whom visual perception is the most useful sensation obtaining information from the outside world. On the other hand, as the entertainment and IT technology developing, people are not only getting information from the real world experienced by themselves, but also getting more and more information by watching TV programs or navigating around the internet. They spend quite a long time every day to watch on the screen of a TV or a display. The brightness of the TV or the display will give the audience a pleasant feeling when they are watching the contents at proper brightness dynamics. In contrast, when the brightness is too high, people will feel dazzled and easy to get tired. Also in the highlight part some image details even get lost. On the contrary, when the brightness is too dark, the details of the images are hard to be represented, and the audience needs to pay extra effort to get the same information as that in the suitable brightness, so that enjoyment of the contents is impaired. All these will give the audience an unpleasant feeling. In addition, the perception of the brightness is influenced by the environment illumination. The goal of this paper is to develop and test EEG-based classifier of human stimuli-response. As the first step, different brightness visual stimuli will be used as it is easily controllable on TV screen. While correlation between EEG-based classification and subjective feeling is a subject of further studies, capability of EEG-based stimuli-response classifier or alternative approach for accomplishing such classification is the premise for any success in this direction. Here, we attempted a preliminary study if this is at all feasible.

To realize this purpose, we created a classifier to decode the different brightness perceptions based on the non-invasive EEG measurement, aiming people who have normal visual ability. Machine learning approaches were used, taking the advantage of no biofeedback training for subjects. We applied an adaptive auto-regressive (AAR) model to extract the features of different brain states corresponding to stimuli in different brightness. A minimum distance analysis (MDA) classifier was created to decode these mind states.

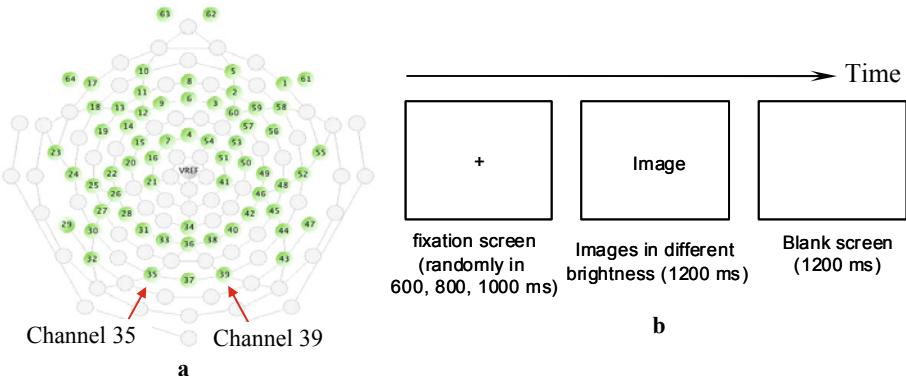


Fig. 1. (a) Positions of electrodes for an EGI 64-channel HydroCel sensor net. (b) Stimulus temporal sequence of one trial.

2 Experiments to Measure the Brain Activity

2.1 Data Acquisition

We used an EGI EEG system with a 64-channel HydroCel Sensor Net to measure the brain activities in the experiments. The positions of the electrode sensors are shown in Fig. 1a. The EEG signals were sampled at 1000 Hz and filtered between 0.01 and 100 Hz.

Stimulus images were adjusted in the way by varying the brightness or contrast to make them appear bright or dim. These stimulus images were displayed on a 40 inch Sony KDL-40X5000 Bravia television set for a wide visual field, with all the automatic adjustment functions turned off. The illuminance of the observation environment is simulated to a living room as about 150 lx. Subjects sat relaxedly on a chair with arm rests in an observation distance of 150 cm to the screen. Four stimulus images with four types of different brightness per image were displayed in random order. Each image repeated for 8 times, so that each brightness condition consisted of 32 trials including four different images. The stimulus sequence in each trial is shown in Fig. 1b. First, a fixation screen was displayed in a randomly varying duration of 600, 800, or 1000 ms. After that, a stimulus image was displayed for 1200 ms, followed by a blank screen for 1200 ms. There were 128 trials in one session.

2.2 ERP Results of Brightness Perception

The brain activities collected by the electrodes located at occipital areas, such as channel 35, 37, 39, were observed with significant differences corresponding to different brightness. Examples of the ERPs (event-related potentials) at electrode channel 35 for one subject are shown in Fig. 2. P1 and P3 components can be observed in all 4 brightness levels, but their average amplitudes and latencies are different. Based on these EEG signals, we created a classifier to decode the different brightness perceptions.

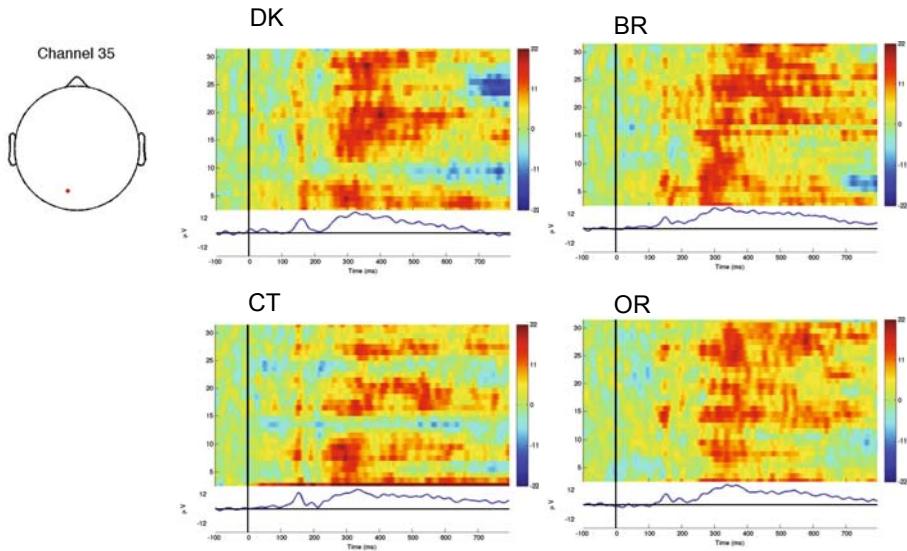


Fig. 2. Examples of ERPs (event-related potentials) for the perception of stimulus images in different brightness conditions at electrode channel 35 on the occipital area for one subject. BR: too bright; CT: too high contrast; DK: too dark; OR: original good.

3 Classification Modeling

3.1 Data Preprocessing

The EEG data were processed offline after the EEG measurement. The acquired EEG raw data were filtered again between 0.5 and 30 Hz. No artifact removing and baseline subtraction were conducted. We used EEG signals from two electrodes of Channel 35 and Channel 39 (Fig. 1a), because their locations are near the occipital visual areas which are considered to be in charge of the processing for brightness perception. The data during the 1200 ms stimulus displaying period were used in our models to decode the different brain states for images in different brightness.

3.2 Feature Extraction by Estimating AAR Parameters

An adaptive autoregressive model was applied to capture the features of the recorded EEG signal. Mathematically, the autoregressive (AR) model (of order p) is described by the following equation [10]:

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + \cdots + a_p y_{k-p} + x_k = \vec{y}_{k-1}^T \vec{a} + x_k$$

where y_i is the observed sample values, a_i represents the autoregressive parameters, x_i represents a zero mean white noise process; \vec{y}_{k-1} is the vector of the past p sample values, and \vec{a} is a vector of p AR parameters.

An adaptive autoregressive (AAR) model has its parameters changing over time to fit the variation of the EEG spectrum. We applied Kalman filtering method to calculate the AAR parameters [11].

3.3 Classification Method

There are many discriminant analysis methods for the feature classification, such as linear discriminant analysis (LDA), logarithmic band power (LBP) estimates, minimum distance analysis (MDA), quadratic discriminant analysis (QDA), and support vector machine (SVM), and so on. We have compared these methods and found that the MDA method achieved the best classification results in our experiments. The distance measure used in our MDA model is the Mahalanobis distance. The Mahalanobis distance $d_c(x)$ of point x with respect to the multivariate normal distribution $N(\mu_c, \Sigma_c)$ is defined by

$$d_c^2(x) = (x - \mu_c) \Sigma_c^{-1} (x - \mu_c)^T,$$

with mean μ_c and the covariance Σ_c are estimated from the training samples [12]. We conducted the classification for each brightness condition against the rest conditions.

3.4 Performance of the Classifier

3.4.1 Evaluation Criteria – Accuracy and Kappa Coefficient

In a classification problem, the proper evaluation of the classifier is described by its confusion matrix defining the relationship between the ‘true’ classes and the output of the classifier. The kappa coefficient is a popular criterion for the agreement between two raters [13]. From the confusion matrix H , we can derive the classification accuracy ACC (overall agreement) as,

$$ACC = p_0 = \frac{\sum_i H_{ii}}{N},$$

where $N = \sum_i \sum_j H_{ij}$ is the total number of samples, H_{ii} are elements of the confusion matrix H on the main diagonal. Based on n_{oi} , n_{io} , the sums of each column and each row, respectively, then the estimate of the chance expected agreement and the kappa coefficient are,

$$p_e = \frac{\sum_i n_{oi} \times n_{io}}{N \times N},$$

$$\kappa = \frac{p_0 - p_e}{1 - p_e}.$$

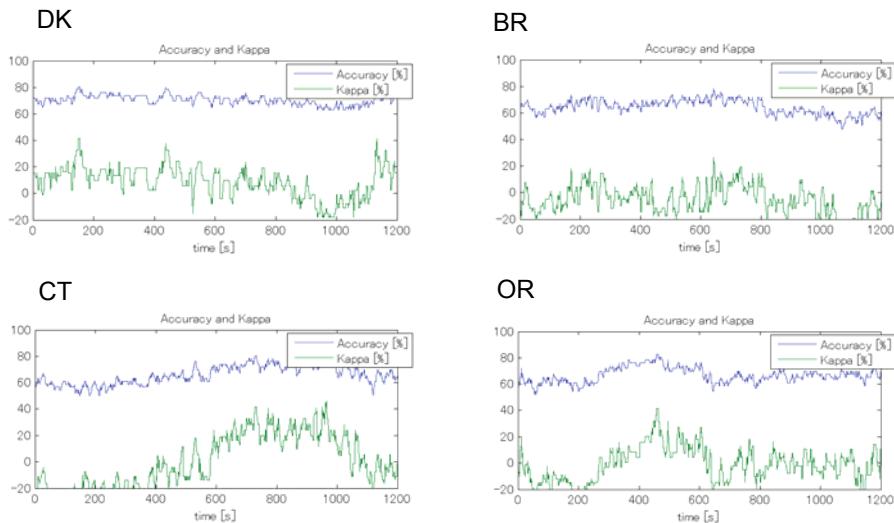


Fig. 3. The accuracies and kappa coefficients of the classifier for the four types of brightness conditions during the 1200 ms displaying time for one subject. BR: too bright; CT: too high contrast; DK: too dark; OR: original good.

3.4.2 Results

Our classifier based on the single trial EEG data achieved the accuracies of around 80%. The accuracies and kappa coefficients of the classifier for the 4 types of brightness conditions during the 1200 ms displaying time for one subject are shown in Fig.3. For the dark image condition, there are accuracy peaks at the time before 200 ms and after 400 ms from the stimulus onset time. This agrees with ERP activity,

Table 1. The best accuracy, kappa coefficient for the brightness conditions

Condition	Dark	Bright	High contrast	Original
Accuracy (%)	80.4	78.3	82.6	82.6
Kappa	0.41	0.27	0.46	0.41

where a significant P1 and P3 components exist. In contrast, the bright condition has higher accuracy after the 200 ms, corresponding to the ERP activity where there is P2 component. In the high contrast brightness condition, the higher accuracies appear after 600 ms from the onset. While for the original suitable brightness images, the accuracy peak is around 400 ms, which is corresponding to the P3 component in ERP.

In addition, the kappa coefficients exceed 40% except for the bright condition, showing that the classification of the brain activities has a moderate agreement with the different brightness conditions. The best accuracy and the kappa coefficient are summarized in Table 1.

4 Discussions

A classifier for the brightness perception was constructed in this paper. The EEG data during the entire stimulus displaying period were used for the model. We did not constrain the signal to certain specific ERP component, such as P3 which is most used in BCI researches so far. Instead we used both P1 and P3 components, as well as other later signals after 400 ms from the stimulus onset.

In the experiments, subjects were asked to click a button corresponding to the brightness state during the observation in order to keep them alert to the stimuli. The average response time of the button pressings by the subject, whose data was shown as the examples in the paper, is 1216 ms and the shortest one is 741 ms. Thus in Fig. 2, only the ERPs before 800 ms are shown. The brain activities during this period were considered to be related with the brightness perception, but without motor action. For our classifier, the best classifications were also attained during this period. Hence our classifier succeeded in the classification based on pure brightness perception. In addition, a small number of trials were used to train the classifier, and the efficiency was high.

In our current experiments, we used the still images, and the EEG data were analyzed offline. An online BCI system responding to dynamic movies should be more exciting, and it is within the range of our future work.

5 Conclusions

In this paper, we created a classifier to discriminate the different brightness perception based on the brain activities of single trials measured by non-invasive EEG measurement. The AAR model were applied to extract the features of different brain states corresponding to stimuli in different brightness, and the minimum distance analysis (MDA) classifier was created to discriminate different brightness perception. We achieved acceptable accuracies, and the kappa coefficients showed that our model has a moderate agreement with the bright perception. This paper is an attempt on constructing a BCI for people with normal vision ability to interact with daily electronic device. The method proposed can classify different EEG activity corresponding to different visual stimuli regarding brightness of images displayed on TV screen. Further investigation is warranted to develop EEG-based stimuli-response classifiers for broader stimulus.

References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. *Clinical Neurophysiology* 113, 767–791 (2002)
2. Sajda, P., Muller, K.-R., Shenoy, K.V.: Brain-Computer Interfaces. *IEEE Signal Processing Magazine* 25, 16–17 (2008)
3. Santhanam, G., Ryu, S.I., Yu, B.M., Afshar, A., Shenoy, K.V.: A high-performance brain-computer interface. *Nature* 442, 195–198 (2006)

4. Middendorf, M., McMillan, G., Calhoun, G., Jones, K.S.: Brain-computer interface based on the steady-state visual-evoked response. *IEEE Transactions on Rehabilitation Engineering* 8, 211–214 (2000)
5. Farwell, L.A., Donchin, E.: Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology* 70, 510–523 (1988)
6. Gerson, A.D., Parra, L.C., Sajda, P.: Cortical origins of response time variability during rapid discrimination of visual objects. *NeuroImage* 28, 342–353 (2005)
7. Wolpaw, J.R., Ramoser, H., McFarland, D.J., Pfurtscheller, G.: EEG-Based Communication: Improved Accuracy by Response Verification. *IEEE Trans. on Rehab. Eng.* 6, 326–333 (1998)
8. Pfurtscheller, G., Neuper, C.: Motor imagery and direct brain-computer communications. *Proc. IEEE* 89, 1123–1134 (2001)
9. Blankertz, B., Dornhege, G., Krauledat, M., Müller, K., Curio, G.: The non-invasive Berlin Brain-Computer Interface: Fast acquisition of effective performance in untrained subjects. *NeuroImage* 37, 539–550 (2007)
10. Schlögl, A., Roberts, S.J., Pfurtscheller, G.: A criterion for adaptive autoregressive models. In: *Proceedings of the World Congress 2000 on Medical Physics and Biomedical Engineering*, Chicago, WE-A328-01 (2000)
11. Vidaurre, C., Schlögl, A., Cabeza, R., Scherer, R., Pfurtscheller, G.: A fully on-line adaptive BCI. *IEEE Transactions on Biomedical Engineering* 53, S1214–S1219 (2006)
12. Schrögl, A., Keinrath, C., Scherer, R., Pfurtscheller, G.: Information transfer of an EEG-based Brain-computer interface. In: *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, Capri, Italy, pp. 641–644 (2003)
13. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 37–46 (1960)

Steady State Visual Evoked Potentials in the Delta Range (0.5-5 Hz)

François-Benoit Vialatte¹, Monique Maurice¹, Justin Dauwels^{1,2},
and Andrzej Cichocki¹

¹ Riken BSI, Wako-Shi, Japan

fvialatte@brain.riken.jp

² Massachusetts Institute of Technology, Cambridge, MA

Abstract. The usually ‘accepted’ limits of Steady State Visual Evoked Potentials are in the 3-60 Hz range. Recent studies reported SSVEP activities below 3 Hz, which remains a matter of debate. We recorded EEG responses to stimuli from 0.5 to 13 Hz. We first confirm the possibility to elicit SSVEP below 3 Hz. Afterwards, for the first time, we show that SSVEP recorded in the δ (0.5-5 Hz) range seem to describe several subsystems, with peaks near 1, 2.5 and 5 Hz (close to subharmonics of 10 Hz). Finally, we report surprising results in the lower frequency ranges, with responses for almost all harmonics (*e.g.* 15 peak responses between 0.5 and 14.5 Hz for stimuli at 1 hz).

1 Introduction

Sensory evoked potentials are electrical potentials, recorded from the nervous system of human or animal, while stimulating sense organs. EEG evoked potentials (EP), as distinct from spontaneous potentials (background EEG), can be interpreted as the reorganization of the spontaneous brain oscillations in response to the stimulus [20]. Contrary to event-related potentials, these potentials are time-locked to the stimulus. Therefore they can be enhanced using averaging techniques on several trials [4]. It is well-known that following the presentation of a transient visual stimuli, an EEG response in the visual areas can be recorded. However, flash VEP and pattern VEP (reversal or offset/onset) are the most completely studied and described types of VEP (see [3] for review).

Such visual stimuli usually induce transient responses of the visual system. However, using long stimulus trains, a stable VEP is produced, which can be displayed using averaging techniques. These EEG waves are termed as “steady-state” visual evoked potentials (SSVEP) of the human visual system [5], and can also be observed with other electrophysiological technique (local field potentials, magnetoencephalography). Steady-state potentials are to be distinguished from transient potentials, because their constituent discrete frequency components remain closely constant in amplitude and phase over a long time period [6].

The usually ‘accepted’ limits of SSVEP are in the 3-60 Hz range. What happens below and above these frequencies is not well-documented. Recent studies reported SSVEP activities below 3 Hz (1-100 Hz EEG in [9], 0-20 Hz MEG in

[10]. Nevertheless, many recent publications still convey the notion that SSVEP can only be recorded above 3 or 4 Hz (*e.g.* above 6 Hz in [11]). The present study seeks to demonstrate the SSVEP phenomena in the low frequency range ($\delta = 0.5\text{--}5$ Hz), using EEG responses recorded during continuous flash VEP stimulation.

2 Methods

2.1 EEG Data – SSVEP in the δ Range

Five subjects without history of epileptic seizure or related troubles (males, age $\simeq 30 \pm 1.6$) gave their informed consent for this experiment. Flickering light can trigger photosensitive epilepsy. Before recording the EEG session, subjects received detailed information about photosensitive epilepsy. After EEG was set-up, subjects were first exposed to flickering light and tested for any pre-symptoms of epilepsy² (no subject were rejected).

EEG was recorded in a dark room with a G-Tec system cap, with a sampling rate of 512 Hz, notch filtering at 50 Hz, and online bandpass filter at 0.01 and 100 Hz. We compared SSVEP every 0.5 hz until 5 Hz with those recorded at 7, 10 and 13 Hz, frequencies which have been thoroughly studied in the literature. Hence, SSVEP were recorded with the stimulus frequencies $\nu_i \in \mathbf{V} = [0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 7, 10, 13]$ Hz. For each frequency, we recorded first a baseline period of 20 sec (rest with eyes opened). Next we alternated 20 times a 10 second period of stimulation³ with 5 second period of rest. Stimulation was performed with a flash VEP, obtained by blinking white square of 5 cm side (*i.e.* 128×128 pixels) at the center of a computer screen. Fixation task was performed with a non blinking gaussian noise of the same dimension. EEG was recorded with 20 channels placed according to the 10-20 international system, together with vertical and horizontal electrooculogram. Epochs with too strong artifacts were rejected⁴. Stimuli were recorded simultaneously in an additional channel (using a G-tec g.TRIGbox). The distance to the screen was controlled for all subject to be $\simeq 1$ m, in a dark room. We averaged the EEG at electrode Oz⁵ over all trials and all subjects (totaling 80-100 trials for each stimulus frequency, as some trials were rejected when noise was too high).

2.2 Fourier Power Analysis

The Fourier power Φ was computed using the Welch method (4 sec. Hanning windows with 50% overlap, whole 10 sec stimulation period) in bins of $s = 0.1$

¹ \pm indicates standard-error.

² During a first trial session pre-symptoms of epilepsy were observed (= from simple headache to EEG sharp waves) - any such symptoms would have lead to rejection of the subject.

³ Long duration is needed for low frequency stimulation, for instance at 0.5 Hz it takes 1 second to stimulate one cycle only.

⁴ And especially one subject fell asleep in the middle of one session.

⁵ Oz = Occipital area channel, where the source of SSVEP is usually detected in scalp EEG. Occipital channels are located above the visual cortex.

Hz for 0.1 to 15 Hz. The signal-to-noise ratio of SSVEP $\text{SNR}(f_p)$ was then computed using the ratio of Fourier power at a given frequency f_p to its n -adjacent frequencies power:

$$\text{SNR}(f_p) = \frac{n \cdot \Phi(f_p)}{\sum_{k=s}^{s+n/2} \Phi(f_p + k) - \sum_{k=s}^{s+n/2} \Phi(f_p - k)} \quad (1)$$

with n pair. This SNR measure have been used successfully to enhance SSVEP peaks [12]. We used $n = 6$, so that the frequencies $f_p \pm 0.4$ Hz were taken into account. In other words, for each stimulus frequency $\nu_i \in \mathbf{V}$, with $\mathbf{V} = [0.5, 1, \dots, 4.5, 5, 7, 10, 13]$ we obtained a Fourier SNR vector $\text{SNR}_i(f_p)$ with $f_p \in [0.5 - 14.6]$ Hz. Fourier power and SNR were computed for each trials, and averaged afterwards. Computations were done with Matlab (The MathWorks, Inc.). Results presented here come from the O2 electrode; scalp propagation will be presented in an extended version of this paper (due to limit of space).

3 Results

3.1 Is There Anything Below 5 Hz?

For each $\nu_i \in \mathbf{V}$, we reported on a curve the average Fourier power value $\overline{\Phi}_{i,s,t}(\nu_i)$ (grand average during stimulation at frequency ν_i , for all trials t and all subjects s , at the frequency ν_i). We compared these Fourier powers against the average Fourier power during the baseline period $\overline{\Phi}_{s,t}$ (average for all subjects, over all frequencies in [0.5–5] Hz). As can be seen on Fig. II for all stimulation frequencies, we observed an increased Fourier power as compared to the baseline. The stimulation induced changes in EEG power for all frequencies below 5 Hz.

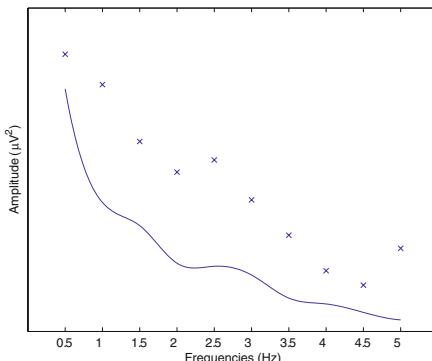


Fig. 1. Average Fourier power. Crosses: average Fourier power $\overline{\Phi}_{i,s,t}(\nu_i)$ during continuous flash VEP stimulation for each $\nu_i \in \mathbf{V}$, depending on the frequency of the stimulation. Lower curve: average power $\overline{\Phi}_{s,t}$ during rest with eyes opened.

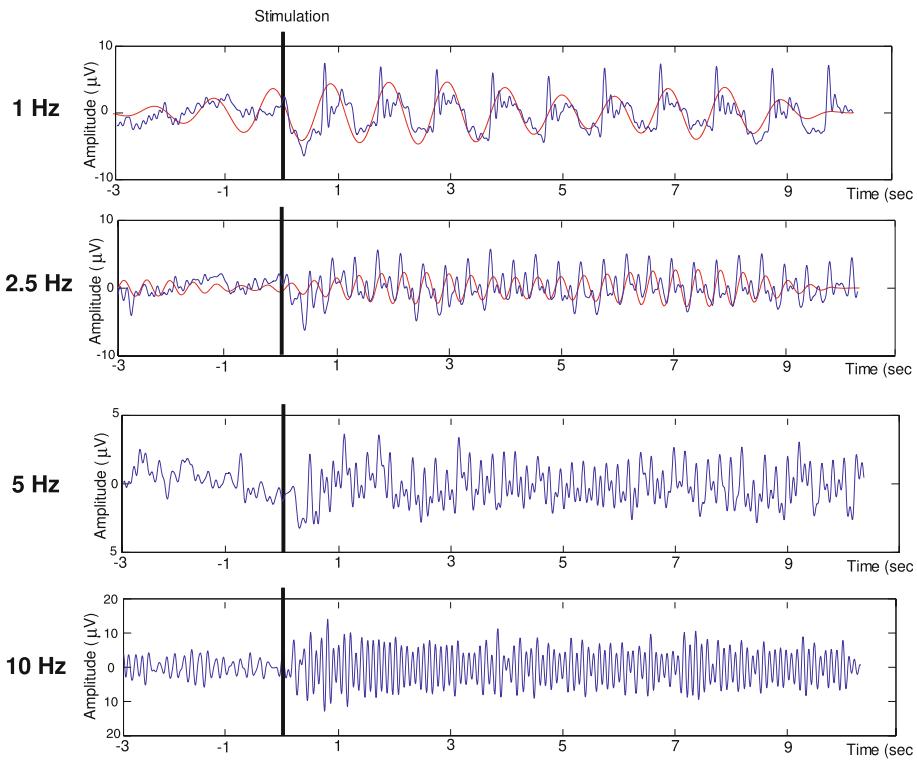


Fig. 2. Averaged SSVEP for the stimulus frequency $\nu_i = 1, 2.5, 5$ and 10 Hz . The curves are periodic, but not sinusoidal. For more convenience, a superimposed red curve representing the curve filtered around the SSVEP frequency $\pm 0.25\text{ Hz}$ is also displayed for the 1 Hz and 2.5 Hz SSVEP (representing the SSVEP without its harmonics).

3.2 Periodic But Not Sinusoidal

One would expect SSVEP to look like a sinusoidal curve. This is however incorrect: SSVEP are known to have strong amplitude at the stimulus frequency, but also at its harmonics. The Fig. 2 shows the SSVEP averages with the stimulus frequency $\nu_i = 1, 2.5, 5$ and 10 Hz (similar results are observed for all frequencies). The curves are periodic and nearly sinusoidal (blue curves), but they seem more complex for lower frequencies (see e.g. the curve at $\nu_i = 1\text{ Hz}$ on Fig. 2 vs. the curve at $\nu_i = 10\text{ Hz}$). This could be explained by two possibilities: either the several harmonics of 1 Hz are superimposed; or what we recorded is not SSVEP, but a succession of transient VEP.

3.3 SSVEP or Transient VEP?

The best solution to check for the presence of SSVEP against transient VEP is to use Fourier representation. It is because SSVEP are stable (not transient),

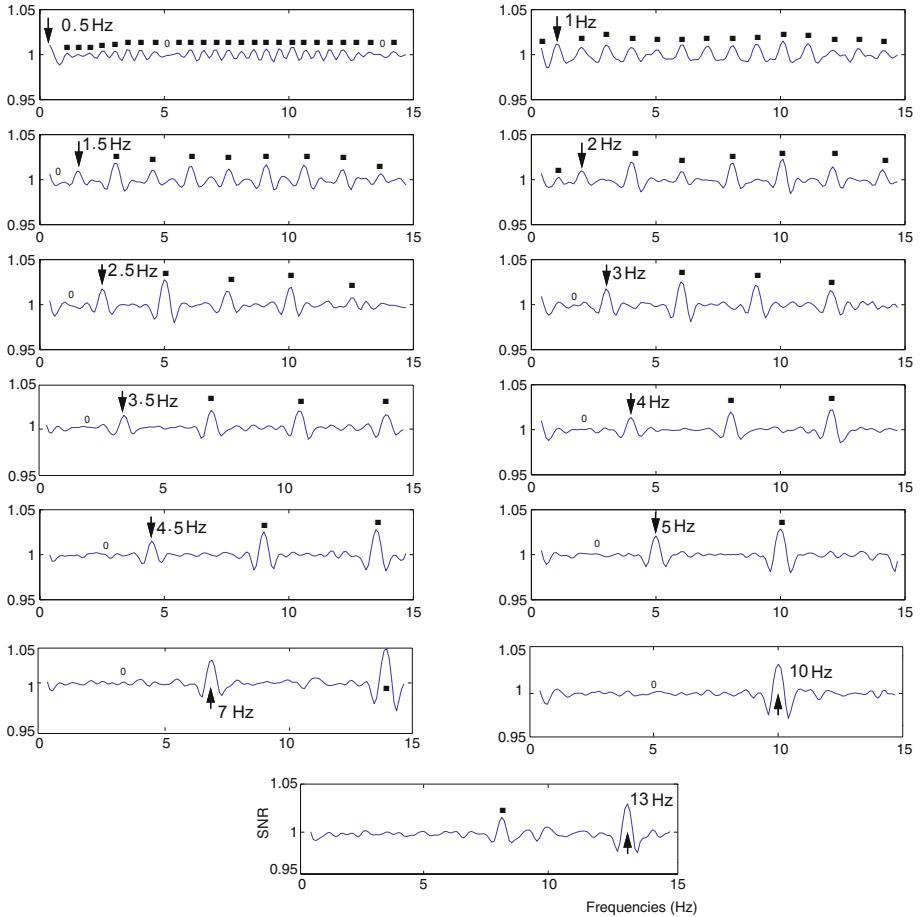


Fig. 3. Averaged Fourier SNR $\overline{SNR}_{i,s,t}$ for each frequencies. Responses are visible at the stimuli frequency, but also at their harmonics. Arrows represents the expected SSVEP frequency, dots the harmonics or subharmonics found, and zeros expected harmonics or subharmonics that were not observed reliably. The evoked activity below $\nu_i = 3$ Hz match the definition of SSVEP: they have nearly constant frequency components at the stimulus harmonics.

that their frequency components remain nearly constant, leading to a significant change of Fourier power at their peak frequency. Furthermore, SSVEP should display changes at harmonics of the stimulus frequency. We averaged the Fourier SNR computed for all subjects and all trials (80-100 trials per frequency, as explained above). The Fig. 3 shows the result $\overline{SNR}_{i,s,t}$ of the Fourier SNR computation for all frequencies during stimulation at frequency ν_i (grand average for all trials t and all subjects s). All stimuli induced a peak at the stimulus frequency and harmonics. The results obtained in the low ($\nu_i = 0.5 - 3$ Hz) δ range prove that SSVEP were elicited. Harmonic responses surprisingly did not

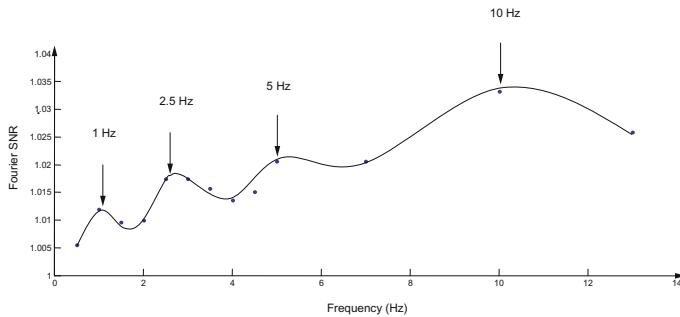


Fig. 4. SSVEP δ range subsystems. The dots represents the average Fourier SNR values $\overline{SNR}_{i,s,t}$ for each stimulus frequency ν_i , over each trial t and each subject s . The well-known peak of power at $\nu_i = 10$ Hz is observed, together with peaks at $\nu_i \simeq 1$, $\nu_i \simeq 2.5$, and $\nu_i = 5$ Hz.

disappear in higher frequency ranges, for instance the $\nu_i = 1$ Hz SSVEP has 15 peaks from 0.5 to 14.6 Hz, including the 0.5 subharmonic. In most cases however the subharmonics were weaker (they were clearly observed only for the $\nu_i = 1$, 2 and 13 Hz SSVEP).

3.4 SSVEP δ Subsystems

For each ν_i , we reported on a curve the average Fourier SNR value $\overline{SNR}_{i,s,t}$ (grand average at the frequency ν_i - i.e. the arrows on Fig 3). This shows the evolution of SSVEP amplitude depending on frequency. The curve (Fig 4) shows four peaks: at $\nu_i \simeq 1$, $\nu_i \simeq 2.5$, $\nu_i = 5$, and $\nu_i = 10$ Hz. All activities below 5 Hz have a lower SNR, which could explain why they often get unnoticed in the literature.

4 Discussion and Conclusion

We have shown that stimulation in frequency range as low as 0.5 Hz elicit periodic nearly sinusoidal evoked potentials (see Fig 2). Contrary to SSVEP in higher frequencies (> 5 Hz), the δ range SSVEP are less sinusoidal. However, in all frequencies, the same phenomenon was observed: an increase of amplitude at the SSVEP frequency during stimulation (see Fig 1), with peaks for all harmonics of the SSVEP frequency (see Fig 3). The distortions observed in lower frequencies are hence attributed to the contribution of their several harmonics and to the poor signal-to-noise ratio of signals in very low frequencies.

Despite years of investigation, SSVEP remains a complex mechanism, which is not yet completely understood. It is usually decomposed in three different subsystems, which can be considered as three functional flicker components [7,6,8]:

1. A primary component dominating in high frequency (gamma 25-60 Hz range).
2. A secondary component dominating in the 15-25 Hz range.
3. A rhythmic after-discharge dominating in low frequency (5-15 Hz range).

SSVEP recorded in the δ ($\nu_i \in [0.5 - 5]$ Hz) range seem to describe several subsystems, with peaks near $\nu_i = 1, 2.5$ and 5 Hz. To the best of our knowledge, these new components were never described. These peaks are, interestingly, close to the 10 Hz subharmonics: $1.25, 2.5$ and 5 Hz (we did not unfortunately record SSVEP at 1.25 Hz). This fits with the theory developed in [9]: SSVEP induced by a flickering light without pattern have been shown to elicit better responses for harmonics of 10 Hz ($10, 20, 40$ and even 80 Hz). Our study thus confirms the importance of the 10 Hz rhythm in SSVEP. We could expect to find other peaks at lower frequencies (*e.g.* 0.75 Hz or 0.375 Hz), but the limited precision of recording devices, and the presence of artifacts in the δ range, would make it very difficult to measure them.

Finally, a crucial question remains: what effect could we expect from stimulations in low frequency ranges? What could be the outcome for the brain of the surprising EEG changes observed (*e.g.* every 0.5 Hz for SSVEP at $\nu_i = 0.5$ Hz, see Fig. 3)? One of our subject for instance consistently fell asleep after a few trials during flickering stimulation below 2 Hz. The effects (especially cognitive effects) for very low frequencies (below 3 Hz) are certainly worth some complementary investigations.

Acknowledgments

Monique Maurice is supported by the JSPS fellowship P-08811. Justin Dauwels is supported by the Henri-Benedictus fellowship from the King Baudouin Foundation and the BAEF.

References

1. Başar, E., Demirrap, T., Schürmann, M., Başar-Eroglu, C., Ademoglu, A.: Oscillatory brain dynamics, wavelet analysis, and cognition. *Brain and Language* 66, 146–183 (1999)
2. Başar, E.: EEG-brain dynamics: Relation between EEG and brain evoked potentials. Elsevier, Amsterdam (1980)
3. Odom, J.V., Bach, M., Barber, C., Brigell, M., Marmor, M.F., Tormene, A.P., Holder, G.E.: Vaegan.: Visual evoked potentials standard. *Doc. Ophthalmol.* 108(2), 115–123 (2004)
4. Dawson, G.D.: A summation technique for the detection of small evoked potentials. *Electroenceph. Clin. Neurophysiol.* 6, 65–84 (1954)
5. Regan, D.: Some characteristics of average steady-state and transient responses evoked by modulated light. *Electroenceph. Clin. Neurophysiol.* 20, 238–248 (1966)
6. Regan, D.: Human brain electrophysiology: evoked potentials and evoked magnetic fields in science and medicine. Elsevier, New York (1989)
7. Spekreijse, H., Estevez, O., Reitz, D.: Visual evoked potential and the physiological analysis of visual process in man. In: Desmedt, J.E. (ed.) *Visual evoked potentials in man: new developments*, pp. 16–85. Clarendon, Oxford (1977)
8. Silberstein, R.B.: Steady-state visually evoked potentials, brain resonances, and cognitive processes. In: Nunez, P.L. (ed.) *Neocortical Dynamics and Human EEG Rhythms*, pp. 272–303. Oxford University Press, Oxford (1995)

9. Herrmann, C.S.: Human EEG responses to 1–100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Exp. Brain Res.* 137(3-4), 346–353 (2001)
10. Fawcett, I.P., Barnes, G.R., Hillebrand, A., Singh, K.D.: The temporal frequency tuning of human visual cortex investigated using synthetic aperture magnetometry. *Neuroimage* 21(4), 1542–1553 (2004)
11. Lin, Z., Zhang, C., Wu, W., Gao, X.: Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs. *IEEE Trans. Biomed. Eng.* 53(12pt2), 2610–2614 (2006)
12. Wang, Y., Wang, R., Gao, X., Hong, B., Gao, S.: A practical VEP-based brain-computer interface. *IEEE Trans. Neural Syst. Rehabil. Eng.* 14(2), 234–239 (2006)

Using Optimality to Predict Photoreceptor Distribution in the Retina

Travis Monk¹ and Chris Harris²

¹ Department of Computer Science, University of Otago,
Dunedin, New Zealand
`montr316@student.otago.ac.nz`

² Centre for Theoretical and Computational Neuroscience,
University of Plymouth, Plymouth, Devon, PL4 8AA, UK

Abstract. The concept of evolution implies that fitness traits of an organism tend toward some constrained optimality. Here, the fitness trait we consider is the distribution of photoreceptors on an organism’s retina. We postulate that an organism’s photoreceptor distribution optimizes some balance between two quantities, a benefit and a cost. The benefit is defined as the area of the field of vision. The cost is defined as the amount of time spent saccading to some target in the visual field; during this time we assume nothing is seen. Three constraints are identified. First, we assume proportional noise exists in the motor command. Second, we assume saccades are a noisy process. Third, we constrain the number of total photoreceptors. This simplified model fails to predict the human retinal photoreceptor distribution in full detail. Encouragingly, the photoreceptor distribution it predicts gets us closer to that goal. We discuss possible reasons for its current failure, and we suggest future research directions.

1 Introduction

Optimality is an enormously powerful and fundamental concept claiming its own branch of physics, engineering, and even economics. It is surprising, then, that one finds relatively few applications of calculus of variations or optimal control theory to biology. This lack of biological applications is even more surprising when one considers that evolution, the fundamental theory of biology, implies optimality of fitness. Fitness refers to innumerable properties (perhaps *every* property) of an organism, from the tapering of bones in animals’ arms and legs to the trajectories of arm movements to photoreceptor distribution on the retina. We can expect to see evolution tending toward optimality, where optimality is gauged in fitness [1]. We should see this concept of optimality cropping up again and again in biological studies to explain behaviours or anatomical features at a fundamental level. Here, we attempt to invoke optimality to derive one particular anatomical feature: the distribution of photoreceptors in the retina.

2 Methods

Due to the computational complexity of the problem at hand, we make some plausible assumptions. First, we assume the retina is a semicircle of radius a ¹. Second, we

¹ Extending the retina to a second dimension is a possible future project.

assume ‘theta-symmetry.’² By stating ‘theta-symmetry’ we mean two things: objects have equal probability of falling in the left or right halves of the retina; and saccades to the left are equivalent to saccades to the right. This forces the performance index to be symmetric about $\theta = 0$. The optimal solution to the performance index will then also be symmetric about $\theta = 0$. Invoking theta-symmetry saves computation time by cutting the number of computations in half.

Our goal is to find the photoreceptor density function $\rho(\theta)$ (units of photoreceptors per unit angle) which minimizes some performance index but also satisfies the constraint:

$$\int_0^{\frac{\pi}{2}} \rho(\theta) d\theta = n \quad (1)$$

where n is the number of photoreceptors in half of the (two-dimensional) retina. Notice the limits of integration; we only consider half of the retina because we assume theta-symmetry.

Our performance index is defined as:

$$Index = \beta(Cost) - \alpha(Benefit) + \lambda \int_0^{\frac{\pi}{2}} \rho(\theta) d\theta \quad (2)$$

We now define our performance index in detail. The performance index is too complicated to solve analytically, so we find an approximate solution numerically. First, if we assume we know the form of photoreceptor density function $\rho(\theta)$, we can solve for the Lagrange multiplier λ explicitly. Assume $\rho(\theta)$ is defined as:

$$\rho(\theta) = \frac{1}{x_1\theta^3 + x_2\theta^2 + x_3\theta + x_4} \quad (3)$$

where θ ranges from 0 to 90 degrees, assuming theta-symmetry. $x = [x_1, x_2, x_3, x_4]$ are scalar variables and we are to find the x that minimizes equation 2. The specific form of equation 3 was chosen because, for a given x , it yields reasonable photoreceptor distributions. We can now solve for λ . Let k be defined as follows:

$$k = \int_0^{\frac{\pi}{2}} \rho(\theta) d\theta \quad (4)$$

Multiplying both sides by n/k and comparing this with equation 1, we see that we must normalize $\rho(\theta)$ by multiplying it by the factor n/k . In other words, we force $\rho(\theta)$ to obey our constraint (equation 1) if we define λ as:

$$\lambda = \frac{n}{k} \quad (5)$$

where k is defined in equation 4.

² Theta represents retinal eccentricity.

Next, we clearly define *Cost* and *Benefit* from equation 2. *Cost* and *Benefit* are easiest to define when thinking in terms of angular resolution. Consider some normalized photoreceptor density function $\rho(\theta)$. We can define the angular resolution of $\rho(\theta)$ as:

$$ar(\theta) = \frac{1}{\rho(\theta)} \quad (6)$$

$ar(\theta)$ yields the distance between adjacent photoreceptors and thus the resolvability of the photoreceptor density function at a given retinal eccentricity θ . Now consider a target of size s at position (r, θ) where r is the distance of the target from the node of the retina. The angular resolution of the target is then s/r .

We define *Benefit* as the area that the photoreceptor density function can cover:

$$Benefit = \int_0^{\frac{\pi}{2}} \int_{r_{\min}}^{r_{\max}} P(r, \theta) r dr d\theta \quad (7)$$

r_{\max} is defined as the maximum distance from the node of the retina that the target can exist and still be resolved. This occurs when the target falls exactly on the fovea, or when the target's angular resolution equals the minimum of $ar(\theta)$.

$$r_{\max} = \frac{s}{\min(ar(\theta))} \quad (8)$$

r_{\min} represents the largest angular resolution of a target considered:

$$r_{\min} = \frac{s}{\left(\frac{\pi}{2}\right)} \quad (9)$$

$P(r, \theta)$ represents the probability of resolving a target at some location (r, θ) . We postulate that repeating the same saccade of *desired* amplitude A will yield a normal distribution of saccadic amplitudes of mean A and standard deviation σ . σ represents the noise associated with saccades and will be discussed in more detail shortly. To clarify the concept of $P(r, \theta)$, consider a photoreceptor distribution as denoted by the solid blue line in figure 1. For targets located at large values of r , denoted by the x below, saccades must be more accurate to resolve the target. Since saccades are noisy, this is improbable and the photoreceptor distribution has a low benefit. At smaller values of r , as denoted by the circle below, the organism has a greater probability of resolving the target after a saccade because the resolvable region is wider.

We can see that $P(r, \theta)$ depends on two things: the width of the resolvable region at some distance r , and the noise of the saccades σ . Specifically:

$$P(r, \theta) = \text{erf}\left(\frac{x}{2\sqrt{2}\sigma}\right) \quad (10)$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (11)$$

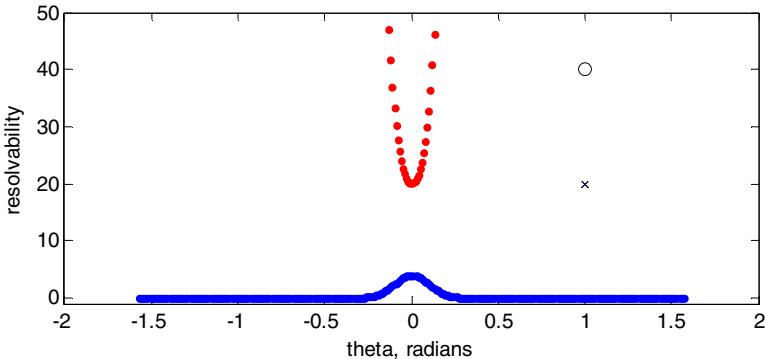


Fig. 1. Exploring the concept of $P(r, \theta)$. It is difficult to saccade the x inside the dotted region, thus $P(r, \theta)$ is low. It is easier to saccade the o inside the dotted region since it is wider, thus $P(r, \theta)$ is high.

where x represents the width of the resolvable region for a target of size s at retinal distance r .

We assume three different types of noise: noise due to targeting error, dynamic noise from making a saccade, and a fixational noise after a saccade. We combine these three noises, remembering to add their variances since variances are orthogonal:

$$\sigma = \sqrt{\sigma_{target}^2 + \sigma_{dynamic}^2 + \sigma_{fix}^2} \quad (12)$$

$$\sigma_{target} = \frac{ar(\theta)}{\sqrt{12}} \quad (13)$$

$$\sigma_{dynamic} = .1 * \sqrt{|\theta_{max} - \theta|} \quad (14)$$

$$\sigma_{fix} = \frac{1}{6} \quad (15)$$

Our expression for *Benefit* (equation 8) is now well-defined.

Cost is defined as follows:

$$Cost = \int_0^{\frac{\pi}{2}} \int_{r_{min}}^{r_{max}} f(A) r dr d\theta \quad (16)$$

where A represents an amplitude of movement and $f(A)$ represents the main sequence of saccades. Consider a target at location (r, θ) . We approximate the time taken to saccade the target at location (r, θ) from its initial location to the position of the highest photoreceptor distribution, θ_{max} . Knowing the amplitude of the desired saccade, $A = |\theta_{max} - \theta|$, we insert this amplitude A into the main sequence to obtain the duration of the saccade, $f(A)$. This time cost integrand is then integrated over the entire resolvable region as defined in equation 16 above. Equations 17 and 18 are the piecewise function we use to approximate the main sequence:

$$f(A) = \frac{50}{2.5119} A^4, A \leq 10 \quad (17)$$

$$f(A) = 2.6A + 24, A > 10 \quad (18)$$

where A is the amplitude of the saccade measured in degrees. Saccade duration $f(A)$ has units of milliseconds here.

α and β are scalar weights that represent the relative importance of *Cost* and *Benefit*. We postulate that different values of α and β will explain different photoreceptor distributions for different species. For instance, consider a low α and high β . This would correspond to an organism falling low on the food chain, perhaps a rabbit. A rabbit will have a uniform photoreceptor distribution so it never has to saccade. The few milliseconds that the rabbit spends saccading around the environment could truly cost the organism its life. Consider now a high value of α and a low value of β . This would correspond to an organism falling high on the food chain, perhaps a primate. An organism that does not face many predators can afford to cluster its photoreceptors in one location, sit contently, and saccade to explore its environment.

Our task is now to find the photoreceptor distribution that minimizes this performance index, which is now well-defined. Specifically, our job is to find input $x = [x_1, x_2, x_3]$ where x defines a photoreceptor distribution by equation 3. Note that x_4 is not a degree of freedom due to normalization.

Our cost space is incredibly complicated and produces several minima. Thus, we decided the best course of action was one of brute force. We consider every combination $x = [x_1, x_2, x_3]$ where $-5 \leq x_i \leq 11$ in increments of .25. The limits of -5 and 11 were arrived at by experimentation. For each polynomial, we calculate a corresponding performance index given by equation 2. To calculate this massive polynomial crunch, we divided our subspace of possible solutions $-5 \leq x_i \leq 11$ into separate sections. The minimal photoreceptor distribution of each section was then computed in parallel. We subsequently compared the minima of each section to find the minimum in the entire subspace of possible answers. Finally, we used this minimum as an initial input for an optimizing algorithm in Matlab called ‘fminsearch.’

This polynomial crunch was completed for a range of values of the coefficient ratio α/β from 0 to 100000. We selected such a wide range of values to get an idea for how it affects the optimal photoreceptor distribution. Based on these results, we ‘honed in’ on a more specific region of the coefficient ratio that produced photoreceptor distributions that closely matched photoreceptor distributions in humans. It is the results from this specific region of α/β that we report here.

3 Results

We made three predictions about how the optimal photoreceptor distribution should change with varying values of α/β . First, for $\alpha/\beta = 0$, we should achieve a

normalized uniform distribution to minimize the time cost of saccading, since *Benefit* is defined to be worthless. Second, for increasing values of α/β , we expect to see more and more photoreceptors piled in the centre of the retina. By grouping photoreceptors, an organism can cover more area with its retina. See equations 7 and 8. Third, as α/β gets very large, the optimal photoreceptor distribution should begin to resemble a Dirac-delta function. The first and last of these three predictions was confirmed, though we omit proof here due to space constraints. The second prediction is confirmed by figure 2 below.

The black trace below, the trace with the lowest amplitude, closely resembles photoreceptor distribution in humans at larger eccentricities. At an eccentricity of 10 degrees and greater, the black trace below is very close to physiological data. However, the distribution does not fall off at the origin as fast as the photoreceptor distribution in humans. In humans, the photoreceptor density falls to about 60% of its peak value at an eccentricity of only one degree. Figure 3 is a blow-up of this black (lowest amplitude) trace about the origin. We can see that the photoreceptor density does not drop to 60% of its value at the origin until the eccentricity is around three degrees.

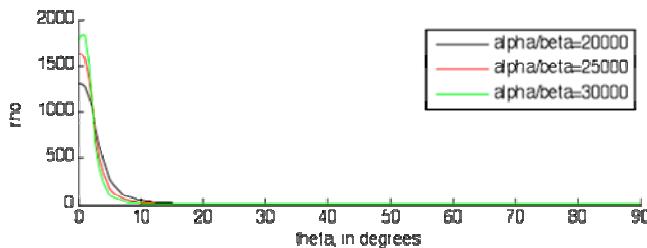


Fig. 2. Photoreceptor densities for three key values of alpha and beta

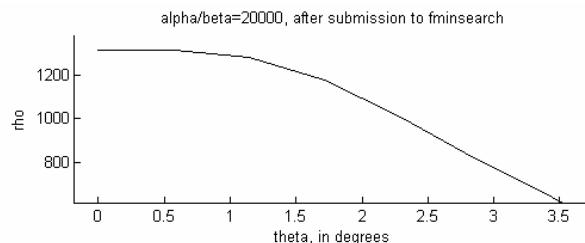


Fig. 3. A close-up of the black trace in figure 2. The photoreceptor distribution does not decline fast enough at the origin.

4 Discussion

Appealing features of our model include its attempt to unify photoreceptor distribution and eye movements, its avoidance of complex mathematical descriptions

of biology, and its reasonable results. Limitations and assumptions include our rudimentary definition of $f(A)$ (our description of the main sequence of saccades), our definition of $\rho(\theta)$, and our definition of *Cost*.

We conclude this brief section with a suggestive list of future research directions. The most glaring assumption in our work is our definition of the main sequence, $f(A)$. First, there is evidence for the existence of a saccadic ‘dead-zone’ where, for small amplitudes of movement A , we slowly drift the eye to the target instead of making a saccadic movement. This ‘dead-zone’ is not included in our definition of $f(A)$. Second, we incorporated the existence of large saccades in the model. However, humans rarely make saccades of such a large amplitude and we should try omitting their existence entirely in future work. Third, we should attempt an alternate definition of equation 3. Equation 3 is, in fact, an implicit constraint. Also keep in mind that our current equation 3 creates a very complicated cost space. Perhaps a different definition of $\rho(\theta)$ will create a cost space that is simpler to analyze and minimize. Fourth, and finally, we do not consider that saccades of some amplitudes are more frequent than saccades of other amplitudes during visual scanning [2]. This could be a significant effect that we neglect entirely. In addition, we could try weighting the importance of saccades depending on their frequency of occurrence.

5 Conclusion

Evolution implies that fitness traits of an organism tend toward some optimality. Here, we attempt to describe the photoreceptor distribution on the human retina as a distribution that optimizes a common-sense cost function. Our cost function is the weighted sum between a *Benefit* and a *Cost*. We define the *Benefit* of a photoreceptor distribution as the amount of area it can cover. We define the *Cost* of that distribution to be the time spent saccading to a target in the resolvable field. The optimal photoreceptor distribution finds an appropriate balance between *Benefit* and *Cost*, depending on the species under consideration. Three constraints are identified. First, we assume the eye saccades according to the main sequence of saccades. Second, we assume saccades are a noisy process. Third, we constrain the number of total photoreceptors.

The optimal photoreceptor distribution produced by our model does not reproduce in detail the human photoreceptor distribution for any balance between *Benefit* and *Cost*. The photoreceptor distribution does not decline quickly enough about the origin to fit experimental data. Also, our performance index is a very unpleasant function of our input distribution, making it difficult to analyze and minimize. Results do show some positive aspects. The fovea is predicted to be in the centre, photoreceptor density falls off reasonably with eccentricity, and the photoreceptor density relates predictably with the balance between *Benefit* and *Cost*. Our results did not fully meet our research goals. However, the information we garnered throughout the process brings us new insight into the problem. This new insight is illustrated by the several future research ideas we discussed.

References

1. Harris, C.M.: On the optimal control of behaviour: a stochastic perspective. *Journal of Neuroscience Methods* 83, 73–88 (1998)
2. Bahill, A.T., Adler, D., Stark, L.: Most naturally occurring human saccades have magnitudes of 15 degrees or less. *Invest. Ophthalmol. Vis. Sci.* 14(6), 468 (1975)

Optical Imaging of Plastic Changes Induced by Fear Conditioning in the Auditory Cortex of Guinea Pig

Yoshinori Ide¹, Johan Lauwereyns², and Minoru Tsukada¹

¹Tamagawa University Brain Science Institute, 6-1-1 Tamagawa-gakuen Machida,
Tokyo, 194-8610, Japan

²School of Psychology, Victoria University of Wellington, Wellington, New Zealand
yide@lab.tamagawa.ac.jp

Abstract. In this study, the plastic change in the auditory cortex induced by fear conditioning with pairing of sound (Conditioned Stimulus, CS) and electric foot-shock (Unconditioned Stimulus, US) was investigated by using of an optical recording. To investigate the effect of associated learning, optical signals in the auditory cortex to CS (12 kHz pure tone) and non-CS (8 kHz and 16 kHz pure tone) were recorded before and after the conditioning. As a result, the response area only to CS increased after the conditioning. On the other hand, to investigate whether auditory information could be retrieved by electric foot-shock after associated learning or not, auditory response to foot-shock alone was also investigated. As a result, the optical response in the auditory cortex to electric foot-shock alone could not be observed before the conditioning but clearly appeared after the conditioning.

Keywords: Auditory Cortex, Optical Imaging, Fear Conditioning, Plasticity.

1 Introduction

Fear conditioning using sound and electric stimuli is commonly used in the researches of emotional memory, many researches of plasticity in the auditory cortex using this paradigm have been carried out. Weinberger et al. reported that receptive field in the auditory cortex are plastically changed by fear conditioning and the best frequency strongly tunes to the frequency used for the conditioning [1]. The information in the auditory cortex is spatially represented as the tonotopic map. Merzenich et al. reported that the response region to the training frequency in the primary auditory cortex enlarged after the frequency discrimination task [2]. In this work, to verify the results for those researches, response in the auditory cortex to CS sound and non-CS sound was investigated before and after the conditioning. Furthermore, to investigate whether auditory information could be retrieved by electric foot-shock after associated learning or not, auditory response to foot-shock was also investigated before and after the conditioning.

2 Methods

Preparation. The experiments were performed in accordance with the guidelines of the Animal Experiments Committee of Tamagawa University. 22 guinea pigs of

250-450 g, 3-6 weeks old females were used as experimental subjects. The body temperature of an animal was maintained at 37°C by using the blanket system for animals (MK-900, Muromachi Kikai Co., LTD.) during the experiment. Each animal was anesthetized with a mixture of Ketamine (40 mg/kg, i.m.) and Xylazine (20 mg/kg, i.m.), and then anesthesia was maintained by supplementary doses of a mixture of Ketamine (20 mg/kg, i.m.) and Xylazine (10 mg/kg, i.m.). The trachea was cannulated and the head was cramped. The scalp was detached and a hole (approximately 10 by 10 millimeter square) was drilled in the left temporal bone and the dura and arachnoid membrane were removed. The auditory cortex was stained for 60-90 minutes with a voltage-sensitive dye, RH795 (0.125 mg/ml, dissolved in saline; molecular Probes). Then, the animal was artificially respiration after inducing paralysis with pancuronium bromide (0.2 mg/kg, i.m.). The experiments were carried out in a dark soundproof room. After the end of the experiment, each animal was administrated by overdose of Nembutal and a cardiac arrest was monitored.

Conditioning. In this work, fear conditioning was carried out by using sound stimulus (CS) and electric foot-shock (US). Three groups (Conditioning group, Pseudo conditioning group and Naive group) were prepared. Optical recording was carried out after the conditioning or pseudo-conditioning. In the conditioning group, six awake guinea pigs were fear-conditioned in the test cage with a grid shock floor. The conditioning protocol is shown in Fig.1. In the conditioning, CS (duration: 5 s, frequency: 12 kHz) was immediately followed by US (duration: 0.5 s, current intensity: 1.0-1.5 mA), and 30 trials were carried out in total. The interval between each trial was randomly 2-6 min.

In the pseudo-conditioning group, six awake guinea pigs were pseudo-conditioned in the same test cage. The pseudo-conditioning group was prepared as a control group. The pseudo-conditioning protocol is shown in Fig.1. In the pseudo-conditioning, in order to prevent conditioning, CS and US were separately presented in CS phase and US phase, respectively. The interval between each stimulus was randomly from 1 to 3 minutes, CS and US were presented 30 times, respectively. In order to confirm CR (the conditioned response), ECG (Electrocardiogram) was measured before and after the conditioning and pseudo-conditioning.

In the naive group, seven guinea pigs were prepared as a control. CS and US were never presented before the optical recording.

The waveform of sound stimuli was generated by MATLAB (Version 6.5, Mathworks). The waveform data generated by MATLAB was output as analog signal through the AD converter board (PCI-3337, Interface) installed in the personal computer. This signal was output from high-frequency speaker (ED1, ES1: 1041, Tucker-Davis Technologies) through the attenuator (PA5, Tucker-Davis Technologies). During the optical recording, sound stimulus was presented to the right ear of guinea pig. The electrodes for foot-shock were equipped with the hind legs using conductive paste. The electric shock (pulse width: 5 ms, frequency: 60 Hz, duration: 0.5 s) was generated by using stimulator (SEN-7203, Nihon Kohden), was applied to hind legs through the isolator (SS-202J, Nihon Kohden). The current intensity was adjusted within the range of 1.0-2.0 mA. The animal condition was monitored by electrocardiogram during the optical recording.

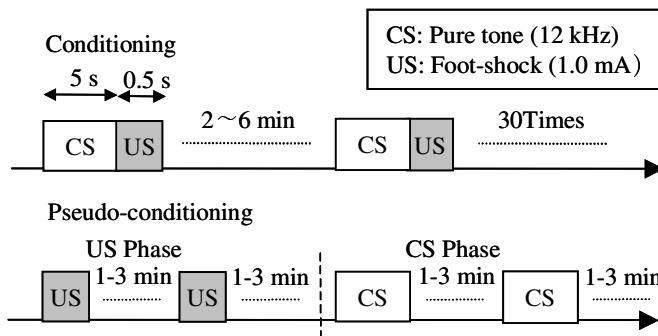


Fig. 1. Conditioning protocol and pseudo-conditioning protocol

3 Results

Auditory responses to pure tone with a frequency of 8 kHz, 12 kHz and 16 kHz were recorded before and after the conditioning. As shown in Fig. 2, the auditory response area markedly increased only for CS sound (12 kHz) after the conditioning, while such an increase of auditory response area could not be observed for the other non-CS sounds.

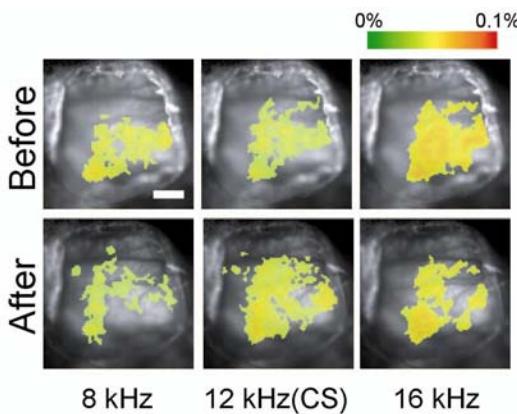


Fig. 2. Auditory responses to CS (12 kHz) and non-CS (8, 16 kHz) recorded before and after conditioning

Auditory responses to electric foot-shock alone were investigated without presenting any sound stimuli. Figure 3 shows auditory response to foot-shock alone in the three groups of conditioning, pseudo-conditioning and naive. Auditory response was observed in both conditioning and pseudo-conditioning groups and no response was observed in the naive group.

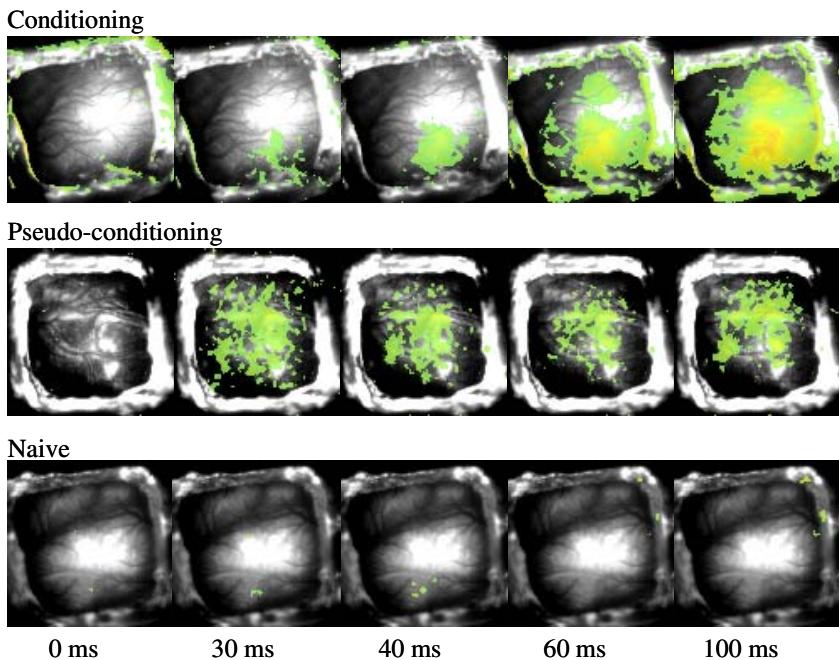


Fig. 3. Auditory responses to electric foot-shock alone in the three groups of conditioning, pseudo-conditioning and naive

Figure 4 shows the time course of the cortical response in fig.3. No response was observed in the Naive group, while response was observed in both conditioning and pseudo-conditioning group. EEG data shows that pseudo-conditioning group were also slightly conditioned, therefore not only conditioning group but also pseudo-conditioning group had a cortical response to foot-shock. The peak level of the conditioning group was almost twice as large as that of pseudo-conditioning group, implying that the magnitude of cortical response to foot-shock indicates the degree of associated learning. Besides, in a normal time course of tone response (gray solid line), the peak onset latency was approximately 40 ms and the offset latency was approximately 100 ms, however, in the time course of auditory response to electric foot-shock alone (black solid line), the onset latency was similar to that for a pure tone, while the rising until the peak was very gentle and the duration was over 450 ms, suggesting that the time course for electric foot-shock was very different from that of tone response.

Next, the area of auditory response to CS sound and electric foot-shock was compared. First, in order to investigate the tonotopic map in auditory cortex of the guinea pig, neural activity in the auditory cortex for each pure tone with a frequency of 4-16 kHz was observed by using of optical imaging (Fig.5). It has already been investigated by using of microelectrode method that field A and field DC that has tonotopicity exist in the rostral and caudal side of auditory cortex of guinea pig, respectively. In the field A, cells that respond to a sound are arranged from rostral to

caudal direction with increasing the sound frequency, to the contrary, in the field DC, cells are arranged from caudal to rostral direction with increasing the sound frequency. Figure 5 indicates such a mirror image symmetrical tonotopic map in the fields A and DC, and these results correspond to past preceding results about tonotopic map recorded by using of microelectrode method and optical imaging method.

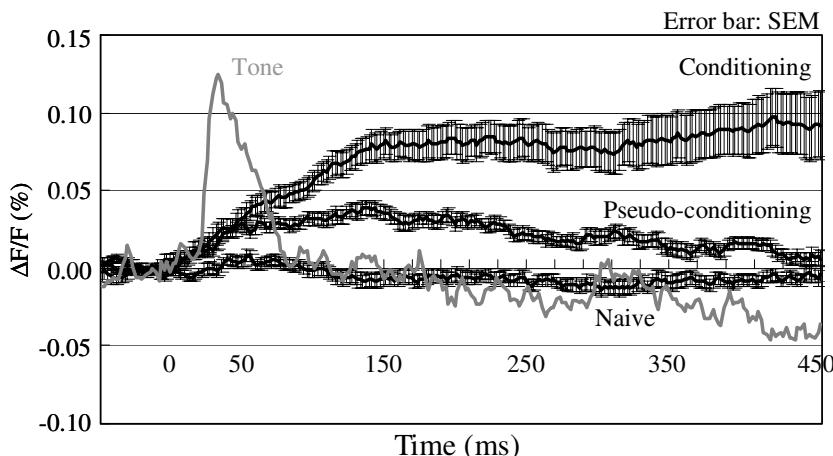


Fig. 4. Time course of auditory response to foot-shock alone in the three groups of conditioning, pseudo-conditioning and naive (Conditioning: n=11, Pseudo-conditioning: n=18, Naive: n=14)

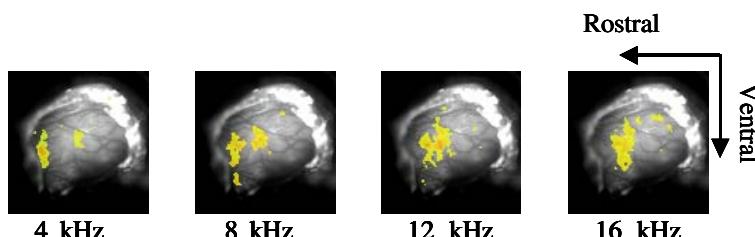


Fig. 5. Tonotopic map in the auditory cortex of guinea pig

Figure 6 shows the auditory responses to CS sound and electric foot-shock when the CS sound was 12 kHz pure tone. As shown in Fig.6, when the conditioning with a CS sound of 12 kHz was carried out, the auditory response area to the electric foot-shock considerably corresponded to that to CS sound. Furthermore, comparing the image at a response time of 14 ms with the tonotopic map for 12 kHz in Fig.5, the area of auditory response to electric foot-shock shifted a little bit to ventral direction, however, a comparative correspondence between their shapes of response area could be observed. Similar results were also obtained when CS sound was 4 kHz.

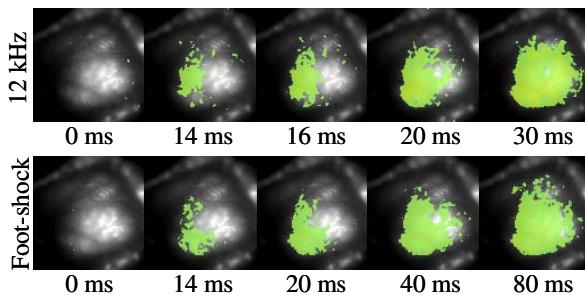


Fig. 6. Auditory responses to CS sound and electric foot-shock when CS was 12 kHz pure tone

4 Discussions

It has been reported that an optical imaging may record responses mainly from the layer II and III in the cortex. Furthermore, it has been reported from the pharmacological experimental data *in vivo* that responses for acoustic stimuli recorded by using optical imaging consists of the three main components as follows: (i) non-NMDA (*N*-methyl-D-aspartate) glutamate receptor (NMDAR) dependent excitatory postsynaptic potential (EPSP), which is a fast response component, (ii) NMDA glutamate receptor dependent EPSP, which is a slow response component and (iii) GABA (γ -amino butyric acid) receptor dependent inhibitory postsynaptic potential (IPSP) [3]. The characteristic of NMDAR dependent EPSP is slow response: the rising velocity is slow and a large response is observed in a late time region of 50 ms later than stimulus onset. The rising velocity of auditory response shown in Fig.4 was slow and a large response was obtained in a late time region of after 50 ms, suggesting that NMDAR dependent late-EPSP might be a dominant component of the auditory response to foot-shock. However, it is difficult to explain the long duration (approximately 500 ms) of the auditory response to foot-shock only by NMDAR.

It has also been reported that plastic changes in the auditory cortex is induced by Acetylcholine (ACh) released from basal forebrain to cortex during the conditioning [4], moreover, ACh affects NMDARs and promotes plastic changes in the auditory cortex [5]. Weinberger's model explains above hypothesis [6]. In his model, when the conditioning with pairing of CS sound and US electric shock is carried out, these information are transferred to amygdala, resulting that ACh is released to the auditory cortex from the basal forebrain. It is considered that this ACh affects NMDA glutamate receptors in layer II/III of the auditory cortex and the plastic changes in the auditory cortex is induced.

On the other hand, it is reported that ACh neuron is weakest neuron in Alzheimer's disease. Sarin is quite strong agonist of ACh, and many victims of the sarin incident in Japanese subway had experience that distant past memories already forgotten were vividly retrieved one after another although they didn't attempt to remember. Therefore, ACh might be deeply involved in both consolidation and retrieval of the memory. Furthermore, it is also reported that ACh release induced by stimulation of basal forebrain produces cortical desynchronization whose duration is as long as 6 seconds [7] and ACh contributes to attentional modulation in V1 for 200-500 ms after

stimulus onset [8], suggesting that ACh released from basal forebrain to cortex is involved in the long duration of the auditory response to foot-shock.

From these viewpoints, following mechanism might be thought. First, CS (tone) information is directly transferred to auditory cortex by way of MGv in thalamus, resulting that NMDARs in layer II/III produces EPSP (normal tone response). Next, US (foot-shock) information is transferred to amygdala by way of MGm in thalamus, and ACh is released from basal forebrain to the auditory cortex. As a result, LTP occurs in NMDARs where EPSP was induced in response to CS. Therefore, after the conditioning, when US alone is presented, ACh released from basal forebrain induces EPSP in the same NMDARs where LTP was induced by conditioning. Thus there is a correlation between auditory response area to CS sound and electric foot-shock, implying that the acoustic information about CS sound is retrieved by electric foot-shock.

References

1. Bakin, J.S., Weinberger, N.M.: Classical conditioning induces CS-specific receptive field plasticity in the auditory cortex of the guinea pig. *Brain Res.* 536, 271–286 (1990)
2. Recanzone, G.H., Schreiner, C.E., Merzenich, M.M.: Plasticity in the frequency representation of primary auditory cortex following discrimination training in adult owl monkeys. *J. Neurosci.* 13, 87–103 (1993)
3. Horikawa, J., Hosokawa, Y., Kubota, M., Nasu, M., Taniguchi, I.: Optical imaging of spatiotemporal patterns of glutamatergic excitation and GABAergic inhibition in the guinea-pig. *J. Physiol.* 497, 629–638 (1996)
4. Kilgard, M.P., Merzenich, M.M.: Cortical map reorganization enabled by nucleus basalis activity. *Science* 279, 1714–1718 (1998)
5. Bandrowski, A.E., Moore, S.L., Ashe, J.H.: Cholinergic synaptic potentials in the supragranular layers of auditory cortex. *Synapse* 41, 118–130 (2001)
6. Weinberger, N.M., Bakin, J.S.: Learning induced physiological memory in adult primary auditory cortex: Receptive field plasticity, model, and mechanisms. *Audiology & Neuro-Otology* 3, 145–167 (1998)
7. Bakin, J.S., Weinberger, N.M.: Induction of a physiological memory in the cerebral cortex by stimulation of the nucleus basalis. *Proc. Natl. Acad. Sci. U S A* 93, 11219–11224 (1996)
8. Herrero, J.L., Roberts, M.J., Delicato, L.S., Giesemann, M.A., Dayan, P., Thiele, A.: Acetylcholine contributes through muscarinic receptors to attentional modulation in V1. *Nature* 454, 1110–1114 (2008)

Possibility of Cantor Coding by Spatial Input Patterns

Yasuhiro Fukushima¹, Minoru Tsukada¹,
Ichiro Tsuda^{2,3}, Yutaka Yamaguti², and Shigeru Kuroda³

¹ Brain Science Institute, Tamagawa University, 6-1-1, Tamagawagakuen,
Machida, 194-8610, Japan

² Research Institute of Electronic Science, Hokkaido University, Japan

³ Department of Mathematics, Hokkaido University, Japan

y-fukusi@lab.tamagawa.ac.jp

Abstract. In rat CA1 pyramidal neurons under sub- and supra-threshold conditions, our previous study showed the potentials of Cantor coding, which is theoretically proposed by Tsuda and Kuroda. However, the coding could be explained by mean rate coding simply depended on input pattern history of the peak amplitude of each input. In order to confirm that Cantor coding includes any other factors except mean rate coding, we applied three spatially different patterns of electrical stimulations with similar peak amplitudes. Although the responses did not show the statistical significant self-similar property, the membrane responses show significant clustering property. Our results suggest that some factors, which does not simply depend on mean firing rate coding, was included in Cantor coding processing.

1 Introduction

Tsuda (2001) and Tsuda and Kuroda (2001, 2004) theoretically predicted the possibility of Cantor coding in CA3-CA1 network. Cantor coding provides an information coding scheme for temporal sequences of events. Our previous study showed the Cantor coding-like property in hippocampal CA1 neurons, where the clustering property was dependent on the magnitude of EPSP (Fukushima et al., 2007). Furthermore, the relation between input pattern and recorded responses was proved to be described as iterated function systems, which provides a direct evidence of the presence of Cantor coding, and also the coding quality was drastically improved by a newly invented virtual reconstruction method using plural neurons' data (Kuroda et al., in preparation).

In our previous results, Cantor coding in CA1 neurons could be also explained by mean rate coding, which simply depend on input pattern history of the peak amplitude of input sequence. However, neural processing in CA1 pyramidal neuron is not simply dependent on strength of input, but also dependent on sequential effect of input patterns (Tsukada et al., 1996, Tsukada et al., 2007). Cantor coding may also be dependent on sequential effect of neural inputs. In order to confirm that Cantor coding includes any other factors except mean rate coding (i.e. temporal pattern sensitivity), we applied three spatially different patterns of electrical stimulations with similar peak amplitudes and analyzed the Cantor coding properties.

2 Materials and Methods

Patch-clamp recording was made from pyramidal neuron of CA1 area in rat hippocampal slice. EPSPs were induced by extracellular electrical stimulation using three theta glass electrodes. A different random sequence of electrical stimulations in successive ten on-off signals (33Hz) containing three on signal, namely three pulses was applied in each stimulus at 10 s intervals during 20 min. The three patterns of electrical stimulation were applied by using one of three stimulating electrodes (Fig. 1), which induces no action potentials during the stimulation. There are two experimental conditions due to the difference between the amplitudes of EPSP by electrical stimulation of three spatial patterns; “different responses” and “similar responses” conditions. The maximum / minimum ratio of EPSP amplitudes between different spatial input patterns was 2.23 ± 0.37 in “different responses” condition, whereas it was 1.36 ± 0.11 in “similar responses” conditions.

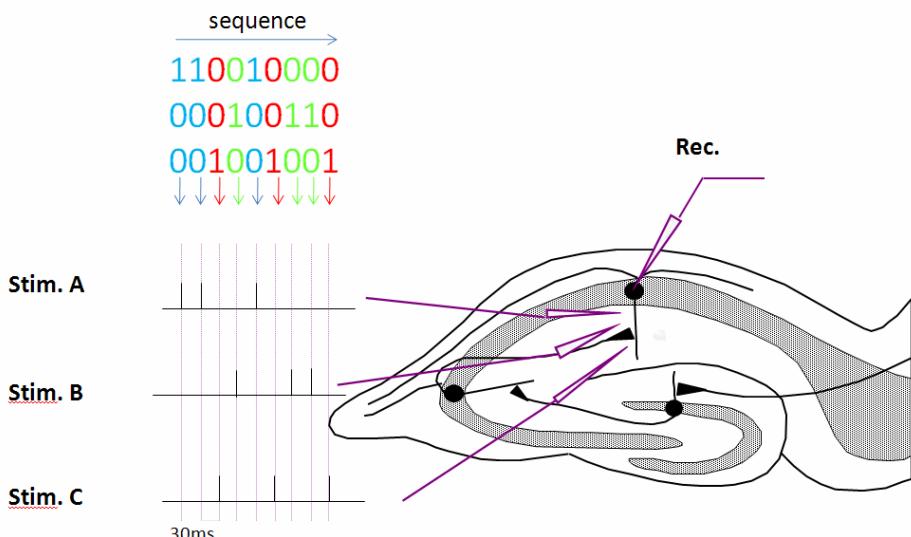


Fig. 1. Spatial input pattern

Both cluster and self-similarity indices were calculated as similar to our previous experiments (Fukushima et al., 2007). A cluster index indicates how well the distribution of responses is clustered by the patterns of electrical stimulation. Each response was categorized into three groups by the current pattern, one preceding pattern, or two preceding pattern of electrical stimulation. The clustering index indicates the probability that the distance between “the value concerned and weighted-center of its own group” is not shortest among the four distances between “the value concerned and weighted center of all four groups”.

A self-similarity index indicates a measure of hierarchical self-similarity. The four weighted centers for each time-history step of the spatial input pattern forms an observed pattern in amplitudes. The self-similarity measure is given by the number of neighboring permutations to get a target-ordered pattern corresponding to the standard pattern (the corresponding pattern to the 1st time-history step). The self-similarity index is normalized by the measure for the chance level.

3 Results and Discussions

Self-similarity index and cluster index were estimated for two conditions: “different responses” and “similar responses”. Self-similarity index in “different responses” condition were significantly lower than chance level at the history pair of “one and two” and “one and three” steps in the sequences (Fig. 2, left). This results are similar to those in “sub-threshold conditions” of our previous study (Fukushima et al., 2007). In contrast, self-similarity index in “similar responses” did not show the significant difference against the chance level at any pairs (Fig.2, right).

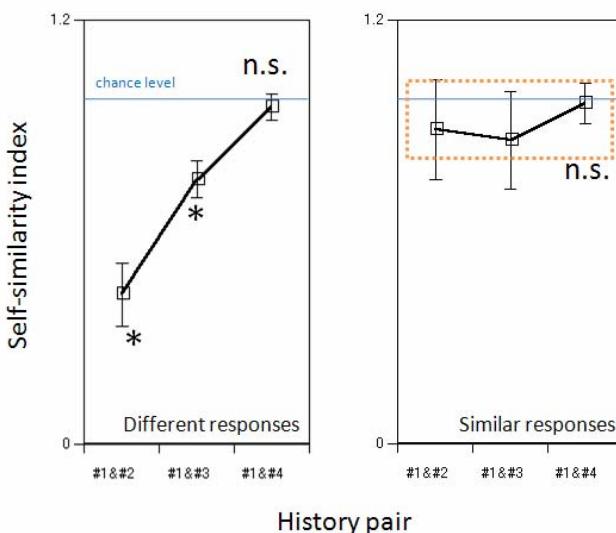


Fig. 2. Self-similarity index by spatial input pattern

Cluster index in “different responses” condition were significantly lower than chance level at the history of one and two steps before the sequences (Fig. 3, left). Although self-similar property did not showed the significant difference against chance level, clustering index in similar responses show significantly lower than chance levels (Fig. 3, right).

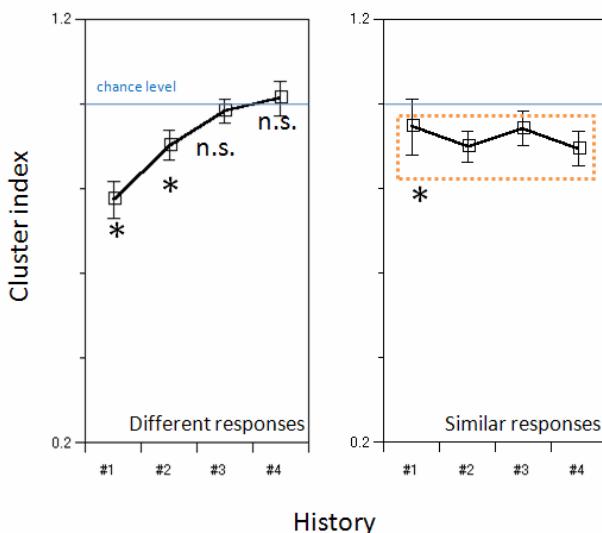


Fig. 3. Cluster index by spatial input pattern

These clustering properties were similar to theoretical Cantor coding system. Our results show that a certain factor, which does not simply depend on mean firing rate coding, was included in Cantor coding processing. These results show that sequential effect of input, which is derived from different site of dendrites, is important for information processing in CA3-CA1 circuit and its memory systems.

Acknowledgments. This study was supported by Grant-in-Aid for a COE Fellow (Y. F) from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan and by the 21st Century Center of Excellence Program (Integrative Human Science Program, Tamagawa Univ.). One of the authors (I.T.) was supported by Grant-in-Aid for Scientific Research on Priority Areas - Integrative Brain Research - from the Ministry of Education, Culture, Sports, Science and Technology of Japan (18019002), and also supported by Grant-in-Aid for Scientific Research on Priority Areas - Understanding of Mobiligence - from the Ministry of Education, Culture, Sports, Science and Technology of Japan (18047001).

References

1. Tsuda, I.: Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems. *Behavioral and Brain Sciences* 24(5), 793–847 (2001)
2. Tsuda, I., Kuroda, S.: Cantor coding in the hippocampus. *Japan J. Indust. Appl. Math.* 18, 249–258 (2001)
3. Tsuda, I., Kuroda, S.: A Complex Systems Approach to an Interpretation of Dynamic Brain Activity II: Does Cantor Coding Provide a Dynamic Model for the Formation of Episodic Memory? In: Erdi, P., et al. (eds.) *Computational Neuroscience: Cortical Dynamics*. LNCS, vol. 3146, pp. 129–139. Springer, Heidelberg (2004)

4. Fukushima, Y., Tsukada, M., Tsuda, I., Yamaguti, Y., Kuroda, S.: Spatial clustering property and its self-similarity in membrane potentials of hippocampal CA1 pyramidal neurons for a spatio-temporal input sequence. *Cognitive Neurodynamics* 1, 305–316 (2007)
5. Tsukada, M., Saito, H.A., Aihara, T., Kato, H.: Hippocampal LTP depends on spatial and temporal correlation of inputs. *Neural Netw.* 9, 1357–1365 (1996)
6. Tsukada, M., Yamazaki, Y., Kojima, H.: Interaction between the Spatio-Temporal Learning Rule (STLR) and Hebb type (HEBB) in single pyramidal cells in the hippocampal CA1 Area. *Cognitive Neurodynamics* 1, 157–167 (2007)

A Neural Network Model for a Hierarchical Spatio-temporal Memory

Kiruthika Ramanathan^{1,*}, Luping Shi¹, Jianming Li¹, Kian Guan Lim¹,
Ming Hui Li², Zhi Ping Ang², and Tow Chong Chong^{1,2}

¹ Data Storage Institute, (A*STAR) Agency for Science, Technology and Research, DSI Building, 5 Engineering Drive 1, Singapore 117608

² Dept of Electrical and Computer Engineering, National University of Singapore
kiruthika_r@dsi.a-star.edu.sg

Abstract. The architecture of the human cortex is uniform and hierarchical in nature. In this paper, we build upon works on hierarchical classification systems that model the cortex to develop a neural network representation for a hierarchical spatio-temporal memory (HST-M) system. The system implements spatial and temporal processing using neural network architectures. We have tested the algorithms developed against both the MLP and the Hierarchical Temporal Memory algorithms. Our results show definite improvement over MLP and are comparable to the performance of HTM.

1 Introduction

The HST-M is developed based on two principles of cognitive modeling. The first is the neural aspect, where, development of recognition is intertwined with that of the brain, both at the cellular and at the modular levels, and where learning is an incremental, connectionist process. The second is the computational modeling, which forces theories of brain structure to be explicit, resulting in a more detailed specification than what is available in works of Psychology and Neuroscience.

Recent books such as Rethinking Innateness [1] and The Algebraic Mind [2] argue that representation of information in the brain can be achieved by implementing a set of connectionist networks. In the field of connectionism, much work has been done that attempt to model cognition based on neural networks. Of these, several models such as Neocognitron [4, 5], HMAX [6, 7] and HTMs [8] use the principles of hierarchy and the uniformity of the neocortex to design algorithms to improve pattern recognition. While Neocognitron and HMAX are spatial algorithms, HTM proposes a Bayesian model that makes use of the additional temporal information in the data to solve the pattern recognition problem, thereby leading to better results.

In this paper, we propose a connectionist model to perform pattern recognition based on the spatial and temporal properties of the data. The model proposed is hierarchical, uniform and learns in an unsupervised manner. While the principle of the model is similar to that of the HTM, the implementation is based on neural networks, as opposed to a Bayesian model that is proposed by Numenta [8]. The model has been tested on three datasets and the results compared both with traditional MLPs and an implementation of HTM [17].

* Corresponding author.

The work of Vernon Mountcastle cited in Hawkins [9] pioneered a set of research experiments on the human neocortex, which showed that the cortical structure is fairly uniform, with the same general computation performed in each unit of the cortex, as shown in Fig. 1. Each of what we consider specialized sections (visual cortex, auditory cortex etc) had, in fact, learnt its task based on the properties of the training patterns that had been presented to it. A section of the brain could, as has been shown in several research works [10, 11], learn over time to solve a different task.

Fig. 1a and b show the general replicating structure of the cortical column. Fig. 1c shows the hierarchical connections between the various layers of the visual cortex, discussed by Cadieu et al [6] and George et al [8].

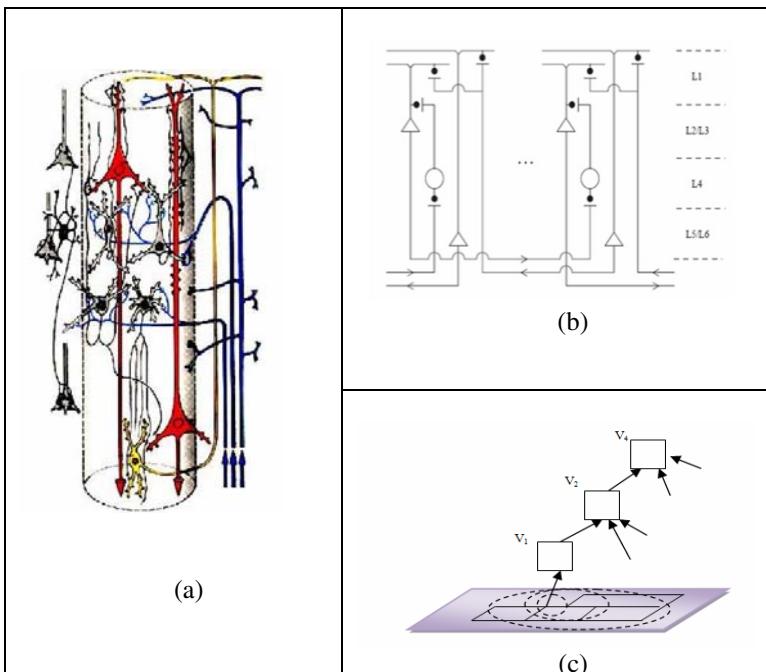


Fig. 1. (a) A depiction off the cortical column as identified by Mountcastle showing the different types of replicating neurons (b) The replicating structure of the cortical column [18] (c). Hierarchy in the layers of the visual cortex when corresponding to an input image.

New evidence comes forth every day that learning is a hierarchical process [12]. Research has long shown that architectures of perception, such as vision and audition are hierarchical in nature. Aspects of memory, as such chunking, also make use of hierarchical architectures. This hierarchical process is not learnt, but is inherent and displayed even in young babies [13].

The HST-M uses principles of uniformity and hierarchy as its roots. A spatio-temporal neural network is the fundamental building block of the HST-M, and is replicated throughout a hierarchical structure.

2 The HST-M Structure and Training Algorithm

Fig. 2b to d shows the structure of the HST-M and that of the component neural networks. We work with a dataset $\mathbf{D} = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n\}$, composed of the patterns $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n$, where each pattern is represented by a m dimensional feature vector such that $\mathbf{P}_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,m}\}$.

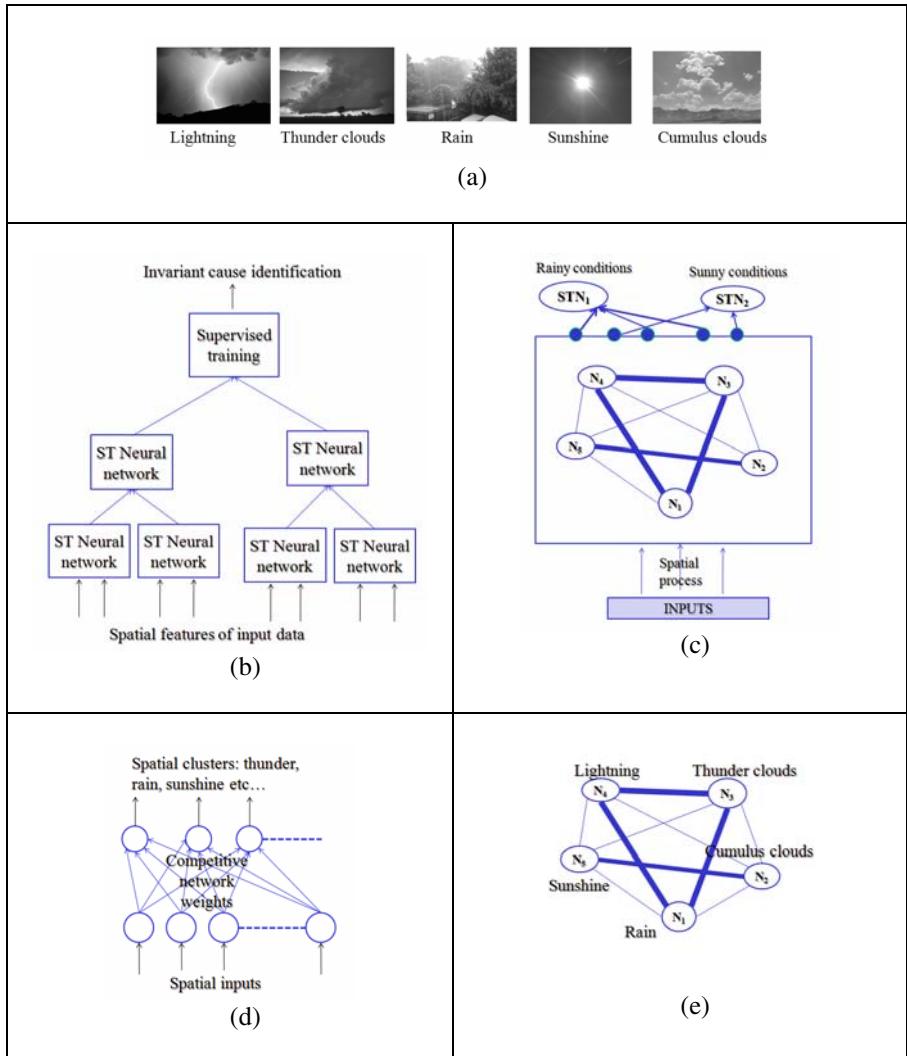


Fig. 2. The connectionist structure of a HST-M showing (a) A sample set of data, (b) The system architecture, (c) the structure of a spatio-temporal node, (d) the spatial processing unit and (e) the temporal processing unit

Consider now, that the dataset \mathbf{D} consists of 32×32 images representing weather conditions, as indicated in Fig. 2a. The images are of a continuing time sequence, and consist of scenes from five categories: {Sunshine, Rain, Cumulus Clouds, Thunder Clouds, Lightning}. Intuitively, we know that, based on the co-occurrence correlation of the information, the data can be divided into two associated groups consisting of {Sunshine, Cumulus Clouds} and {Rain, Thunder Clouds, Lightning}. In other words, the patterns in the associated groups have a higher probability of appearing together in time.

The HST-M processes the patterns by first splitting the m features into k sub vectors of arbitrary size, such that $\mathbf{P}_i = \{\mathbf{p}_{i,1}, \mathbf{p}_{i,2}, \dots, \mathbf{p}_{i,k}\} \text{ where } k < m$. Each sub vector $\mathbf{p}_{i,j}$ is fed into one of the lower Spatio-Temporal (ST) Neural Networks of the HST-M network in Fig. 2b. The input to a node in the upper layer is the concatenation of the outputs of the lower layer nodes linked to it.

With reference to the example in Fig. 2, each image is truncated into a smaller segment, say 8×8 . The lower layer of the HST-M is therefore made up of 4 ST Neural Networks, each processing a segment of the image.

Each ST Neural network Fig. 2c is made up of the following architecture: (a) An incremental competitive network (Fig. 2d) that processes the input data according to its spatial properties and (b) A Temporal neural network (Fig. 2e) that connects spatial groups according to their temporal co-occurrence correlation.

Spatial information processing using a competitive neural network: For the problem in Fig. 2, each competitive neural network [3] illustrated in Fig. 2d takes as its input a 64 dimensional vector, representing a segment of the image. The competitive neural network determines the number of spatial clusters in the image segments, thereby dividing the data into 5 spatial clusters, as indicated in Fig. 2d.

Our implementation of competitive networks differs from the traditional Kohonen implementation in that it uses a user defined threshold value τ_1 to dynamically change the number of neurons. The algorithm begins by setting the weights of the first neuron to correspond to the first pattern. Using a defined threshold, the weight and number of neurons is incremented until the system stagnates. The algorithm is therefore suited for incremental adaptive learning.

Temporal information processing: Based on the non-directional co-occurrence correlation of the input data, the weights are adjusted to values as indicated in Fig. 2e, using the update rule in (1). In general, where $S_{(t=i)}$ represents the competitive neuron activated at time $t=i$, and where α and β are two outputs of the spatial processor, and \mathbf{S} represents the state space of the spatial network outputs, a discrete update rule can be expressed as

$$\forall \alpha, \beta \in \mathbf{S}, \text{if } |S_{(t=i)} = \alpha| \& |S_{(t=i+1)} = \beta| W_{\alpha, \beta} ++ \quad (1)$$

Where W refers to the weights of the temporal network.

An ST Neural Network (Fig. 2) therefore consists of a temporal network, whose inputs are the outputs of the spatial network. A maximum group size is imposed to

form groups of temporal patterns based on the weights. Based on the processing of the spatio-temporal neurons, the images in Fig. 2a will be finally grouped into two clusters, indicated in Fig. 2b, representing “rainy conditions” and “sunny conditions” respectively.

Hierarchical learning: The implementation of hierarchical learning ensures that each node performs a decision at its own level. For instance, the node which processes information represented in the lower left hand corner of the images in Fig. 2a is likely to form only one temporal group as all the images being given to the system show a black patch. The ability of each ST Neural Network to make inferences at its own level makes the system robust to scaling and skewing, as can be seen from the experimental results.

3 Experimental Results

3.1 Details of Experimental Data

Experiments were run on three datasets: Bitworm, LMH letter recognition, and Pictures. Of these, the Bitworm and the Pictures datasets are benchmark data obtained from Numenta [16]. The LMH letter recognition dataset was generated by us. The details of the datasets are given in Table 1.

Table 1. Details of experimental datasets

Name	Number of inputs	Number of Classes	Number of patterns for training	Testing data
Bitworm	16	2	400	Scaled worms of size 3 to 13 bits
LMH letter recognition	256 (16x16)	3	300	Scaled and Skewed images
Pictures	1024 (32x32)	41	398	Distorted images and Gaussian noise

The bitworm data [16] represents a one dimensional 8 bit worm moving left and right across the 16 bit screen. The worm belongs to one of two classes, solid and textured. Samples of the position of the worm at different time intervals were taken to train the HST-M. Generalization on the bitworm HST-M network was performed using discrete, patterns of solid and textured bitworms. The worms presented to the system range in length between 3 and 13 bits.

The LMH training data consists of moving 16x16 images of the three letters L,M and H, where the patterns undergo training on scaling and translation. Testing is performed using a set of distorted patterns. Fig. 3 shows samples of training and testing data. In the pictures dataset [16] we used consists of 398 black and white images which represent 41 kind of moving objects. The HST-M is tested using noisy data, where the training patterns are distorted using a Gaussian noise of varying standard deviations.

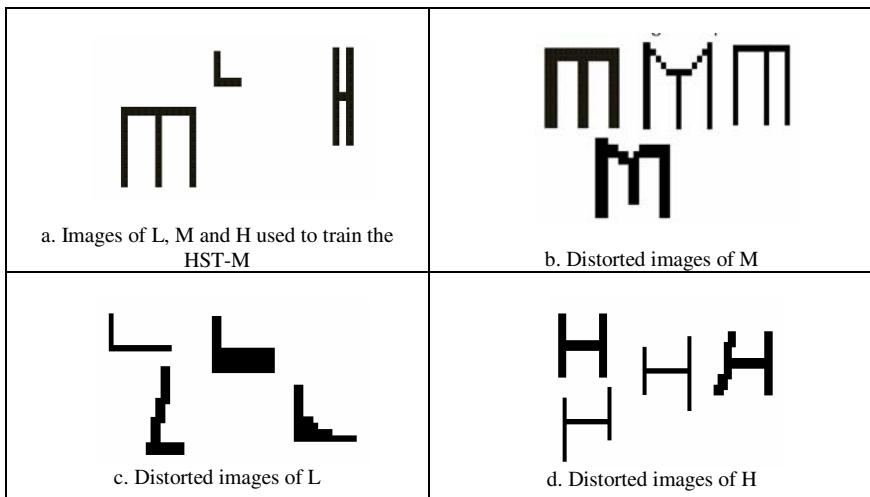


Fig. 3. Examples of training and testing images of the LMH data for the HST-M memory. The images in b, c and d were used in testing to determine the generalization accuracy of the system.

3.2 Experimental Results

Table 2 shows the experimental results obtained using the various datasets. The results of the HST-M was compared with two networks, traditional MLP implemented using Cascade Correlation [15] with the Quickprop [14] algorithm and an implementation of the HTM architecture by Numenta which proposes the Spatio Temporal architecture using Bayesian probabilistic models [8].

Table 2. Details of experimental results showing the mean and standard deviation of HST-M accuracy compared with the accuracy off MLP and HTM

Name	HSTM architecture	Size of input vector to each bottom node	Mean HST-M accuracy (Std dev) (%)	MLP accuracy (%)	Accuracy of HTM implementation (%)
Bitworm	Multiple	4/8	96.41 (2.46)	60.24	97.86
LMH letter recognition	16,4,1	16	94.43 (5.85)	33.33	92.31
Pictures	32,16,4,1	16	83.8 (9.57)	10.52	80.62

The HST-M algorithm shows a definite improvement of traditional MLP methods. The average accuracy obtained is also comparable to the performance of HTMs. The maximum accuracy obtained is 99.3%, 96.0% and 91.2% for the bitworm, LMH and pictures data respectively, which is higher than the results obtained by HTM.

4 Conclusions and Future Work

The HST-M builds up upon existing literature, making use of the concepts of hierarchy, spatial, and temporal processing to classify patterns. This method of training is considered effective as it models the hierarchical structure of the neo-cortex, thereby achieving time invariant pattern recognition. In this paper, we have studied the method of developing and training a hierarchical spatio-temporal memory structure using neural network algorithms and representations and have tested the system developed on several datasets. The initial results obtained are encouraging, especially in the field of image processing and recognition, where the training algorithms show capability in recognizing noisy and distorted images and is also capable of dealing with scaling and skewing.

Avenues for further work in this area are many. In the algorithm end, they include optimizing the individual neural network algorithms used in training, reducing the number of user-input parameters by automatically determining them in relation to the training set, and determining automatically the optimal hierarchical structure for a problem by using a growing algorithm. In the application end, potential future work includes experimenting with real world, grayscale data and experimenting in other sensory domains.

References

1. Elman, J.L., Bates, E.A., Johnson, M.H., Smith, A.K., Parisi, D., Plunkett, K.: *Rethinking Innateness: A Connectionist Perspective on Development (Neural Networks and Connectionist Modeling)*. MIT Press, Cambridge (1996)
2. Marcus, G.F.: *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. MIT Press, Cambridge (2001)
3. Kohonen, T.: *Self Organizing Maps*. Springer, Berlin (1997)
4. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* 36(4), 93–202 (1980)
5. Fukushima, K., Miyake, S., Ito, T.: Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics* 13(3), 826–834 (1983)
6. Cadieu, C., Kouh, M., Pasupathy, A., Conner, C., Riesenhuber, M., Poggio, T.A.: A Model of V4 Shape Selectivity and Invariance. *J. Neurophysiol.* 98, 1733–1750 (2007)
7. Riesenhuber, M.: Appearance Isn't Everything: News on Object Representation in Cortex. *Neuron* 55, 341–344 (2007)
8. George, D., Hawkins, J.: A hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex. *International Joint Conference on Neural Networks* 3, 1812–1817 (2005)
9. Hawkins, J., Blakeslee, S.: *On Intelligence*. Owl Books, New York (2004)
10. Rakic, P., Bourgeois, J.P., Eckendorf, M.F., Zecevic, N., Goldman-Rakic, P.S.: Concurrent Overproduction of synapses in diverse regions of the primate cerebral cortex. *Science* 232, 232–235 (1986)
11. Stelzner, D.J.: Short-circuit recovery from spinal injury. *Nature Medicine* 14(1), 19–20 (2008)
12. Van Hessen, D.C., Anderson, C.H., Felleman, D.J.: Information processing in the primate visual system: an integrated systems perspective. *Science* 2255(5043), 419–423 (1992)

13. Babies use grown up memory tricks. New Scientist Magazine 17 (July 19, 2008)
14. Fahlman, S.E.: Faster learning variations on back-propagation: An empirical study. In: Proceedings of the 1988 Connectionist models Summer School (1988)
15. Fahlman, S.E., Lebiere, C.: The cascade correlation learning architecture. In: Advances in neural information processing systems 2, pp. 524–532. Morgan Kaufmann, San Francisco (1990)
16. Numenta, <http://www.numenta.com>
17. George, D., Jaros, B.: The HTM learning algorithms,
[http://www.numenta.com/for-developers/education/
Numenta-HTM_Learning_Algos.pdf](http://www.numenta.com/for-developers/education/Numenta-HTM_Learning_Algos.pdf)
18. Thomas, D.: A computational model of the cerebral cortex. In: The Proceedings of Twentieth National Conference on Artificial Intelligence (AAAI 2005), pp. 938–943. MIT Press, Cambridge (2005)

Time-Varying Synchronization of Visual ERP during Sentences Identification

Minfen Shen¹, Jialiang Chen¹, and K.H. Ting²

¹College of Engineering, Shantou University, China (515063)

mfshen@stu.edu.cn

²College of Engineering, The University of Hong Kong, Hong Kong, China

Abstract. The study of the synchronization of EEG signals can help us to understand the underlying cognitive processes and detect the learning deficiencies. The cognitive and information processing take place in different brain regions at different time. To investigate how these distributed brain regions are linked together and the information is exchanged with time, this paper proposes a modern time-frequency coherent analysis that employs an alternative way for quantifying synchronization with both temporal and spectral resolution. Wavelet coherent spectrum is defined and computed from the EEG data set such that the cross wavelet magnitude spectra serves to indicate the degree of coherence and the cross wavelet phase can be used to provide the direction of information flow between channels on different brain regions. Several real ERP data are collected based on the cognitive tasks of sentences identification in both English and Chinese. It is observed from the time-varying synchronization that there are obviously differences during identifying both Chinese and English sentences.

Keywords: Synchronization, EEG signals, Wavelet, Time-varying coherence.

1 Introduction

Electroencephalogram (EEG) is regarded as the summed electrical activity of very large numbers of neurons. When the learning task is carried out, the corresponding cognitive and information processing take place in different brain regions. It was shown that the brain works as a distributed parallel processing system. Studying both local and long-range neuronal synchronization can help to understand the underlying cognitive processes. To measure degree of synchronization conventional coherence analysis is an useful tool for the study as it is a linear method to calculate correlation between two time series in the frequency domain. However its lack of temporal resolution limits its application to stationary processes [1-3].

In this contribution, a coherency measure via continuous wavelet transform is proposed. Wavelet transform provide sufficient temporal and spectral resolution for time series analysis. While wavelet analysis has already been a very popular analysis tool to study non-stationary signal including EEG, cross wavelet analysis can explore the amount of synchronized activities among multiple channels[4-6]. Among various mother wavelets, the Morlet wavelet is chosen as we can benefits from its nature as a

complex wavelet, which provides phase relationship between the time series under study. To investigate the synchronization of the EEGs when the subjects were identifying the specified sentences, the proposed method was used for quantifying synchronization with both temporal and spacial resolution which enables to indicate the degrees of synchronization at each time during sentences identification[7-9].

2 Time-Varying Coherence

Generally, cross-correlation function has been widely used to quantify the linear relationship among the frequency components of two signals. Let $x(t)$ and $y(t)$ denote two different random processes that are both wide-sense stationary and ergodic, the corresponding cross-correlation function is defined as

$$R_{xy}(\tau) = E[x(t)y(t)] \quad (1)$$

where $E[\cdot]$ denotes the statistical expectation. Thus, the magnitude-squared coherence (MSC) can be obtained from the normalization of the cross-spectral density function

$$\gamma_{xy}^2(f) = \frac{|S_{xy}(f)|^2}{S_x(f)S_y(f)} \quad (2)$$

The bound of MSC is between 0 and 1.

The conventional coherence above based on the Fourier analysis is only valid for analyzing the long-term stationary processes. However, the investigation of both local and long range neuronal synchronization enable to understand the underlying cognitive processes, such as memory, remember and learning. Thus, the time-varying characteristics of the synchronization in the signals should be considered since the practical EEG/ERPs are highly non-stationary. The dynamic spectral analysis of a non-stationary process $x(t)$ can be expressed in terms of the TFD: $S_x(t, f)$. Its corresponding time-varying cross-spectral density function is defined as

$$S_x(t, f) = S_x(t, f)S_y(t, f) = |S_{xy}(t, f)|e^{j\phi_{xy}(t, f)} \quad (3)$$

Similarly, the time-varying coherence between non-stationary process $x(t)$ and $y(t)$ is given as

$$\gamma_{xy}(t, f) = \frac{S_{xy}(t, f)}{\sqrt{S_x(t, f)S_y(t, f)}} = |\gamma_{xy}(t, f)|e^{j\phi_{xy}(t, f)} \quad (4)$$

where the amplitude of $\gamma_{xy}(t, f)$ measures the degree of the local synchronization, while the $\phi_{xy}(t, f)$ provides the information regarding the phase difference between two process $x(t)$ and $y(t)$ at time t.

3 Wavelet-Based Coherence Analysis

Wavelet has proved to be a very effective tool for analyzing non-stationary signal. For this reason, we combine the wavelet transform with the coherence to form a novel time-varying synchronization analysis that explores the amount of synchrony among the multiple channel brain activities. The benefit of using the wavelet transform is based on the fact that the time-resolution is changing with frequency so that the multi-resolution of the signal become possible. Compared with Fourier analysis, wavelet analysis provides more useful information regarding the nature of the underlying EEG signals and the corresponding cognitive process.

Let $x(t)$ denotes the underlying signal, the wavelet transform of $x(t)$ with respect to the specified wavelet function $\phi_{xy}(t, a)$ can be defined as

$$W_x(\tau, a) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} x(t) \phi\left(\frac{t-\tau}{a}\right) dt \quad (5)$$

where $\phi(t)$ represents a mother wavelet. The wavelet transform, characterized by time-shifting and scaling a mother wavelet, become a multi-resolution analysis tool which is very adequate to characterize the time-varying synchronization of the non-stationary EEG signals. The cross wavelet spectrum between $x(t)$ and $y(t)$ is defined as

$$WS_{xy}(\tau, a) = W_x(\tau, a) W_y^*(\tau, a) = |WS_{xy}(\tau, a)| \cdot e^{j\phi_{wxy}(\tau, a)} \quad (6)$$

Thus, the time-varying coherence in (4) can be generalized to the wavelet coherence such that

$$\gamma_{wxy}(\tau, a) = \frac{|WS_{xy}(\tau, a)|}{\sqrt{WS_x(\tau, a) \cdot WS_y(\tau, a)}} \quad (7)$$

The corresponding magnitude squared coherence and the phase difference can be defined

$$\gamma_{wxy}^2(\tau, a) = \frac{|WS_{xy}(\tau, a)|^2}{WS_x(\tau, a) \cdot WS_y(\tau, a)} \quad (8)$$

$$\phi_{wxy}(\tau, a) = \tan^{-1} \left[\frac{\text{Im}[WS_{xy}(\tau, a)]}{\text{Re}[WS_{xy}(\tau, a)]} \right] \quad (9)$$

On the one hand, the wavelet MSC is bounded between 0 and 1, which provide the measurement of time-varying synchronization of two non-stationary EEG signals within a specified scale or frequency band and time instant. On the other hand, the phase difference $\phi_{xy}(\tau, a)$ between two EEG signals enables to reveal the direction of the information flow of the transient brain activities when cognitive task is processed. To provide both time-varying synchronization and phase information between two EEG signals, the continuous wavelet transform based on complex Morlet wavelet is employed to calculated the time-varying coherence.

4 Simulations

To test the behavior of the proposed method, two hypothetical non-stationary sinusoidal functions with 1000ms length were generated as:

$$x_1(t) = \begin{cases} \sin(10\pi t + 30\pi t / 180) & \text{for } 0 \leq t \leq 250\text{ms} \\ \sin(70\pi t) & \text{for } 250 \leq t \leq 500\text{ms} \\ \sin(40\pi t - 70\pi t / 180) & \text{for } 500 \leq t \leq 750\text{ms} \\ \sin(24\pi t + 100\pi t / 180) & \text{for } 750 \leq t \leq 1000\text{ms} \end{cases} \quad (10)$$

$$x_2(t) = \begin{cases} \sin(10\pi t + 30\pi t / 180) & \text{for } 0 \leq t \leq 250\text{ms} \\ \sin(70\pi t - 100\pi t / 180) & \text{for } 250 \leq t \leq 500\text{ms} \\ \sin(40\pi t) & \text{for } 500 \leq t \leq 750\text{ms} \\ \sin(24\pi t + 30\pi t / 180) & \text{for } 750 \leq t \leq 1000\text{ms} \end{cases} \quad (11)$$

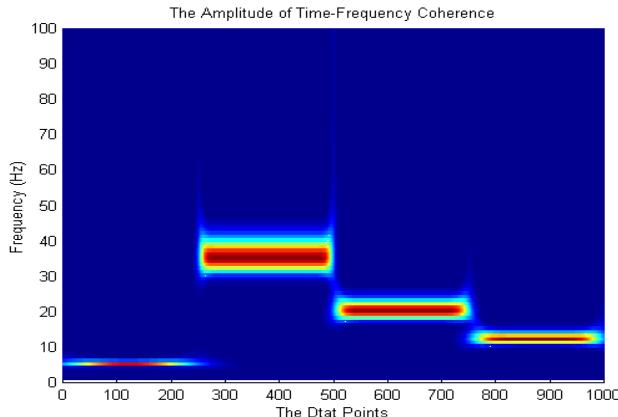


Fig. 1. The time-varying synchronization of the underlying signals is clearly identified

Fig. 1 demonstrates the wavelet coherence with frequency ranging from 1 to 100 Hz and time scale from 1 to 1000ms. The time-varying coherent structures of the underlying test signals were clearly identified. The phase differences for segment 2, 3 and 4 were also clearly estimated in Fig. 2 that is consistent to the mathematical representations.

The amplitude coherence analysis result shows that these two simple piecewise stationary have the full synchronization in different segments with different frequency at 5Hz, 35Hz, 20Hz and 12Hz. This simulation indicates that the proposed wavelet coherence enables to quantify the degree of the time-varying synchronization and the corresponding direction of information flow between two channel non-stationary EEG signals. Thus, the temporal resolved coherence of two signals was clearly provided. Moreover, the wavelet coherence is highly sensitive with time and related with the local synchronization of the signals.

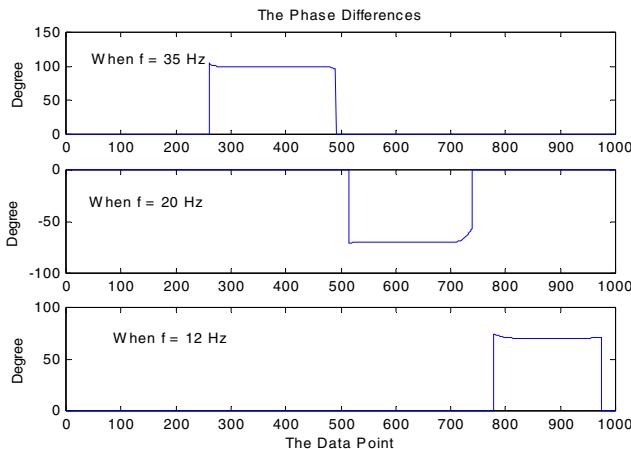


Fig. 2. The phase relationship, from top to bottom: the phase differences for segment 2, 3 and 4

5 Analysis of EEG Signals during Sentence Identification

The linguistic ability of human being is one of the many salient characteristics that distinguish us from other species. To understand the linguistic ability of human brain at the physiological level, the measurement of EEG provides valuable insight. Once an individual is exposed to words (visual) and/or speech (auditory), the brain start to sense, percept and understand it and try to response in one's behavior. During these processes, the electrical signals of the brain, which represents the communication between neurons, can be captured by electrodes on scalp. For language studies in particular, we can know when and where the brain:

- (1) senses the visually or auditory presented stimuli;
- (2) recognizes the stimuli lexically as words and understands their meaning;
- (3) combines the meaning of words at the sentential level;
- (4) produces proper behavioral response to the stimuli.

As EEG signals are highly non-stationary, it is the wavelet method described in the previous sections that provides powerful tools to investigate the time-frequency properties of the distributed brain network. 22 normal healthy females were participated in a language education experiment. 11 of the participants were Primary 4 pupils with mean age 9 years, while the rest 11 were Form 4 pupils with mean age 15 years. The purpose of study is to investigate the effectiveness of various language teaching methods. After attending certain language courses, the subjects were tested with EEG experiments. In the experiment, the language materials were assumed at the appropriate level and the subjects were assumed know the material well.

The EEG experiment required the subjects to judge the syntactic correctness of visually presented sentences. 30 of the presented sentences were in Chinese and another 30 were in English. Half of the sentences contained syntactic errors. For Chinese, 12 sentences contained quantity errors and 3 contained pronoun errors. For

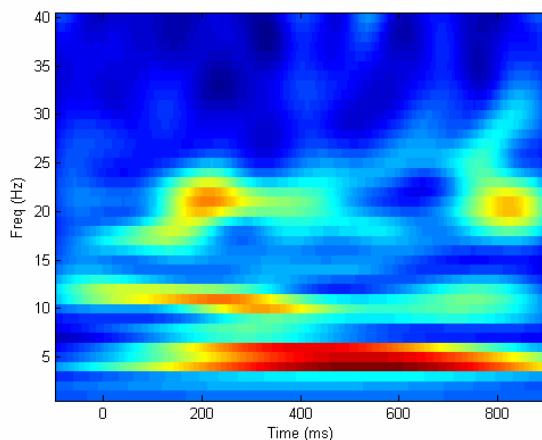


Fig. 3. The averaged wavelet magnitude spectrum over the 129 channel EEG signal for a subject reading a grammatically incorrect English sentence

English, only tense errors were introduced. After the subject had watched the complete sentence, they had to press one of the two buttons to show their judgment on the correctness of the sentence. EEG data for each subject was collected from the 128-channel EEG system, sampled at 250Hz. ERPs were segmented with 100ms before and 900ms after stimulus onset.

The multi-channel EEG data are transformed by complex CWT. As an example, we select one of the trials for demonstration. Fig. 3 and 4 show the response when the subject was reading the grammatically incorrect English sentence “Last Sunday, Ann will happy.” The ERP was segmented at the moment when the word “will” was presented. In order to extract time-frequency features over the highly redundant transformed coefficients, the CWT coefficients are averaged across channels (see Figure 3). The “hot spot” at 100ms-12Hz, 200ms-20Hz, 300ms-10Hz and 500ms-5Hz are extracted for scalp mapping (see Figure 4).

In the experiment for the neuro-cognitive study of Chinese and English language comprehension, there are 30 stimuli (sentences) presented to the subject for each language. Here we only extracted a few results as a demonstration of how the proposed time-frequency method can be applied for the qualitative analysis of EEG in real situation. From the primary results, we can observe the wavelet magnitude and phase spectrum over the scalp surface for both English and Chinese sentence comprehension. With the color-coded scalp map, we can briefly conclude the few stages of neural processing during first and second language comprehension. For the chosen example, it shows that a fairly strong alpha component lied at about 10-12Hz is induced from the early stage of the brain process, and last quite a long duration for about 400ms. This alpha component exhibits a bit different behavior for the early stage part (~100ms) and the later stage part (200-400ms). The early alpha focuses mainly at the occipital area which is quite consistent with the experimental paradigm, as the sentence stimuli are presented visually to the subject. The neural synchronization at this early stage is merely local within the visual area. On the other

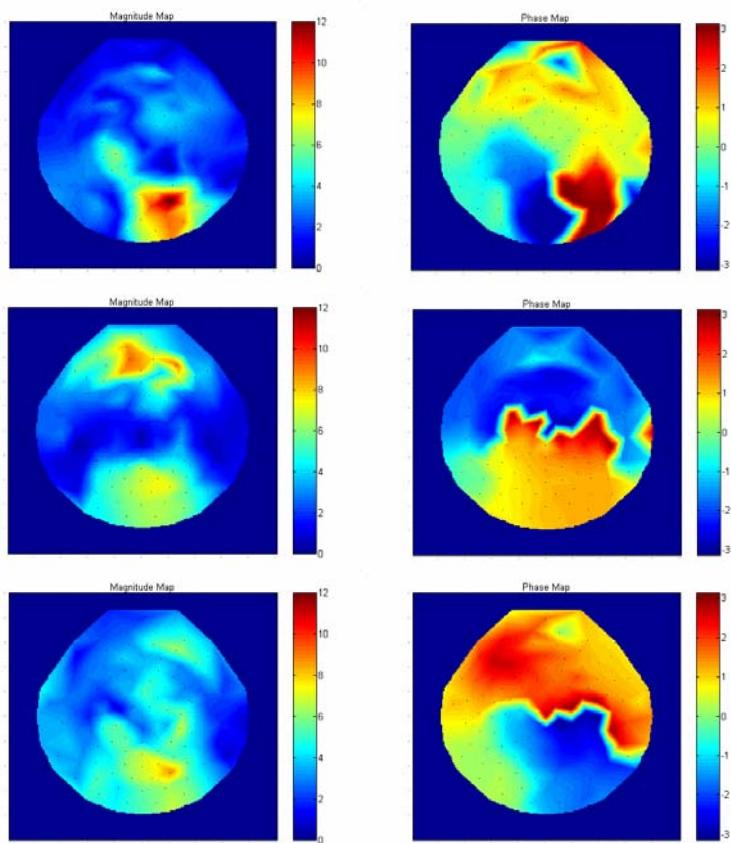


Fig. 4. The 3 phases of brain process for a subject comprehending the word “will” in the grammatically incorrect sentence “Last Sunday, Ann will happy.” The scalp map is extracted at 100ms-12Hz, 200ms-20Hz and 300ms-10Hz.

hand, the later stage alpha component focuses both on occipital and frontal region, with approximately opposite phase between the anterior and posterior area. The scalp maps in Fig.4 shows that there is a long range synchronization between the primary processing area (occipital) and higher-order processing area (frontal). For the late process stage, a low frequency ($\sim 4\text{Hz}$) component dominates with a local synchronization near central to frontal area. As the experiment requires the subject to do syntactic judgment on the presented sentences, the result is consistent to the traditional finding on the syntactic-sensitive P600 which shows more posterior distribution than the semantic-sensitive parietal N400.

6 Conclusions

An time-frequency coherent analysis of visual ERP signal for studying brain synchrony in a visual target detection experiment was proposed. Continuous wavelet

transform based on the complex wavelet is first applied on the raw multichannel ERP signals. Cross wavelet spectra is defined such that the cross magnitude spectra reveals the localized coherency power in the time-frequency plane and the cross phase spectra reveals the direction of information flow from one channel to another. Both simulations and the real ERP signals analysis have shown its efficiency of providing local synchronization between two non-stationary ERP recordings. Real visual ERP data are collected based on the cognitive tasks of sentences identification in both English and Chinese. It is observed from the wavelet synchronization that there are obviously differences during identifying both Chinese and English sentences.

Acknowledgement

This work is supported by the National Natural Science Foundation (60571066) and the Natural Science Foundation of Guangdong, respectively.

References

- Nunez, P.L., Srinivasan, R.: EEG coherence I.: statistics, reference electrode, volume conduction, laplacians, cortical imaging and interpretation at multiple scales. *Electroen. and Clin. Neurophysiol.* 103, 499–515 (1995)
- Anderson, C.W., Stoltz, E.A.: Multivariate autoregressive models for classification of spontaneous electroencephalogram during mental tasks. *IEEE Trans. Biomet. Eng.* 45(3), 277–286 (1998)
- Moller, E., Schack, B.: Instantaneous multivariate EEG coherence analysis by means of adaptive high-dimensional auto regressive models. *J. of Neuroscience Methods* 105, 143–158 (2001)
- Andrew, C., Pfurscheller, G.: Dependence of coherence measurements on EEG derivation type. *Med. & Biol. Eng. & Comput.* 34, 232–238 (1996)
- Schack, B., Rappelsberger, P.: Adaptive phase estimation and its application in EEG analysis of word processing. *J. of Neuroscience Methods* 93, 49–59 (1999)
- Schack, B., Grieszbach, G.: Dynamic cross spectral analysis of biological signals by means of bivariate ARMA processes with time dependent coefficients. *Med. & Bio. Eng. & Com.* 33, 605–610 (1995)
- Schack, B., Krause, W.: Instantaneous Coherence as a Sensible Parameter for Considering Human Information Processing. In: *IEEE Proceedings of ICPR 1996*, pp. 45–49 (1996)
- Weiss, S., Rappelsberger, P.: EEG coherence within 13–18 Hz band as a correlate of a distinct lexical organization of concrete and abstract nouns in humans. *Neuroscience Letters* 209, 17–20 (1996)
- Miltner, W.H.R., Braun, C.: Coherence of gamma-band EEG activity as a basis for associative learning. *Nature* 397, 434–435 (1999)

Neural Mechanism of Synchronous Firing of Inferior Temporal Cortex in Face Perception

Kazuhiro Takazawa¹ and Yoshiki Kashimori^{1,2}

¹ Graduate School of Information Network Sciences,

University of Electro-Communications, Chofu, Tokyo, 182-8585 Japan

² Department of Applied Physics and Chemistry, University of Electro-Communications,

Chofu, Tokoyo, 182-8585 Japan

kashi@pc.uec.ac.jp

Abstract. Understanding the neural mechanism of object recognition is one of the fundamental challenges of visual neuroscience. However, little is known about how the information about a whole object and its parts are represented in inferior temporal (IT) cortex. To address this issue, we focus on the neural mechanism of face perception. To investigate the mechanism, we made a model of IT cortex, which performs face perception via an interaction between different IT networks. The model was made based on the face information processed by three resolution maps in early visual areas. The network model of IT consists of four kinds of networks, in which the information about a whole face is combined with the information about its face parts and their arrangements. We show here that the learning of face stimuli makes the functional connection between these IT networks, causing synchronous firing of IT neurons. The model seems to be compatible with experimental data about dynamic properties of IT neurons.

1 Introduction

We can recognize rapidly and effortlessly complex visual scenes. Such amazing ability in visual recognition needs the effective processing of visual information along the multiple stages of visual pathways. Neurophysiological experiments have provided evidence for a “simple-to-complex” processing model based on a hierarchy of increasing complex image features. It has been also shown that inferior temporal (IT) cortex is shape-selective [1,2]. However, little is known about how the information about a whole object and its parts are represented in IT cortex. To address this issue, we focus on the neural mechanism of face recognition. It is a good model system for studying the neural mechanism of object recognition, because face is a whole image that consists of its parts such as eyes, nose, and mouth.

Recent experiments have reported dynamic responses of IT neurons in face perception. Hirabayashi and Miyashita [3] demonstrated that the IT neurons exhibit different responses in spike synchronicity for face and non-face images. Face-like objects elicited neuronal activities that were more correlated than the corresponding non-face like objects, whereas the firing rates of the same neurons did not show a significant difference. Sugase et al. [4] found that single IT neurons conveyed two

different scales of facial information in their firing patterns, starting at different latencies. They showed from the statistical analysis for spikes of IT neurons that global information about categorization of visual stimuli was conveyed in the early part of the responses, and fine information about identity and expression was conveyed later. However, less is understood the neural mechanism underlying the dynamic properties of IT neurons in face perception.

In the present study, we present a model of IT cortex, which performs face perception via an interaction between three IT layers and face recognition layer (FRL). The three IT layers encode the information about face features processed by V4 maps with different spatial resolutions. We show using the computational model that the spike synchronicity between the IT layers and FRL is needed to make functional connection between these layers by the Hebbian learning. After the learning, the information about face parts and the whole face are combined by the synchronous firing of these layers, leading to a face perception. The feedback from FRL to IT layers allows monkey to perform “global-to-fine” perception of faces.

2 Model

The network model of face perception is shown in Fig. 1. The model consists of three IT layers and face recognition layer (FRL). ITX (X=B, M, F) receive the information of face image processed by three V4 layers with different spatial resolutions. The ITB encodes the outline of retinal image with a broad spatial resolution. The ITM encodes the essential features of face parts and their arrangements, which is represented by V4M with a middle resolution. The ITF encodes the fine features of face parts such as eyes, nose, and mouth, by binding the information of fine features encoded by V4F. The FRL combines the information of three IT layers, and represents a unique image of face stimulus to perform the identification of faces.

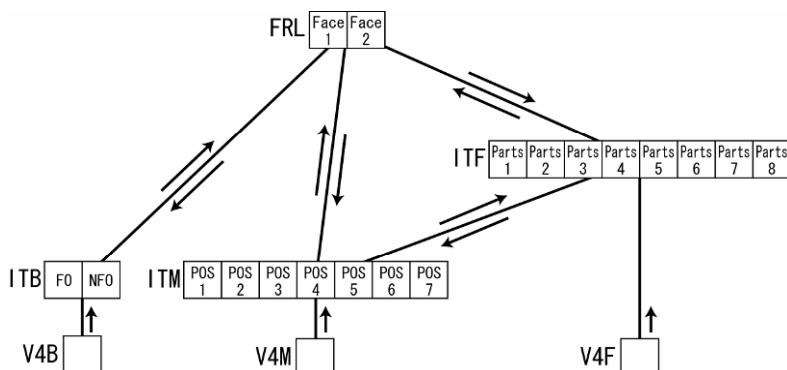


Fig. 1. The neural network model of IT cortex. The ITX (X=B, M, F) receive the outputs of V4X with different spatial resolutions. The “B”, “M”, and “F” mean the broad, middle, and fine resolutions, respectively. FRL(Face recognition layer) has the neurons tuned to different faces.

The neurons of IT layers and FRL were modeled with Hodgkin-Huxley model [5], because we focus on the neural mechanism underlying the spike synchrony between these layers. The membrane potential of i th neuron in ITX layer ($X=B, M, F$), $V_{i,ITX}$, is determined by

$$C_m \frac{dV_{i,ITX}}{dt} = -g_{Na}m^3h(V_{i,ITX} - V_{Na}) - g_Kn^4(V_{i,ITX} - V_K) - g_L(V_{i,ITX} - V_L) + I_{FF} + I_{FB}, \quad (1)$$

where C_m is the membrane capacitance. g_z ($z=Na, K, Cl$) are the channel conductance of ion z , and m , n , and h are the probability with which the channel is in the open state. I_{FF} and I_{FB} are the feedforward and feedback currents, respectively. These currents were calculated with synaptic currents described by α -function. The membrane potentials of FRL neurons were also determined by the Hodgkin-Huxley equation similar to Eq. (1).

The weights of synaptic connection between ITX and V4X ($X=B, M, F$) were learned with Kohonen's self-organized maps[6]. After the learning, the feature detective neurons responding to outline of faces were organized in ITB layer, the neurons responding to the arrangement of face parts were done in ITM layer, and those responding to the face parts such as eyes, nose, and mouth were done in ITF layer.

The neurons in three IT layers and FRL are reciprocally connected with synapses, as shown in Fig. 1. The synaptic connection between ITM and ITF and the connection between FRL and ITX ($X=B, M, F$) were made based on Hebbian learning. Then the weight of synaptic connection between i th neuron in X layer and j th neuron in Y layer, $w_{ij,XY}^Z(t)$, is determined by

$$\tau_w \frac{dw_{ij,XY}^Z}{dt} = -w_{ij,XY}^Z + \eta S_{iX} S_{jY}, \quad (2)$$

$$S_U = \frac{1}{1 + \exp(-(V_U - V_U^{th})/\epsilon)}, \quad (U = iX, jY), \quad (3)$$

where S_{iX} and S_{jY} are the neural activities in i th neuron of X layer and j th neuron of Y layer, respectively. $Z=FF, FB$, where FF and FB means the feedforward and feedback connections, respectively.

In the learning period of face, face stimuli were frequently presented, while non-face stimuli with random arrangement of face parts were rarely presented. For we encounter various faces every day, but we do rarely faces with random arrangement of their parts. The duration of stimulus presentation was 1500 msec, and the time interval between the current presentation and the next one was set to be 500 msec.

3 Results

3.1 Emergence of Spike Synchronization between the Layers Involved in Face Perception

Figure 2 shows the temporal variations of spiking of ITM and ITF neurons during the Hebbian learning. The spike synchronicity between ITM and ITF neurons in response to face stimulus gradually increases as the learning proceeds, whereas that in response to non-face stimulus does not increase. The difference in the synchronicity comes from the frequency of stimulus presentation. The frequent presentation of face stimuli facilitates Hebbian learning of synaptic connection between ITM and ITF neurons, as shown in Fig. 3, because the frequent presentation causes the increased chance of coincident spiking of ITM and ITF neurons. The spike timing of both neurons becomes closer as the number of coincident spikes are increased, resulting in the

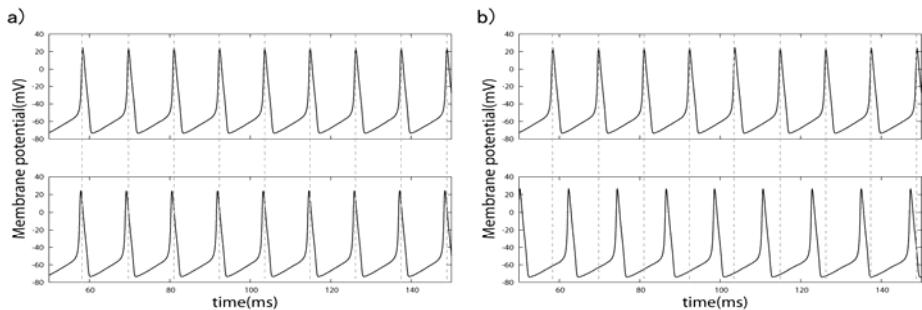


Fig. 2. Temporal variations of spiking of a pair of ITM and ITF neurons. Responses of these neurons to a face stimulus (a) and a non-face stimulus (b).

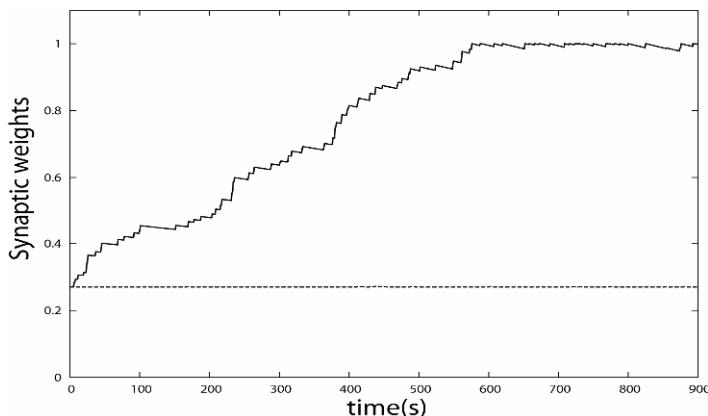


Fig. 3. Temporal variations of the weights of synaptic connection from ITM to ITF neurons. The solid and dashed lines represent the synaptic weights in the application of a face stimulus and a non-face stimulus, respectively.

increase of the learning efficacy of synaptic weights. That is, the coincident spiking causes the increased efficacy of the learning, and the increased synaptic weight facilitates synchronous firing of ITM and ITF neurons. Thus, the synaptic change induced by the coincident spiking and the synchronous firing interact with a positive feedback loop. In contrast, the presentation of non-face stimuli does not facilitate the synaptic learning, as shown in Fig. 3, because infrequent presentation of non-face stimulus causes little chance of coincident spiking of ITM and ITF neurons. The spike synchronicity between ITM and ITF neurons is clearly shown by the cross-correlations of spike intervals shown in Fig. 4.

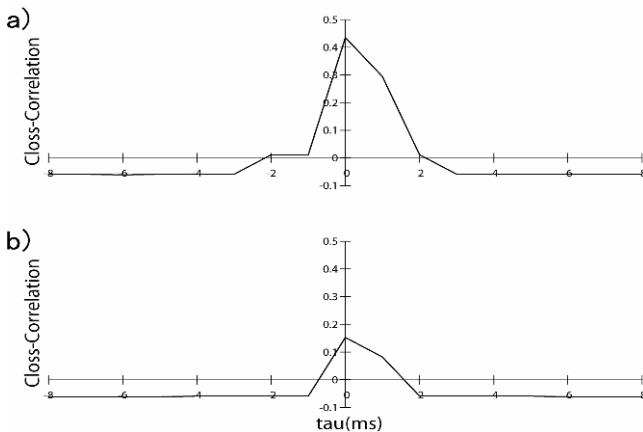


Fig. 4. Cross-correlations of spikes of a pair of ITM and ITF neurons. They were calculated for face stimulus (a) and non-face stimulus (b), respectively.

3.2 Synchronous Firing of ITF Neurons in Face Perception

Figure 5 shows the firing properties of ITF neurons during face perception after the learning. The presentation of face stimulus causes the synchronous firing between ITF neurons encoding face parts, as shown in Fig. 5a. Although these ITF neurons are not directly connected with each other, the connections of them with ITM and FRL neurons allow the ITF neurons to fire synchronously. In contrast, the presentation of non-face stimulus does not cause the synchronous firing between these ITF neurons, as shown in Fig. 5b. This is due to the absence of the connections of the ITF neurons with ITM and FRL neurons. The firing rates of the ITF neurons are almost the same in the responses to face and non-face stimuli, as shown in Fig. 5c. These results are in agreement with the results by Hirabayashi and Miyashita [3].

3.3 Role of ITB Layer in Face Perception

Figure 6 shows the responses of FRL neurons to face 1 stimulus. In the early period of face perception, the FRL neuron tuned to face 1 and that tuned to face 2 have the increased firing rates, as shown in Fig. 6, because the FRL neurons receive the information about the broad image of face 1 encoded by ITB layer. After the early

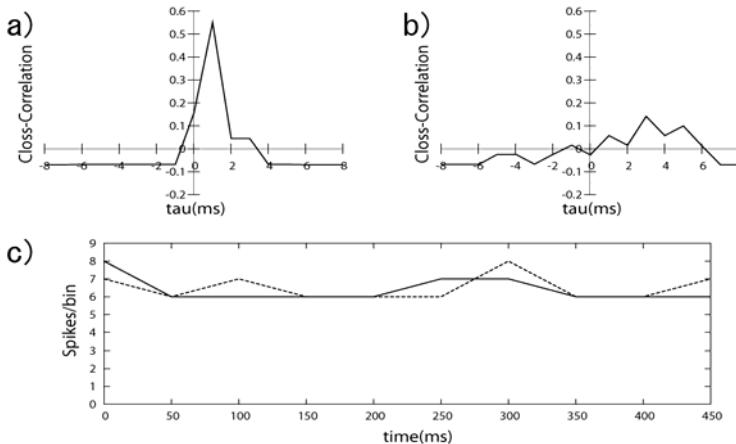


Fig. 5. Response properties for a pair of ITM neurons encoding the features of eye and nose. Cross-correlations of the spikes of these neurons in response to face stimulus (a) and non-face stimulus (b). (c) Firing rate of these neurons for face and non-face stimuli. The solid and dashed lines indicate the firing rates for face and non-face stimuli, respectively.

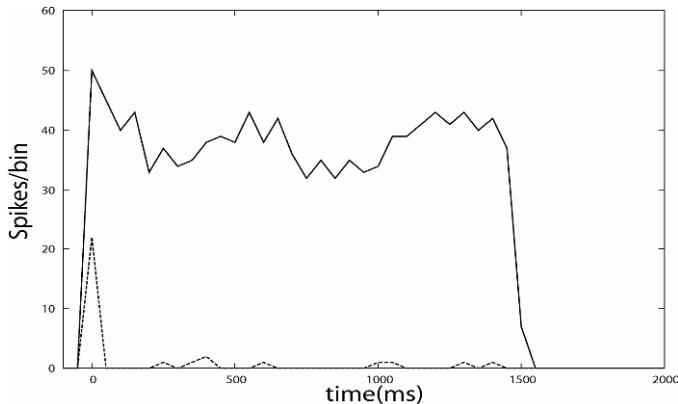


Fig. 6. Temporal variations of FRL neurons tuned to face 1 and face 2. The solid and dashed lines indicate the firing rates of FRL neurons in response to face1 and face2 stimuli, respectively.

period, the FRL neuron tuned to face 1 keeps the firing rate, while the FRL neuron tuned to face 2 decreases the firing rate, and then becomes silent, as shown in Fig. 6. The difference in the firing rates of FRL neurons comes from the binding of the information of top-down signal and the information about the details of face1 encoded by ITM and ITF. The result is consistent with the result by Sugase et al. [4].

Thus ITB sends broad but fast information about face stimulus to FRL, sending a top down signal from FRL to ITF and ITM and giving the information about possible faces. Then the refinement of the relevant face is made by the combination of the top-down signal and the feedforward signals received by ITM and ITF neurons.

4 Conclusion

In the present study, we have presented a model of IT cortex, which performs face perception via an interaction between the three IT layers and FRL. The spike synchrony, induced by the learning due to coincident spiking, is needed to make the functional connection between the IT layers and FRL. After the learning, the information about face parts and the whole face are combined by the synchronous firing of the IT neurons. Furthermore, “global-to-local” processing of face information could be mediated by the top-down signal from FRL to ITM and ITF layers, elicited by the fast feedforward signal from ITB to FRL. Further studies are required to understand the mechanism underlying efficient processing of information about various faces in IT cortex.

References

1. Logothetis, N.K., Sheinberg, D.L.: Visual object recognition. *Annu. Rev. Neurosci.* 19, 577–621 (1996)
2. Tanaka, K.: Inferotemporal cortex and object vision. *Annu. Rev. Neurosci.* 19, 109–139 (1996)
3. Hirabayashi, T., Miyashita, Y.: Dynamically modulated spike correlation in monkey inferior temporal cortex depending on the feature configuration within a whole object. *J. Neurosci.* 25, 10299–10307 (2005)
4. Sugase, Y., Yamane, S., Ueno, S., Kawano, K.: Global and fine information coded by single neurons in the temporal visual cortex. *Nature* 400, 870–873 (1999)
5. Koch, C.: *Biophysics of Computation*. Oxford Univ. Press, New York (1999)
6. Kohonen, T.: *Self-organized Maps*. Springer, Berlin (1995)

Part IV

Bioinformatics

Clustering of Spectral Patterns Based on EMD Components of EEG Channels with Applications to Neurophysiological Signals Separation

Tomasz M. Rutkowski¹, Andrzej Cichocki¹,
Toshihisa Tanaka^{2,1}, Anca L. Ralescu³, and Danilo P. Mandic⁴

¹ RIKEN Brain Science Institute, Wako-shi, Saitama, Japan
tomek@brain.riken.jp
<http://www.bsp.brain.riken.jp/>

² Tokyo University of Agriculture and Technology, Tokyo, Japan
tanakat@cc.tuat.ac.jp

³ Computer Science Department, University of Cincinnati, Ohio, USA
anca.ralescu@uc.edu
<http://www.ececs.uc.edu/~aralescu/>

⁴ Department of Electrical and Electronic Engineering, Imperial College London, UK
d.mandic@imperial.ac.uk
<http://www.commsp.ee.ic.ac.uk/~mandic/>

Abstract. The notion of information separation in electrophysiological recordings is discussed. Whereas this problem is not entirely new, a novel approach to separate muscular interference from brain electrical activity observed in form of EEG is presented. The EEG carries brain activity in form of neurophysiological components which are usually embedded in much higher in power electrical muscle activity components (EMG, EOG, etc.). A novel multichannel EEG analysis approach is proposed in order to discover representative components related to muscular activity which are not related to ongoing brain activity but carry common patterns resulting from non-brain related sources. The proposed adaptive decomposition approach is also able to separate signals occupying same frequency bands what is usually not possible with contemporary methods.

1 Introduction

A problem of separation electrical brain activity, captured in form of EEG, from the remaining electric signals originating in human bodies (EOG, EMG, ECG, etc.) including the other surrounding interference (electrical devices in close proximity, etc.) is an obstacle of many experiments related to neuroscience. Those problems are often addressed in papers presenting contemporary findings of engineering approaches to neuroscience [123].

In the presented paper experiments were conducted in Advanced Brain Signal Processing Laboratory of RIKEN Brain Science Institute, Japan. Every subject was asked to wear an EEG cap with electrodes connected to the following front

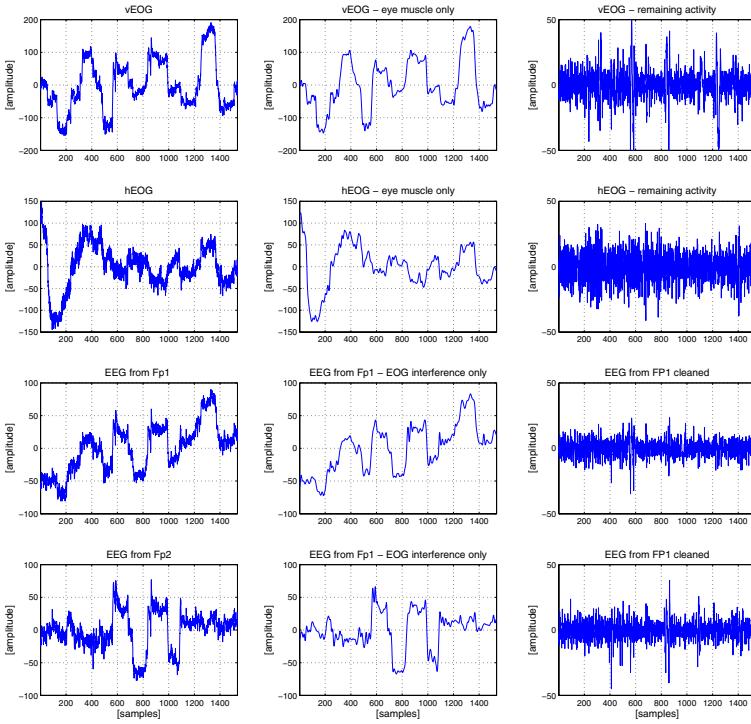


Fig. 1. An illustration of the muscular interference problems in EEG. The right column presents from the top two EOG reference recordings capturing vertical (*vEOG*) and horizontal (*hEOG*) recorded simultaneously with following two EEG front head channels (*Fp1* and *Fp2*). The high amplitude of eye movements interference is very simple to grasp within EEG channels as well their complex mixture. The results of the proposed approach (see Section 2) of EEG and muscular artifacts separation are presented in the middle and right columns of the above picture. The middle column presents “clean” muscle activity, which is an expected result for EOG and an interference in case of EEG. The right column presents the only electrophysiological activity. The spectral analysis of the above signals is presented in Fig. 2.

head channels *Fp1*, *Fp2* as in extended 10/20 EEG recording systems [4]. Additionally two ocular channels *vEOG*, *hEOG* to capture vertical and horizontal eye movements were recorded respectively. As stimuli surround sound stimuli in 7.1 speaker system environment was presented and subjects were asked to try to localize the direction. The paradigm caused often subjects to move their eyes involuntary so the eye movement interference caused by the muscles became a significant obstacle. This paper addresses a problem of ocular artifacts removal from frontal head EEG channels.

The aim of this paper is therefore twofold *as follows*: (i) to introduce a novel computational framework based on Empirical Mode Decomposition (EMD) convenient for simultaneous data conditioning and information separation within

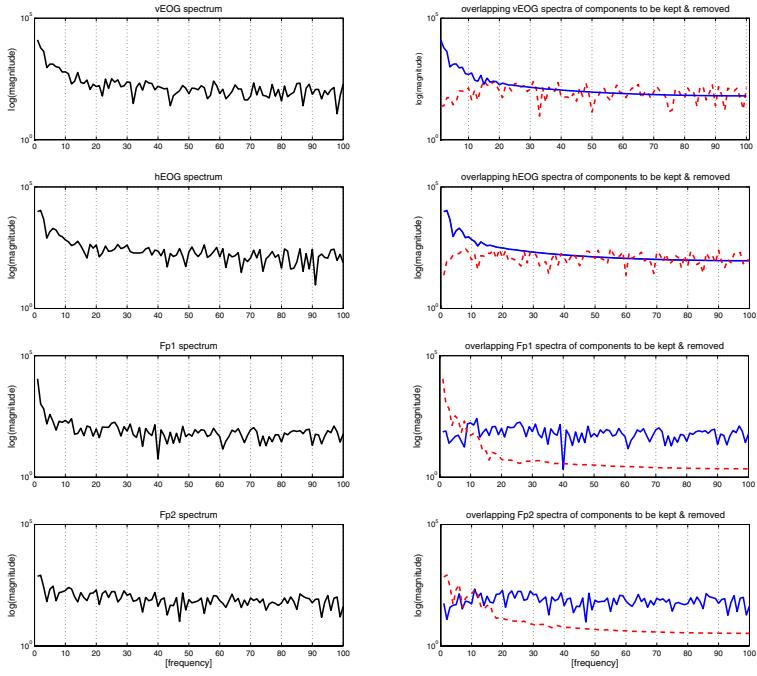


Fig. 2. A comparison of original spectra of EOG and EEG channels is presented in left column of the above figure. The right column presents the separated signals within each channel with utilization of the method described in Section 2. The red/dotted-line spectra represent interferences which were removed from the signals while blue/solid-lines represent the remaining signals of interest (pure eye muscle activity for EOG channels and neurophysiological activity only for EEG). It is interesting to note that the proposed approach with EMD analysis let us keep underlying EEG activity in the frequency bands dominated by EOG, what would be not possible to achieve with classical filtering methods - the whole band would be classified as interference.

the captured EEG data, and (ii) to provide an insight into possible clustering of different electrical activities originating in different sources of human body, in particular, to separate eye movements from neurophysiological signals.

2 Methods

EEG signals recorded on the scalp levels are usually highly contaminated by noise due to a very low level of neurophysiological signals and due to higher power of general electrophysiological signals, such as EMG or EOG, as well as the presence of different devices in the recording environment which cause electromagnetic interference. To tackle these problems, the current study proposes to use empirical mode decomposition (EMD) [5], a new technique to decompose

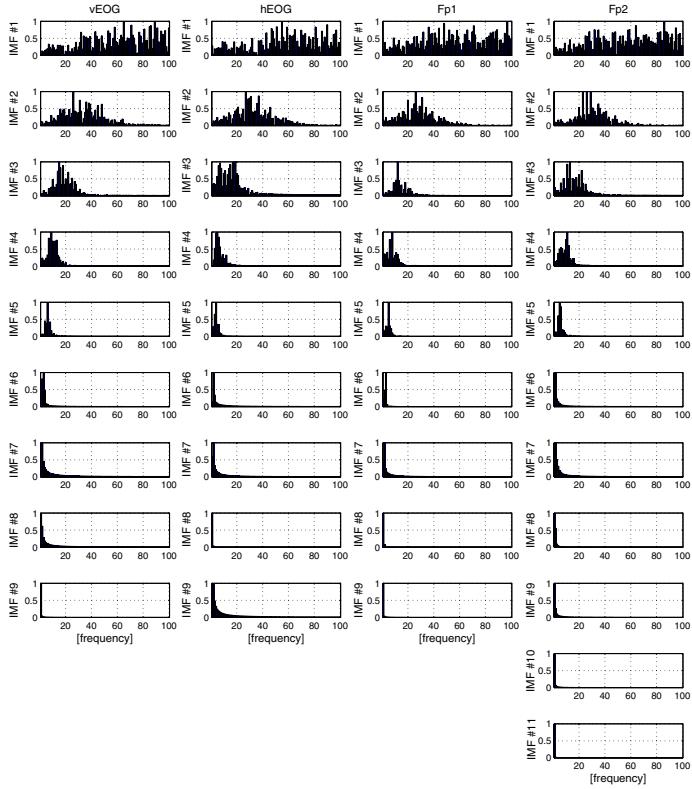


Fig. 3. Spectral features from all IMFs resulting from EMD analysis. The spectra were normalized for each IMF for visualization purposes. It is interesting to note that several IMFs from a single EEG channel (a single column in the above figure) occupy often same frequency bands (not possible to achieve with contemporary bandpass filtering methods). The presented vectors of spectral features were further utilized for cross-correlation and clustering analysis in order to identify common patterns resulting from EOG interference.

EEG signals. Once the components from each channel are obtained we aim to compare them in spectral domain in order to separate common activities propagated across the human scalp from local neurophysiological ones.

2.1 Empirical Mode Decomposition (EMD) in Application to EEG

EMD utilizes empirical knowledge of oscillations intrinsic to a signal in order to represent them as a superposition of components, defined as *intrinsic mode functions* (IMF), with well defined instantaneous frequencies. To obtain an IMF from a single channel EEG it is necessary to remove local riding waves and asymmetries, which are estimated from local envelope of minima and maxima of the

waveform. The technique of finding IMFs corresponds thus to separation of band limited semi-orthogonal components from recorded EEG. It also corresponds to eliminating riding-waves from the signal, which ensures that the instantaneous frequency will have no fluctuations caused by an asymmetric wave form. In each cycle, the IMF is defined by zero crossings and involves only one mode of oscillation, thus not allowing complex riding waves. Notice that an IMF is not limited to be a narrow band signal, as it would be in classical Fourier or wavelets decompositions. In fact, an IMF can be both amplitude and frequency modulated simultaneously, as well as non-stationary or non-linear.

EMD decomposes a signal in hand into IMFs [5] and represents in form of “oscillatory modes” which satisfy the following two conditions:

- (i) the number of extrema and the number of zero crossings are either equal or differ at most by one;
- (ii) at any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero.

Since IMF represents an oscillatory mode within a signal; its periods, which are defined by zero crossings, correspond to the only *one* mode of oscillation. Both the amplitude and frequency of this oscillation may vary over time, in other words, the oscillation is not necessarily stationary nor narrow-band.

The process of extracting an IMF from a signal $x(t)$ is called “the sifting process” [5] and consists of the following steps:

1. determine the local maxima and minima of $x(t)$;
2. generate the upper and lower signal envelope by connecting those local maxima and minima respectively by an interpolation method (e.g., linear, spline, piece-wise spline [5,6]) (in this paper the linear method was chosen);
3. determine the local mean $m_1(t)$, by averaging the upper and lower signal envelope;
4. subtract the local mean from the data: $h_1(t) = x(t) - m_1(t)$.

Ideally, $h_1(t)$ satisfies the conditions (i) and (ii) of an IMF, however, typically this procedure needs to be repeated until the first IMF is extracted. In order to obtain the second IMF we applied the sifting process to the residue $\epsilon_1(t) = x(t) - \text{IMF}_1(t)$, obtained by subtracting the first IMF from $x(t)$; the third IMF is in turn extracted from the residue $\epsilon_2(t)$ and so on. The decomposition is complete when two consecutive sifting results are similar; the empirical mode decomposition of the signal $x(t)$ may be written as:

$$x(t) = \sum_{k=1}^n \text{IMF}_k(t) + \epsilon_n(t), \quad (1)$$

where n is the number of extracted IMFs, and the final residue $\epsilon_n(t)$ is either the mean trend or a constant. Fig. 3 shows an example of a spectral analysis of EMD applied to an EEG signal; observe that IMFs across different channels represent different oscillatory modes within the signal. Note that the IMFs are

Result of the cluster analysis in form of a cluster dendrogram

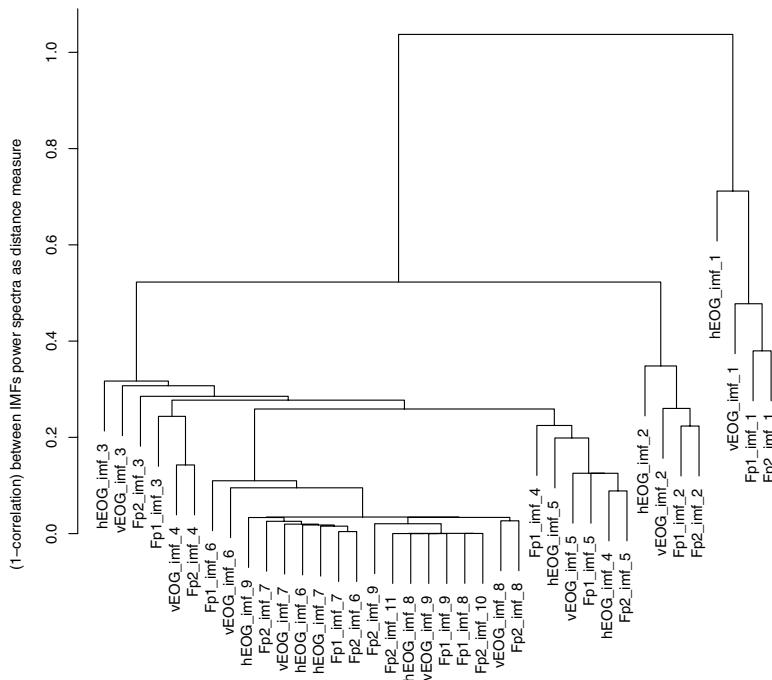


Fig. 4. IMF spectral distances cluster analysis results revealing similarity among several components from different channels. Note the three groups of clusters resulting from cross-correlation analysis for IMFs spectral features. In the presented paper the components having spectral distances lower than 0.1 to the similar components in other electrodes were considered as common EOG activity. The resulting reconstruction of neurophysiological signals and “clean” EOG is presented in right column of Fig. □ The diagram was created with [8].

not guaranteed to be mutually orthogonal, but are often close to orthogonal; note also noteworthy that IMFs are adaptive, that is, two independent realizations of a signal with the same statistics may have a different number of IMFs.

2.2 Spectral Clustering of EMD Components

In order to compare all IMFs extracted from the analyzed channels (two EOG and two EEG in this paper) we propose to transform them to Fourier domain in order to capture spectral content carried by all of them. The power spectra of all IMFs (note that adaptive nature of EMD may result in different numbers of IMFs in each channel) are presented in form of normalized vectors in Fig. 3. This figure also captures the strength of EMD allowing us to extract signal subcomponents occupying same frequency bands which would be not possible

with classical time–frequency analysis methods. The spectra are further treated as features and compared for their similarity.

We use correlations between variables as “a distance measure” in order to capture spectral similarity across the IMFs. Once cross–correlation analysis is performed for all IMFs from all analyzed channels a hierarchical cluster analysis using a set of dissimilarities for the n objects to be clustered is performed [7] with utilization of “R” package [8]. Initially, each vector representing power spectrum values is assigned to its own cluster and then the algorithm proceeds iteratively, at each stage joining the two most similar clusters. Such procedure continues until there is just a single cluster. At each stage distances between clusters are recomputed by the Lance–Williams dissimilarity update formula with a single linkage method clustering method. The single linkage method is closely related to the minimal spanning tree concept and it adopts a “friends of friends” strategy for clustering [7]. A result of such procedure is presented in Fig. 4 where two sets of clusters are visualized. The first set is for distances below 0.1 and those components from different channels are classified as similar and originating from very strong EOG interference. The reconstruction (summation) for those IMFs is presented in the middle panel of Fig. 4 and it vividly confirms the hypothesis. The remaining set of clusters with distances above 0.1 represent IMFs carrying neurophysiological signals only and the resulting reconstruction is presented in the right panel of Fig. 4. Spectral analysis of the proposed separation is visualized in Fig. 2. Note that the proposed method saves brain–activity–related low frequency spectral content of neurphysiological signals under very strong EOG interference (see the range 0Hz to about 10Hz which is not completely “zeroed,” what would happen with classical *highpass* filtering approaches).

3 Conclusions

A framework to separate brain activity from interfering muscle activity in EEG has been presented. This has been achieved by proposing a new EEG decomposition technique, which allows a flexible sub-band signal decomposition while preserving the nonlinear and non-stationary nature of the signals. The so obtained components from each EEG channel have been compared in spectral domain analyzed with clustering technique in order to identify those similar across channels. The resulting reconstruction has allowed us to separate common muscular interferences from underlying brain activity.

This is a step forward toward creating user friendly brain–machine–interfaces that would be free for common interferences resulting from human body activity.

Acknowledgment. This work was supported in part by AOARD grant “Advanced Brain Signal Processing for the Study of Sleep Inertia” and by JSPS and Royal Society under the Japan-UK Research Cooperative Program 2007 & 2008.

References

1. Joyce, C.A., Gorodnitsky, I.F., Kutas, M.: Automatic removal of eye movement and blink artifacts from eeg data using blind component separation. *Psychophysiology* 41(2), 313–325 (2004)
2. Rutkowski, T.M., Cichocki, A., Ralescu, A.L., Mandic, D.P.: Emotional states estimation from multichannel EEG maps. In: Wang, R., Gu, F., Shen, E. (eds.) *Advances in Cognitive Neurodynamics: Proceedings of the International Conference on Cognitive Neurodynamics 2007*. Neuroscience, pp. 695–698. Springer, Heidelberg (2008)
3. Looney, D., Li, L., Rutkowski, T.M., Mandic, D.P., Cichocki, A.: Ocular artifacts removal from EEG using EMD. In: Wang, R., Gu, F., Shen, E. (eds.) *Advances in Cognitive Neurodynamics: Proceedings of the International Conference on Cognitive Neurodynamics 2007*, pp. 831–835. Springer, Heidelberg (2008)
4. Niedermeyer, E., Da Silva, F.L. (eds.): *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*, 5th edn. Lippincott Williams & Wilkins (2004)
5. Huang, N., Shen, Z., Long, S., Wu, M., Shih, H., Zheng, Q., Yen, N.C., Tung, C., Liu, H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 454(1971), 903–995 (1998)
6. Rutkowski, T.M., Cichocki, A., Mandic, D.P.: Information Fusion for Perceptual Feedback: A Brain Activity Sonification Approach. In: *Signal Processing Techniques for Knowledge Extraction and Information Fusion. Signals and Communication*, pp. 261–274. Springer, Heidelberg (2008)
7. Murtagh, F.: Multidimensional clustering algorithms. *COMPSTAT Lectures* 4 (1985)
8. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008),
<http://www.R-project.org>

Consensus Clustering Using Spectral Theory

Mariá Cristina Vasconcelos Nascimento, Franklinia Maria Bragion de Toledo,
and André C. Ponce Leon Ferreira Carvalho

Instituto de Ciências Matemáticas e de Computação (ICMC),
Universidade de São Paulo (USP) – São Carlos, SP, Brasil
`{mariah,fran,andre}@icmc.usp.br`

Abstract. Consensus clustering is a well studied methodology to find partitions through the combination of different formulations or clustering partitions. Different approaches for dealing with this issue using graph clustering have been proposed. Additionally, strategies to find consensus partitions by using graph-based ensemble algorithms have attracted a good deal of attention lately. A particular class of graph clustering algorithms based on spectral theory, named spectral clustering algorithms, has been successfully used in several clustering applications. However, in spite of this, few ensemble approaches based on spectral theory has been investigated. This paper proposes a consensus clustering algorithm based on spectral clustering. Preliminary results presented in this paper show the good potential of the proposed approach.

1 Introduction

Clustering is an unsupervised technique that looks for groups of similar objects in a dataset. There are several approaches for clustering algorithms, which might differ in several important aspects and be derived by different methodologies. Clustering algorithms may either explore well founded concepts by using mathematical and statistical theories or rely on heuristics. The latter represents the majority of the most popular algorithms.

A group of clustering algorithms, known as graph clustering, deals with clustering algorithms based on graph theory. According to [1], due to the lack of a precise mathematical formulation to define a distinct set of concepts on cluster analysis, a formal study of the methodologies is necessary. Graph theory in cluster analysis tries to provide the required mathematical foundation. Moreover, graph clustering allows clusters with different shapes, which is useful in several applications. Some examples of graph clustering are the algorithms based on the node coloring and minimum spanning tree principles [23], on the node partitioning problem [45], on network flows [6] and on spectral theory [78].

Due to the different clustering criteria adopted, each clustering algorithm has its bias, therefore being more adequate for particular data conformations. Even the same algorithm may produce different partitions for different runs. Thus, it is not easy to define what is the most appropriate algorithm for a given dataset. An alternative to reduce this limitation is the combination of different

partitions using ensemble. Ensemble or consensus approaches are very popular in classification problems and have been successfully applied in the last years to clustering applications.

This paper proposes and investigates a consensus clustering algorithm that combines a set of partitions from the *k-means* algorithm [9] and from the *unnormalized spectral clustering* algorithm [10]. The proposed algorithm builds an incidence matrix, which stores the number of times that each two objects belong to the same cluster considering every partition. A Laplacian graph of random walks is used in the *unnormalized spectral clustering* algorithm proposed in [11].

This paper is organized as follows. Section 2 presents the main concepts of clustering and clustering ensemble. Section 3 describes the proposed ensemble algorithm with two variations. Section 4 has the algorithms, the validation measure and the datasets used in the experiments, as well as the settings used for the proposed ensemble. Section 5 reports the results of experiments, whereas Section 6 concludes the paper.

2 Clustering

Clustering algorithms are widely used to find meaningful groups of objects in a dataset when no or little prior information is available [9]. Several clustering algorithms have been proposed in the literature, each looking for clusters according to a different criterion [12]. The criterion will directly influence the shape and nature of the clusters found. Besides, different similarity measures will lead to different partitions.

Clustering algorithms can be organized according to different aspects, like *crisp clustering*, if each object either belongs or does not belong to a given cluster; and *fuzzy clustering*, if each object belongs to each cluster in conformity with a membership degree; *hierarchical clustering* algorithms, which organize partitions in a hierarchical structure; and *partitioning clustering* algorithms that form a partition with empty intersection between any two clusters. All these variations have been applied to various applications, including gene expression analysis [13], image segmentation [14], natural language processing [15] and protein sequence multiple alignment [16]. In this work we will be concerned with crisp partitioning clustering.

Clustering algorithms may find partitions at different refinement levels (different numbers of clusters or clusters densities), depending on their parameter settings [9]. These characteristics make parameter tuning of clustering algorithms usually more complex than the parameter tuning of classification algorithms. A few validation indexes have been proposed in the literature to evaluate the quality of partitions. However, they are usually biased towards the clustering criterion adopted.

It must be observed that each algorithm looks for an homogeneous structure (all clusters are in accordance with the same cluster definition), while data can present an heterogeneous structure (each cluster conforms to a different clustering criterion) [12].

Every clustering algorithm has its bias. Each algorithm has its advantages and limitations. The selection of the best algorithm for a given dataset is not a trivial task. Even the same algorithm may produce different partitions in different runs. In order to reduce the uncertainty regarding the selection of a clustering algorithm and have final partitions with a more stable quality, cluster ensemble techniques have been investigated in the last years.

Ensemble or consensus clustering consists of combining different clustering partitions in order to obtain a final diversified partition of high quality, according to the validation index adopted. The base partitions usually are either produced by different runs of the same clustering algorithm, using the same or different parameter values, named homogeneous ensemble, or originated from different clustering algorithms, named heterogeneous ensemble.

A cluster ensemble algorithm usually can be divided into two phases: the generation of a diverse set of individual (base) partitions and the combination of these partitions in a consensus partition. Several different strategies can be followed in these two steps [17][18][19][20][21][22][23][24].

3 Proposed Algorithm

The proposed consensus clustering is based on the spectral clustering algorithm presented in [1], which uses the Laplacian graph of random walks. It consists of the relaxation of the NCut problem associated with the use of the incidence matrix instead of the distance matrix. In order to explain how the algorithm works, let n be the number of objects of the dataset and m be the number of clusters to be formed. The proposed algorithm consists of the following steps.

1. Obtain a set of initial clustering partitions.
2. Build an incidence matrix $A = [a_{ij}]_{n \times m}$ according to the number of times that the objects i and j belong to the same cluster (considering every partition from Step 1).
3. Calculate the matrix $P = [p_{ij}]_{n \times n}$ such that $P = D^{-1}W$ and $D = [d_{ij}]_{n \times n}$ is the diagonal matrix with $d_{ii} = \sum_{j=1}^n a_{ij}$ for $1 \leq i \leq n$.
4. Find the random walks Laplacian matrix $L = [l_{ij}]_{n \times n}$, with $L = I - P$, where I is the identity matrix.
5. Find the m least eigenvalues of the Laplacian matrix and their corresponding eigenvectors.
6. Build the matrix $S_{n \times m}$ with the first eigenvectors settled in increasing order of eigenvalues on its columns.
7. Apply the kmeans algorithm in the matrix S in order to obtain the final partition with m clusters.

We propose two variations for our clustering ensemble algorithm. In our first variation, named Ensemble kmeans algorithm (Ekmeans), we use different partitions from the kmeans algorithm in order to obtain the initial set of clustering partitions. In the second variation, Ensemble Ukmeans algorithm (EUKmeans), we use distinct partitions from the unnormalized spectral kmeans algorithm to compose the initial partitions.

4 Experiments

This section presents the algorithms investigated and the experiments carried out in order to investigate the proposed approach. In the experiments, we used ten datasets, which are summarized on Table 1. The last column $\#Clusters$ indicate the number of clusters in each of known structures of each dataset. The MiRNA dataset has two possible 2-classes classifications. The use of the number of classes in a dataset for cluster analysis is a common approach to evaluate cluster algorithms.

Table 1. Main characteristics of the datasets

Type	Name	#Objects	#Attributes	#Clusters
Biological	Protein [25]	698	125	4,27
	Breast [26]	699	9	2
	Novartis [27][28]	103	1000	4
	Leukemia [29]	327	271	7
	Lung [30]	197	1000	4
	MiRNA [31]	218	217	2,3,4,9,20
Benchmark	Ecoli [32]	336	7	8
	Iris [33]	150	4	3
	Glass [34]	214	9	6

In the evaluation of the proposed ensemble variations, we compared them with the following clustering algorithms: kmeans, kmedians, Ukmeans (unnormalized spectral kmeans algorithm), PAM (Partition around medoids) and Cluto. These comparisons were based on the Corrected Rand index (CRand) proposed in [35]. We used the kmeans, kmedians and PAM implementations from the R-project from the package *cluster* and Cluto from the implementation reported in [36].

For each algorithm, the number of clusters must be defined a priori. The kmeans algorithm searches for the centroids of the partition that minimize the distance between each objects and its nearest centroid. The kmedians algorithm looks for the median of each cluster, medoids, for a similar minimization. These two algorithms randomly creates the initial partition, leading to a distinct final partition for each run. Each one of these algorithms was run 30 times and the partition with the minimum within-cluster distance was adopted.

PAM is a deterministic algorithm based on the minimization of the dissimilarities between each object and the medoid of its cluster. It is divided into two stages: (1) define a set of initial medoids; (2) fine tunes the medoids by swapping objects between the clusters [37].

Ukmeans is a spectral clustering algorithm that looks for the Laplacian graph of a similarity matrix to produce a partition. The Laplacian graph is defined by the matrix $L = D - W$, where D is the diagonal matrix with $d_{ii} = \sum_{j=1}^n a_{ij}$ for $1 \leq i \leq n$, and $A = [a_{ij}]_{n \times n}$ is defined as the adjacency matrix of the 15 nearest neighbor graph evaluated by the Euclidean distance. To use Ukmeans, it is necessary to find the m least eigenvalues and their associated eigenvectors. A

matrix E_l is created with these m eigenvectors settled into columns in the eigenvalues increasing order. The kmeans algorithm clusters the E_l rows, producing the final partition.

Cluto is a multi-level partitioning algorithm based on the 15 nearest neighbor graph. The objective function maximizes, for each cluster, the sum of within cluster similarity divided by the number of objects in the cluster [36].

In order to evaluate the proposed ensembles, we used 3 partitions, each one with 2, 3 and 4 clusters to be combined, in both Ekmeans and Eukmeans. These choices as well as the number of neighbors in the k nearest neighbor graph were based on preliminary tests that showed more stability with these values. The results from the experiments are reported next.

5 Experimental Results

Table 2 shows the CRand values of every algorithm described in Section 4, including the proposed ensembles. The columns Dataset and m indicate, respectively, the tested dataset and the number of clusters considered, which is equal to the number of classes in the dataset. The columns *Clustering algorithms* and *Ensembles* show the CRand results for the investigated algorithms. We used the Euclidean distance metric as the distance measure between the objects.

According to the experimental results shown in Table 2, Ekmeans was the most successful alternative, with the highest CRand in 7 structures. These results were significantly better than the other algorithms in 3 structures: Protein with

Table 2. CRand index validation results for the individual algorithms and the proposed ensemble variations

Dataset	m	Clustering algorithms					Ensembles	
		Ukmeans	kmeans	kmedians	PAM	Cluto	Ekmeans	Eukmeans
Protein	4	0.186	0.232	0.185	0.203	0.274	0.234	0.159
Protein	27	0.119	0.135	0.130	0.090	0.100	0.220	0.195
Breast	2	0.834	0.828	0.782	0.828	0.818	0.834	0.828
Novartis	4	0.973	0.946	0.946	0.897	0.973	0.973	0.571
Leukemia	7	0.758	0.748	0.746	0.750	0.665	0.796	0.618
Lung	4	0.379	0.356	0.277	0.326	0.170	0.384	0.349
MiRNA	2	0.444	0.471	0.523	0.564	0.603	0.432	0.471
MiRNA	2	-0.038	0.095	0.063	0.028	-0.077	-0.035	0.095
MiRNA	3	0.091	0.160	0.160	0.155	0.199	0.210	0.090
MiRNA	4	0.401	0.415	0.419	0.430	0.193	0.401	0.618
MiRNA	9	0.043	0.041	0.064	0.060	0.018	0.020	0.173
MiRNA	20	0.326	0.371	0.376	0.379	0.307	0.318	0.313
Ecoli	8	0.303	0.427	0.414	0.409	0.34	0.389	0.514
Iris	3	0.745	0.730	0.744	0.730	0.732	0.745	0.730
Glass	6	0.214	0.179	0.211	0.141	0.174	0.203	0.139

27 clusters, Leukemia with 7 clusters and MiRNA with 3 clusters. Ekmeans was able to find different partitions for these structures.

The other variation proposed, Eukmeans, achieved the second best results together with Ukmeans. Both achieved the best CRand 4 times. Nevertheless, 3 of the Ukmeans best CRand values were similar to those obtained by Ekmeans, while only one of the Eukmeans best results was also obtained by another algorithm. Furthermore, Eukmeans achieved considerably higher CRand values in 3 structures: MiRNA with 4 and 9 clusters and Ecoli with 8 clusters. These results suggest that both proposed ensemble algorithm variations are promising alternatives for clustering problems.

6 Final Remarks

This paper proposed an ensemble algorithm based on graph spectral theory for clustering data. The proposed methodology is based on the study of the incidence matrix produced by the use of different partitions. The tendency of the clustering algorithm is to search for more stable partitions, since it places in the same cluster objects found together a large number of times. A bound for this problem could be found through the relaxation of the semi-definite NCut problem. In the use of this algorithm, initially, a graph of the 15 nearest neighbors of each dataset was found. The ensemble produced a combination of clustering partitions using two sets of partitions: 2, 3 and 4 cluster partitions from kmeans and 2, 3 and 4 cluster partitions of the unnormalized kmeans algorithm. A random walks Laplacian graph was derived to find the spectral clustering.

A set of experiments were carried out where the two ensembles were compared with 5 different clustering algorithms. In the experimental results, the proposed ensemble presented the best performance in finding the best partitions concerning the CRand validation index. The ensemble of kmeans partitions had the best results in virtually half of the 15 structures.

Acknowledgments

The authors would like to thank the support received from FAPESP and CNPq, and Dr. William H. Wolberg, from the University of Wisconsin Hospitals in Madison, for the breast cancer dataset.

References

1. Beliakov, G., King, M.: Density based fuzzy c-means clustering of non-convex patterns. European Journal of Operational Research 127(3), 717–728 (2006)
2. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers C-20, 68–86 (1971)
3. Xu, Y., Olman, V., Xu, D.: Minimum spanning trees for gene expression data clustering. Genome Informatics 12, 24–33 (2001)

4. Charon, I., Hudry, O.: Noising methods for a clique partitioning problem. *Discrete Appl. Math.* 154(5), 754–769 (2006)
5. Ji, X., Mitchell, J.E.: Branch-and-price-and-cut on the clique partitioning problem with minimum clique size requirement. *Discrete Optimization* 4, 87–102 (2007)
6. van Dongen, S.M.: Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht (2000)
7. Forman, J.J., Clemons, P.A., Schreiber, S.L., Haggarty, S.J.: Spectralnet—an application for spectral graph analysis and visualization. *BMC Bioinformatics* 6 (2005)
8. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. *Pattern Recognition* 41(1), 176–190 (2008)
9. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
10. Von Luxburg, U.: A tutorial on spectral clustering. *Statistics and Computing* 17(4), 395–416 (2007)
11. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis And Machine Intelligence* 22, 888–905 (2000)
12. Law, M.H.C., Topchy, A.P., Jain, A.K.: Multiobjective data clustering. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 424–430 (2004)
13. Higham, D.J., Kalna, G., Vass, J.K.: Spectral analysis of two-signed microarray expression data. *Mathematical Medicine and Biology* 24, 131–148 (2007)
14. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(11), 1101–1113 (1993)
15. Ushioda, A., Kawasaki, J.: Hierarchical clustering of words and application to nlp tasks. In: Ejerhed, E., Dagan, I. (eds.) Fourth Workshop on Very Large Corpora, Somerset, New Jersey, pp. 28–41. Association for Computational Linguistics (1996)
16. Krause, A., Stoye, J., Vingron, M.: Large scale hierarchical clustering of protein sequences. *BMC Bioinformatics* 6, 15 (2005)
17. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3, 583–617 (2002)
18. Greene, D., Tsymbal, A., Bolshakova, N., Cunningham, P.: Ensemble clustering in medical diagnostics. In: IEEE Symposium on Computer-Based Medical Systems, pp. 576–581 (2004)
19. Fern, X.Z., Brodley, C.A.: Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the 21 st International Conference on Machine Learning, Banff, Canada (2004)
20. Topchy, A.P., Law, M.H.C., Jain, A.K., Fred, A.L.: Analysis of consensus partition in cluster ensemble. *Icdm*, 225–232 (2004)
21. Topchy, A., Jain, A.K., Punch, W.: Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1866–1881 (2005)
22. Hadjitarov, S.T., Kuncheva, L.I., Todorova, L.P.: Moderate diversity for better cluster ensembles. *Inf. Fusion* 7(3), 264–275 (2006)
23. Hadjitarov, S.T., Kuncheva, L.I.: Selecting diversifying heuristics for cluster ensembles. In: Haindl, M., Kittler, J., Roli, F. (eds.) *MCS 2007. LNCS*, vol. 4472, pp. 200–209. Springer, Heidelberg (2007)
24. Riesen, K., Bunke, H.: Classifier ensembles for vector space embedding of graphs. In: Haindl, M., Kittler, J., Roli, F. (eds.) *MCS 2007. LNCS*, vol. 4472, pp. 220–230. Springer, Heidelberg (2007)

25. Ding, C.H.Q., Dubchak, I.: Multi-class protein fold classification using support vector machines and neural networks. *Bioinformatics* 17, 349–358 (2001)
26. Bandyopadhyay, S., Maulik, U.: An evolutionary technique based on k-means algorithm for optimal clustering in rn. *Information Sciences* 146(1-4), 221–237 (2002)
27. Su, A.I., Cooke, M.P., Ching, K.A., Hakak, Y., Walker, J.R., Wiltshire, T., Orth, A.P., Vega, R.G., Sapinoso, L.M., Moqrich, A., Patapoutian, A., Hampton, G.M., Schultz, P.G., Hogenesch, J.B.: Large-scale analysis of the human and mouse transcriptomes. *Proceedings of the National Academy of Sciences* 99, 4465–4470 (2002)
28. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Technical report, Broad Institute/MIT - Kluwer Academic Publishers, Dorchet (2003)
29. Yeoh, E.J., Ross, M.E., Shurtleff, S.A., Williams, W.K., Patel, D., Mahfouz, R., Behm, F.G., Raimondi, S.C., Relling, M.V., Patel, A., Cheng, C., Campana, D., Wilkins, D., Zhou, X., Li, J., Liu, H., Pui, C.H., Evans, W.E., Naeve, C., Wong, L., Downing, J.R.: Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell* 1, 133–143 (2002)
30. Bhattacharjee, A., Richards, W.G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E.J., Lander, E.S., Wong, W., Johnson, B.E., Golub, T.R., Sugarbaker, D.J., Meyerson, M.: Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma sub-classes. *Proc. Natl. Acad. Sci. USA* 98(24), 13790–13795 (2001)
31. Lu, J., Getz, G., Miska, E.A., Alvarez-Saavedra, E., Lamb, J., Peck, D., Sweet-Cordero, A., Ebert, B.L., Mak, R.H., Ferrando, A.A., Downing, J.R., Jacks, T., Horvitz, R.R., Golub, T.R.: Microrna expression profiles classify human cancers. *Nature* 435(7043), 834–838 (2005)
32. Nakai, K., Kanehisa, M.: Expert system for predicting protein localization sites in gram-negative bacteria. *PROTEINS: Structure, Function, and Genetics* 11, 95–110 (1991)
33. Fisher, R.A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7, 179–188 (1936)
34. Evett, I.W., Spiehler, E.J.: Rule induction in forensic science. In: KBS in Government, Online Publications, pp. 107–118 (1987)
35. Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* 2, 193–218 (1985)
36. Zhao, Y., Karypis, G.: Criterion functions for document clustering - experiments and analysis. Technical report, University of Minnesota, Department of Computer Science (2003)
37. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. Wiley, New York (1990)

On the Synchrony of Morphological and Molecular Signaling Events in Cell Migration

Justin Dauwels^{1,2,3}, Yuki Tsukada^{1,4,6}, Yuichi Sakumura^{4,6}, Shin Ishii^{4,6,5}, Kazuhiro Aoki⁵, Takeshi Nakamura⁵, Michiyuki Matsuda⁵, François Vialatte³, and Andrzej Cichocki³

¹ These authors contributed equally

² Massachusetts Institute of Technology, Cambridge, MA

³ RIKEN Brain Science Institute, Saitama, Japan

⁴ Nara Institute of Science and Technology (NAIST), Nara, Japan

⁵ Kyoto University, Kyoto, Japan

⁶ Institute for Bioinformatics Research and Development (BIRD), Japan Science and Technology Agency (JST)

jdauwels@mit.edu, {yuki-ts,saku}@is.naist.jp,
{ishii@i,kaoki@path1.med,tnakamr@lif,matsudam@path1.med}.kyoto-u.ac.jp,
{fvialatte,cia}@brain.riken.jp

Abstract. This paper investigates the dynamics of cell migration, which is the movement of a cell towards a certain target area. More specifically, the objective is to analyze the causal interdependence between cellular-morphological events and molecular-signaling events. To this end, a novel data analysis method is developed: first the local morphological changes and molecular signaling events are determined by means of edge evolution tracking (EET), next the interdependence of those events is quantified through the method of stochastic event synchrony (SES).

The proposed method is applied to time-lapse fluorescence resonance energy transfer (FRET) images of Rac1 activity in motile HT1080 cells; the protein Rac1 is well known to induce filamentous structures that enable cells to migrate. Results show a significant delay between local Rac1 activity events and morphological events. This observation provides new insights into the dynamic relationship between cellular-morphological change and molecular-signaling of migrating cells, and may pave the way to novel biophysical models of cell migration.

1 Introduction

In recent years, a wide variety of high-throughput approaches for biological experiments has emerged. Such experiments not only generate a vast amount of data, in addition, the scientific questions addressed through that data are often non-standard, and require entirely novel data analysis techniques. A prime example is micro-array data in genomics and proteomics.

In this paper, we deal with a similar example, i.e., live cell imaging data; this data allows us to gain more insight into the dynamics of cellular mechanisms. In particular, we investigate *cell migration*, which is the movement of a cell towards

a certain target area; it is one of the central processes in the development and maintenance of multicellular organisms. For example, tissue formation during embryonic development, wound healing and immune responses all require the orchestrated movement of cells in a particular direction to a specific location. Errors during this process have serious consequences, including mental retardation, vascular disease, tumor formation and metastasis. Despite the obvious importance, the dynamics behind cell migration are poorly understood. Such understanding may lead to the development of novel therapeutic strategies for controlling, for instance, invasive tumor cells [1]. Although the recent advancement of both molecular biology and the technique of live cell imaging shed some light on the mechanisms of cellular morphological regulation [2,3], and in parallel, some computational methods for quantitative analysis of time-lapse images have been developed (see, e.g., [4]), there remains a wide variety of computational challenges. One of those challenges is to analyze the interdependence of different kinds of events in time-lapse images of cells during cell migration.

In this study, we analyze *fluorescence resonance energy transfer* (FRET) images of Rac1 in motile HT1080 cells, which are human fibrosarcoma cells [7]. The protein Rac1 is well known to induce filamentous structures that enable cells to migrate. We analyze the statistical relation between Rac1 activity events and morphological events. For this purpose, we develop a novel computational method for quantifying those interdependencies. This method consists of two steps:

1. We first determine the morphological and molecular activity events from the time-lapse microscopy images. To this end, we apply *edge evolution tracking* (EET) [5]: it identifies cellular-morphological events and molecular-activity events, and determines the distance in space between those events, taking the evolution of the cell morphology into account.
2. Next we quantify the interdependence between those morphological and activity events by means of *stochastic event synchrony* (SES) [6]: using the EET distance measure, SES tries to align cellular morphological with molecular activity events; the better this alignment can be carried out, the more similar the event sequences are considered to be. In addition, the method is able to determine delays between both event sequences.

This paper is organized as follows. In the next section, we describe the edge evolution tracking (EET) method; in Section 3, we briefly review the stochastic event synchrony (SES) method. In Section 4, we combine those two methods to analyze live cell FRET data of Rac1 in motile HT1080 cells. In Section 5, we discuss our results, and in Section 6, we offer some concluding remarks.

2 Edge Evolution Tracking

In earlier work [5], we developed an algorithm, referred to as *edge evolution tracking* (EET), to quantify the dynamics of local cell morphological changes from FRET images. The method focusses on local *area* change instead of local

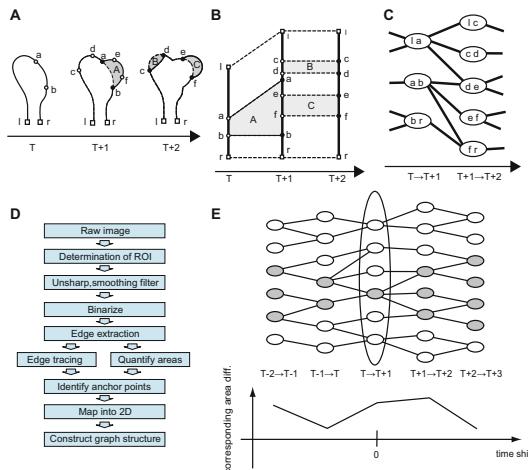


Fig. 1. Edge evolution tracking. (A) Cell boundary, anchor points *a-f* and area differences *A-C* from *T* to *T + 2*. The points *l* and *r* denote terminals of the boundary; (B) Segment transition diagram; (C) Non-vanishing segment graph; (D) Flow chart of EET; (E) Evolution of particular (non-vanishing) segment over time.

structural change as morphological property. As a consequence, EET does not make a clear distinction between structural differences such as filopodia and lamellipodia. However, such distinction is not required in the current study, and EET is therefore suitable for our purposes.

2.1 Principle

EET divides the cell edge boundaries in segments, and tracks how those segments evolve over time. By tracking boundary segments, one can identify which segments protrude, retract, or remain at the same position. The EET method consists of the following steps:

1. As a first step, the time-lapse microscopy images are preprocessed: we apply a smoothing filter, and next we threshold the pixel intensities, resulting in binary images. From those binary images, we extract the cell boundaries.
2. The cell boundaries are divided into segments (see Fig. 1A). The boundary points of each segment are identified as “anchor points” (lowercase letters in Fig. 1A). Those anchor points often separate parts of the boundary that protrude, retract, or remain unchanged. We also determine the center of each segment and its width.
3. We determine the correspondence between segments of consecutive time instances (see Fig. 1B). Some segments vanish in the next time instance, others remain. For example, segment [*a, e*] disappears in the transition from *T + 1* to *T + 2*. Non-vanishing segments may either grow, shrink, or remain unchanged as they evolve to the next time instance. In the segment-transition

diagram of Fig. II B, the anchor points of non-vanishing segments at consecutive time instances are interconnected by dotted lines, and the associated area differences are labeled. The sequence of non-vanishing segments may be represented by a graph structure as depicted in Fig. II C. Each node represents a non-vanishing segment. Two segments at consecutive time instances are connected if they overlap.

A flow chart of the EET procedure is shown in Fig. II D.

Once the EET method has been applied, one can compute several interesting quantities. For example, it allows us to investigate the evolution of a particular segment over time, using the graph of Fig. II C. As an illustration, Fig. II E (top) shows the time course of the colored non-vanishing segment at transition $T \rightarrow T + 1$; more specifically, it depicts all corresponding segments before T and after $T + 1$ (colored in grey). Fig. II E (bottom) shows how the area differences of that family of corresponding segments vary over time. Obviously, one may analyze all other nodes of Fig. II E (top) in a similar fashion.

We wish to point out that the parameters of the preprocessing step affect the results of EET. However, those results are consistent once the preprocessing parameters have been determined, even if the cells show strongly time-varying behavior.

2.2 Morphological and Activity Events

Activity events A_k are defined as follows. First one computes the local activity along each *point* P of the cell boundary; it is defined as the average intracellular FRET value inside a circle with center at P and radius R . The local activity of a *segment* is defined as the average of the local activity within that segment. If the local activity of a particular (non-vanishing) segment is *larger* than a certain threshold θ_{ac} , we say that an activity event (“activity burst”) has occurred at that segment of the cell boundary. In principle, one may define activity dips in a similar fashion. In this paper, however, we will merely consider activity bursts.

Morphological events $M_{k'}$ are extracted from the area differences. If the area difference of a particular (non-vanishing) segment is *larger* than a certain (positive) threshold θ_{ar} , we say that a morphological event has occurred at that segment of the cell boundary, in particular, a protrusion event. In principle, one may define retraction events in a similar fashion. In this paper, however, we will only study protrusion: more specifically, we wish to investigate how activity bursts are related to protrusion events.

Each event A_k and $M_{k'}$ is described by four parameters: (i) occurrence time; (ii) location on the cell boundary, which is given by the center of the corresponding segment; (iii) width, which is defined as length of the corresponding segment; (iv) amplitude (either local activity or area difference).

2.3 Spatial Distance between Activity and Morphological Events

In order to define the spatial distance between activity and morphology events A_k and $M_{k'}$ respectively, we make use of the diagrams obtained through EET

analysis. If both events occur at the same time instance, their distance is defined as the curve length of the shortest path *within the cell* that connects the two events. The shortest path is not necessarily a straight line, and the distance between A_k and $M_{k'}$ is in general non-euclidean. If the events A_k and $M_{k'}$ occur at different time instances, we first identify the segments $\tilde{M}_{k'}$ corresponding to $M_{k'}$ at the time frame of A_k . Since the segments $\tilde{M}_{k'}$ are neighboring, they can be treated as one “large” segment. We use segments $\tilde{M}_{k'}$ at time frame of A_k and not vice versa, since the boundary segments are defined based upon morphological characteristics. Eventually, the distance between A_k and $M_{k'}$ is defined as the distance between A_k and $\tilde{M}_{k'}$, i.e., the length of the shortest path *within the cell* that connects A_k with the center of $\tilde{M}_{k'}$.

3 Stochastic Event Synchrony

In earlier work [6], we developed a method to quantify the interdependence of two event sequences (“point processes”), referred to as *stochastic event synchrony* (SES). We will first briefly describe SES in the context of the application considered in [6], i.e., point processes in time-frequency domain. Next we outline how SES can be applied to morphological and activity events.

3.1 Bump Models in Time-Frequency Domain

In [6], one wishes to determine the similarity of pairs of signals (e.g., EEG signals). The latter are first transformed into time-frequency domain. The resulting time-frequency maps are usually sparse, and contain distinct “bumps” of activity (see Fig. 2). The EEG signals are modeled as sequences of such bumps, which are event sequences in time-frequency domain [8]. The resulting bump models X and X' , representing the most prominent oscillatory activity, are then aligned (see Fig. 3): bumps in one time-frequency map may not be present in the other map (“non-coincident” bumps); other bumps are present in both maps (“coincident bumps”), but appear at slightly different positions on the maps.

The black lines in Fig. 3(right) connect the centers of coincident bumps, and hence, visualize the offset in position between pairs of coincident bumps. Stochastic event synchrony consists of five parameters (SES) that quantify the alignment of two bump models:

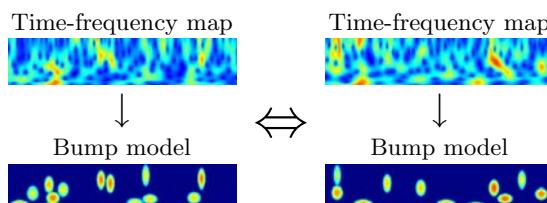


Fig. 2. Bump models. Top: time-frequency maps of the two signals; bottom: bump models extracted from the time-frequency maps.

- ρ : fraction of non-coincident bumps,
- δ_t and δ_f : average time and frequency offset respectively between coincident bumps,
- s_t and s_f : variance of the time and frequency offset respectively between coincident bumps.

The parameters ρ and s_t are related to similarity, since they quantify the synchrony between bump models; low ρ and s_t implies that the two bump models at hand are well synchronized.

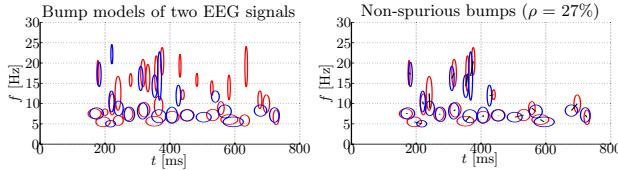


Fig. 3. Coincident and non-coincident activity (“bumps”). (left) bump models of two signals (one in red, the other in blue); (right) coincident bumps; the black lines connect the centers of coincident bumps.

The alignment of the two event sequences (cf. Fig. 3 (right)) is cast as a statistical inference problem [6]. The associated probabilistic model depends on the SES parameters $\theta = (\delta_t, \delta_f, s_t, s_f)$ besides the following two kinds of latent variables: (i) binary variables $C_{kk'}$, associated to each pair of bumps X_k and $X'_{k'}$, where $C_{kk'} = 1$ indicates that event X_k is coincident with event $X'_{k'}$, and where $C_{kk'} = 0$ otherwise; (ii) binary variables B_k and $B'_{k'}$, where $B_k = 1$ and $B'_{k'} = 1$ indicate that X_k and $X'_{k'}$ respectively are non-coincident. The latent-variable model is of the form:

$$\begin{aligned} p(x, x', b, b', c, \theta) \propto & \prod_{k=1}^n (\beta \delta[b_k - 1] + \delta[b_k]) \prod_{k'=1}^{n'} (\beta \delta[b'_{k'} - 1] + \delta[b'_{k'}]) \\ & \cdot \prod_{k=1}^n \prod_{k'=1}^{n'} \left(\mathcal{N}(t'_{k'} - t_k; \delta_t, s_t) \mathcal{N}(f'_{k'} - f_k; \delta_f, s_f) \right)^{c_{kk'}} \\ & \cdot \prod_{k=1}^n \left(\delta[b_k + \sum_{k'=1}^{n'} c_{kk'} - 1] \right) \prod_{k'=1}^{n'} \left(\delta[b'_{k'} + \sum_{k=1}^n c_{kk'} - 1] \right) p(\delta_t) p(s_t) p(\delta_f) p(s_f), \quad (1) \end{aligned}$$

where β is a constant, n and n' is the total number of bumps in X and X' respectively, and $\mathcal{N}(x; m, s)$ stands for a univariate Gaussian distribution with mean m and variance s [6]. Note that the Gaussian factors in (1) correspond to an *euclidean* metric in the time-frequency domain. For convenience, we choose improper priors $p(\delta_t) = 1 = p(\delta_f)$ and conjugate priors $p(s_t)$ and $p(s_f)$. The SES parameters $\theta = (\delta_t, \delta_f, s_t, s_f)$ and the latent variables C, B and B' are determined jointly by MAP estimation; we refer to [6] for more details.

3.2 Events in Cell Migration

Instead of two bump models X and X' , we now have an activity event sequence A and a morphological event sequence M . The events in those sequences also have two coordinates: time t and position u along the cell boundary; the parameters δ_f and s_f are replaced by δ_u and s_u . There are two important differences with the previous application: (1) the timing offset δ_t is no longer (almost) zero; (2) the distance between events is no longer euclidean, since it is computed based upon the EET graph (cf. Section 2.3). The factor $\mathcal{N}(f'_{k'} - f_k; \delta_f, s_f)$ in (1) is replaced by $\mathcal{N}(g(t_k, u_k, t'_{k'}, u'_{k'}); \delta_u, s_u)$, where g corresponds to the EET distance measure, and t, t', u , and u' are the occurrence times and positions of the activity events A and morphological events M respectively.

4 Application

We applied our methods to the fluorescence resonance energy transfer (FRET) images of motile HT1080 cells, which are human fibrosarcoma cells [7]. More precisely, FRET technique was used for monitoring activity of *Rac1*; the latter is a member of Rho family small GTPases (see previous studies on FRET imaging, e.g., [5]). Rho GTPases act as binary switches by cycling between inactive and active states, and play a key role in linking biochemical signaling with biophysical cellular behaviors; this is achieved mainly through reorganization of the actin and microtubule cytoskeleton. In particular, *Rac1* is well known to induce specific filamentous actin structures called *lamellipodia*.

The data set consists of 3 FRET movies; each movie is recorded from a different HT1080 cell, has a time-lapse (frame-rate) of 1 frame per minute, and

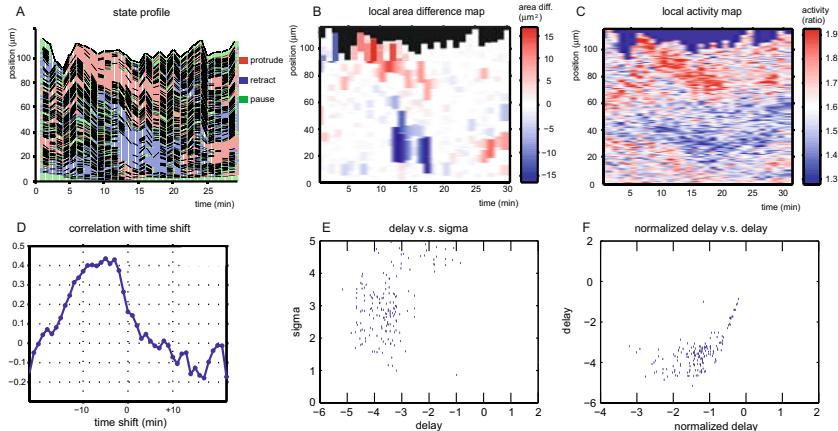


Fig. 4. Results of EET and SES. (A) Area difference values; (B) Activity values; (C) Local morphological status. Red, blue and green denote protrusion, retraction and pause, respectively; (D) Cross-correlation between area difference and activity map. (E) Distribution of delay parameter and (F) distribution of normalized delay parameter.

lasts about 30 minutes. The HT1080 cells were uniformly stimulated with nerve growth factor, 24 hours before the movies were recorded. The cells showed spontaneous migration during the recording.

5 Results and Discussion

First we applied EET to the FRET time-lapse images in order to determine the morphological and molecular-signaling events. This procedure resulted in the area difference map, activity map, and state profile for each of the three FRET movies. Figs. 4 illustrates those three maps for one of the analyzed movies (A, B, and C respectively). In each of the three maps, the horizontal and vertical axis represent time and position along the boundary respectively. The area difference map (see Fig. 4A) and local activity map (see Fig. 4B) depict the area difference and local activity respectively for each (non-vanishing) segment of the cell boundary as a function of time. The state profile (see Fig. 4C) depicts the morphological state of each (non-vanishing) segment of the cell boundary as a function of time. The morphological and molecular-signaling events were extracted from the area difference and local activity map respectively by thresholding (with thresholds θ_{ar} and θ_{ac} respectively).

Next we applied the stochastic event synchrony (SES) method in order to analyze the interdependence between morphological events and activity events. For reasonable values of the thresholds θ_{ar} and θ_{ac} , activity kernel radius R , parameter β , we found that the delay δ_t between morphological events and activity events is negative (see Fig. 4E); in other words, on average the morphological events occur first, followed by activity events. We observed this in each of the three movies.

As an alternative approach, we calculated cross-correlation between the area difference and the activity map (see Fig. 4D), as in a previous study [5]. The cross-correlation shows high correlation at negative time-shifts, in agreement with the SES analysis. However, it is hard to extract a precise value for the delay from the cross-correlation function shown in Fig. 4D, since the peak is wide. On the other hand, SES yields more precise estimates of delay, i.e., the parameter δ_t ; the uncertainty in δ_t is quantified by σ_t , the standard deviation of the timing jitter, more precisely, by the ratio $|\delta_t|/\sigma_t$ (see Fig. 4F); the latter is significantly larger than 1 in the application at hand, and hence, the estimate δ_t can be considered reliable.

6 Conclusion

We studied the dynamics of cell migration, particularly, we analyzed the causality between cellular-morphological events and molecular-signaling events. To this end, we developed a novel data analysis method that blends edge evolution tracking (EET) with the method of stochastic event synchrony (SES).

We used that method to study time-lapse fluorescence resonance energy transfer (FRET) images of Rac1 activity in motile HT1080 cells; our results indicate

that local Rac1 activity events are delayed w.r.t. morphological events. This seems to suggest that the cell first changes its shape, probably as a random fluctuation, and next there is an influx of Rac1, most likely required to *sustain* that morphological change. Obviously, further investigations are required to unravel the underlying molecular mechanisms.

In future work, we will apply our method to a large number of FRET movies in order to confirm our findings. In addition, we will incorporate more detailed biophysical processes into our method, which should allow us to refine our observations.

References

1. Lauffenburger, D.A., Horwitz, A.F.: Cell migration: a physically integrated molecular process. *Cell* 84, 359–369 (1996)
2. Dormann, D., Weijer, C.J.: Imaging of cell migration. *The EMBO Journal* 25(15), 3480–3493 (2006)
3. Pollard, T.D.: The cytoskeleton, cellular motility, and the reductionist agenda. *Nature* 422, 741–745 (2003)
4. Gerlich, D., Mattes, J., Eils, R.: Quantitative motion analysis and visualization of cellular structures. *Methods* 29, 3–13 (2003)
5. Tsukada, Y., Aoki, K., Nakamura, T., Sakumura, Y., Matsuda, M., Ishii, S.: Quantification of local morphodynamics and local GTPase activity by edge evolution tracking. *PLoS Comput. Biol.* 4(11), e1000223 (2008)
6. Dauwels, J., Vialatte, F., Rutkowski, T., Cichocki, A.: Measuring neural synchrony by message passing. In: *Advances in Neural Information Processing Systems* 20. MIT Press, Cambridge
7. Rasheed, S., Nelson-Rees, W.A., Toth, E.M., Arnstein, P., Gardner, M.B.: Characterization of a newly derived human sarcoma cell line (HT-1080). *Cancer* 33, 1027–1033 (1974)
8. Vialatte, F., Martin, C., Dubois, R., Haddad, J., Quenet, B., Gervais, R., Dreyfus, G.: A machine learning approach to the analysis of time-frequency maps, and its application to neural dynamics. *Neural Networks* 20, 194–209

MISCORE: Mismatch-Based Matrix Similarity Scores for DNA Motif Detection

Dianhui Wang and Nung Kion Lee

Department of Computer Science and Computer Engineering
La Trobe University, Melbourne, Victoria, 3086, Australia
`dh.wang@latrobe.edu.au`

Abstract. To detect or discover motifs in DNA sequences, two important concepts related to existing computational approaches are motif model and similarity score. One of motif models, represented by a position frequency matrix (PFM), has been widely employed to search for putative motifs. Detection and discovery of motifs can be done by comparing kmers with a motif model, or clustering kmers according to some criteria. In the past, information content based similarity scores have been widely used in searching tools. In this paper, we present a mismatch-based matrix similarity score (namely, MISCORE) for motif searching and discovering purpose. The proposed MISCORE can be biologically interpreted as an evolutionary metric for predicting a kmer as a motif member or not. Weighting factors, which are meaningful for biological data mining practice, are introduced in the MISCORE. The effectiveness of the MISCORE is investigated through exploring its separability, recognizability and robustness. Three well-known information content-based matrix similarity scores are compared, and results show that our MISCORE works well.

1 Introduction

Motif refers to a collection of transcription factor binding sites (or simply, binding sites) located in promoter regions of genes. Detection of binding sites is crucial in deciphering gene regulatory networks. In the past years, computational tools have been developed to discover putative binding sites and achieved some promising results. The rapid sequencing of genome data with related species for footprinting resulted in further discovery of many unknown binding sites. In addition, collections of true binding sites obtained using SELEX, chromatin immunoprecipitation (ChIP) or other wet lab techniques have become more readily accessible from several public databases. Examples of these databases are JASPAR [1] and TRANSFAC [2]. Motif models can be constructed from these collections, and it can be used to find potential novel binding sites in DNA sequences using a model specific scoring or similarity function.

Motif detection is a unique classification task because only a small number of positive examples are usually available and a large number of negative examples are ill-defined. This problem is usually referred as one-class instead of multiple

classes classification problem. Standard classification approaches, such as neural networks, decision tree and support vectors machine, could not perform well due to the imbalanced data in this domain applications. It is also noticed that the nearest neighbor classifiers suffer from exhaustive comparisons between a testing kmer and all negative examples. Hence, a widely adopted method for motif detection is to model the available binding sites and use it for classification or prediction. Position frequency matrix (PFM) [12] is the most widely used motif model due to its simplicity and small parameters set. More complicated models have been proposed but they do not always outperform this simpler model.

This paper aims to develop model-based classification techniques for resolving the problem of motifs detection. It is assumed that information on motifs associated with some certain transcription factor is available. A binary matrix encoding method for DNA data representation and a mathematical formula for computing a similarity score between a kmer and the PFM are given. Our proposed MISCORE has evolutionary interpretations and it can be employed in other biological data mining practices. The effectiveness of our proposed solution for motif detection is evaluated using some benchmarked datasets.

2 Methods

2.1 Motif Model

First, some notations used in this paper are defined. Let S be a motif that consists of a set of binding sites associated with a certain transcription factor. It is assumed that each binding site in S has a fixed length k , which can be achieved through multiple alignment tools. A kmer is a subsequence of length k in DNA sequences, i.e., $T_1T_2 \cdots T_k$, where $T_j \in \Sigma = \{A, C, G, T\}$, $j = 1, 2, \dots, k$. In this paper, a binary matrix representation is used, which is compatible to the PFM model. The encoded kmer is given by $e(kmer) = [a_{ij}]_{4 \times k}$, $a_{ij} = 1$ if $T_j = V_i$, otherwise $a_{ij} = 0$, where $(V_1, V_2, V_3, V_4) = (A, C, G, T)$.

Motifs can be expressed as consensus or probabilistic profiles [12]. Profile representations such as Position Frequency Matrix (PFM) assign a relative frequency to each possible nucleotide at each position in the motif. It is a $4 \times k$ matrix and each column vector represents the position wise observed nucleotides (i.e. A, C, G, T) frequency in a motif. To avoid zero division in evaluating relative entropy of the PFM, Bayesian estimation techniques [9] are used to add pseudo count values to the entries of the PFM.

2.2 Mismatch Based Matrix Similarity Score

To build a PFM model based classifier for motifs detection, it is fundamental to define a proper similarity function that reflects the closeness concept in the biological sense. Due to functional associations of binding sites [3], they are evolutionary constrained as compared to background sequences [8]. Hence, a kmer is likely to be a true site if it has limited mismatches to every binding site in a motif. This understanding forms the basis of this work.

The following proposition gives a way to compute the number of average mismatch between a kmer and a motif model.

Proposition 1: Let $R_{[0,1]}^{4 \times k}$ and $R_{\{0,1\}}^{4 \times k}$ represent the sets of real matrices with size $4 \times k$ and entries taking values in unitary interval $[0, 1]$ and binary values, respectively; $S = \{K_p \in R^{4 \times k}, p = 1, 2, \dots\}$ be a motif, which is modeled by its PFM. Define a generalized Hamming distance function over $R_{[0,1]}^{4 \times k}$ as follows:

$$d(M_1, M_2) = \frac{1}{2} \sum_{j=1}^k \sum_{i=1}^4 w_j |m_{i,j}^{(1)} - m_{i,j}^{(2)}|, \quad (1)$$

where $0 \leq w_j \leq 1, j = 1, 2, \dots, k$.

Then, for any kmer $K \in R_{\{0,1\}}^{4 \times k}$, we have

$$\frac{1}{|S|} \sum_{K_p \in S} d(K, K_p) = d(K, \frac{1}{|S|} \sum_{K_p \in S} K_p). \quad (2)$$

where $|S|$ represents the cardinality of the set S .

Let $K = [e_{i,j}] \in R_{\{0,1\}}^{4 \times k}$ and the PFM model be

$$M = \frac{1}{|S|} \sum_{K_p \in S} K_p = [m_{i,j}]. \quad (3)$$

Then, (2) can be rewritten as:

$$d(K, M) = \sum_{j=1}^k w_j [1 - g_j], \quad (4)$$

where $g_j = m_{i,j}$ if $e_{i,j} = 1$ for $j = 1, 2, \dots, k$.

Without confusion, hereafter, we will refer to the average mismatch score between a kmer and a motif simply as *average mismatch*. The average mismatch score given in (4) has several useful properties. Firstly, it can be used in k-NN classifiers design, where the average mismatch can be used as a similarity metric. The true binding sites can be considered as evolved results from common ancestor sites, represented by the PFM and has constrained average mismatch to it. Hence, the motif detection task is to search for these neighboring kmers. Secondly, it can be employed in other data mining or statistical algorithms for motif discovery. For instance, the PFM can be regarded as a prototype or center of a model cluster, and the average mismatch can be used to assign data into clusters.

Based on (2), we can define an model mismatch score (MMS) of a motif, that is:

$$R_s = \frac{1}{|S|} \sum_{K_p \in S} d(K_p, M). \quad (5)$$

Now, our motif detection rule can be stated as follows:

$$\text{If } d(X, M) \leq (1 + \lambda)R_s, \text{ Then } X \in \text{motif}, \quad (6)$$

where the $\lambda > 0$ is a cut-off threshold.

Usually, the MMS is rather conservative for motif detection, especially for the case that binding sites are not conserved. Thus, we need a proper scalar multiplied by the MMS to enlarge the possibility to detect weak binding sites. The setting of this cut-off value often incur the trade-off between sensitivity and specificity. That is, a higher cut-off threshold will increase the inclusion of true positives and at the same time false positives as well.

To compare with other similarity scores, a normalized average mismatch d_n is given below:

$$d_n(K, M) = \frac{d_{\max}(M) - d(K, M)}{d_{\max}(M) - d_{\min}(M)}, \quad (7)$$

where $d_{\min}(M) = \sum_{j=1}^k w_j [1 - \max_i \{m_{i,j}\}]$, $d_{\max}(M) = \sum_{j=1}^k w_j [1 - \min_i \{m_{i,j}\}]$.

3 Performance Evaluation

MISCORE is evaluated by using three criteria: separability, recognizability and robustness. We compared MISCORE with three others well-known similarity scores used in motif discovery algorithms: PWM [12], MATCH[7] and MATInspector [10].

Artificial sequences were generated by using RSAT tool with 3-rd order Markov model for performance comparisons. The real binding sites used in this study were obtained from RegulonDB 6.0 (Jan 1 2008) [5], TRANSFAC [14] and JASPAR [11]. The *Saccharomyces cerevisiae* chromosomes were downloaded from SGD[2].

3.1 Separability

Separability is the ability of discriminating the true sites from random kmers during motif detection. To understand this better, we first consider statistically significance of the motif model scores. Fig. 1 illustrates the normalized scores distribution for four tools. The normalized scores from all methods, particularly our method, show a normal distribution. Thus, it is a practical measure of statistical significance when separating motifs from random kmers according to the central limit theorem. The score distributions for both real and artificial sequences share quite similar patterns for all methods.

To examine the power of our method in distinguishing true binding sites from random kmers, four proteins (transcription factors) were employed, they are: CRP (161), ArcA(157), FIS(142), and MA00078(31). We examined their effectiveness using the Receiver Operator Cost (ROC) and area under ROC (aROC)

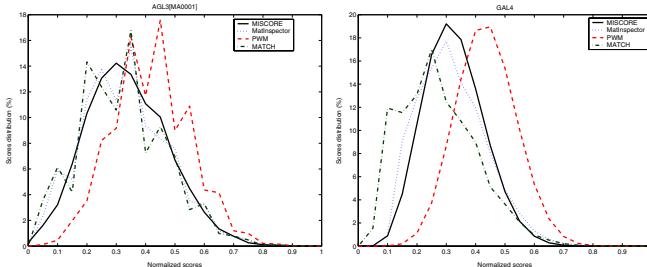


Fig. 1. a) This graph shows the normalized scores distribution for artificial sequences. It was obtained by scanning the PFM constructed using 25 AGL protein (MA0001) against 20 random sequences with length 1000bp each. The normalized scores for each method were binned and their relative frequencies were plotted. b) The motif profile from GAL4 protein was used to scan Chromosome II of *Saccharomyces cerevisiae*. There are verified evidences in the literature that this protein binds to several sites in this chromosome [21615].

Table 1. List of GAL4 transcription factor binding sites discovered by MISCORE using $\lambda = 0.032$

Chr	d	d_n	Strand ⁺	Site	Start [!]	Evidence
2	6.73	0.89	+	CGGTCAACAGTTGTCCG	275703	SGD:S000082749*
2	7.45	0.82	+	CGCTCAACAGTGCTCCG	275790	SGD:S000082751*
2	6.45	0.92	+	CGGGGCACAGCCTCCG	278587	SGD:S000082758*
4	7.18	0.85	+	CGGTCCACTGTGTGCCG	463143	SGD:S000083177*
4	7.00	0.87	-	CGGCACACAGTGGACCG	1068760	SGD:S000083177*
7	7.27	0.84	-	CGGTGCAATATACTCCG	835496	SGD:S000083752*
12	6.45	0.92	+	CGGCGGTCTTCGTCCG	289797	SCPD(YLR081W/GAL2)
13	6.45	0.92	+	CGGCGCACTCTCGCCCG	171422	SGD:S000085008*
14	7.36	0.83	+	CGGAGTGCTCTCGCCG	488275	SGD:S000085433*
14	7.36	0.83	+	CGGGGCAGACTATTCCG	550746	Vashee et al., 1993

⁺ forward strand(+), reverse complement(-). *SGD evidence as reported by [6].

[!] the position is relative to the chromosome.

[4]. Fig. 2 shows that MISCORE performance is comparable to other three methods, and slightly better as compared to MatInspector and MATCH. The aROC for MISCORE is better than these two methods for CRP and FIS datasets. For the FIS dataset, our method clearly outperforms other methods. From this comparative study, we can conclude: (i) In terms of separability, our MISCORE performs comparably to all other methods, and outperforms MatInspector and MATCH for three testing datasets; (ii) For the conserved motif (e.g. MA0078), all methods perform evenly. However, for longer and less conserved motifs, MISCORE and PWM perform better.

To further compare the effectiveness of our proposed MISCORE for separability, the normalized average mismatch score was used to rank the true binding sites against the random kmers. This measure is an indicator of distinguishability of the metrics. Fig. 3 shows the results of motif ranking comparisons in terms of

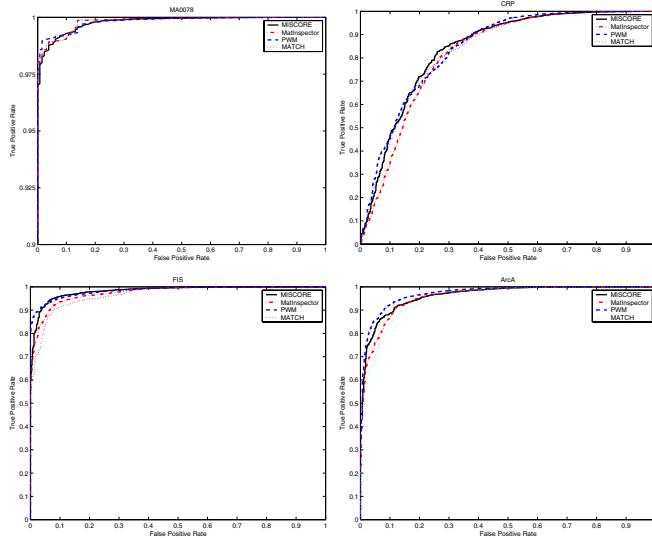


Fig. 2. Comparisons of four methods for motif detection in artificial sequences using ROC. Each motif is partitioned randomly into two parts: model set M (60%) and test set T (40%). The M set was used to construct the PFM, whereas the binding sites in T were randomly planted into random sequences (500bp and one planted binding site per sequence). A total of randomly partitioned 10 sets were created from each original set. Each method was used to scan the random sequences in order to generate the ROC points. The ROCs from these 10 sets were averaged and plotted (See Fig. 2). In the recognition, only an exact match was considered a true positive. The area under ROCs (aROC) for MISCORE, MATINSPECTOR, PWM and MATCH are CRP(0.8381, 0.8201, 0.8387, 0.8324) ArcA(0.9632, 0.9578, 0.9716, 0.9480), MA0078(0.9980, 0.9982, 0.9982, 0.9981) and FIS(0.9820, 0.9742, 0.9839, 0.9652) respectively.

frequency. MISCORE demonstrates stronger power for separability compared to other three tools. For examples, MISCORE has 978 times to have smaller number of false positives than that for PWM, that is, $\sum_{i=1}^{2^{199}} \sigma(|Z_{pwm}(K_i)| - |Z_{miscore}(K_i)|) = 978$, where $\sigma(p) = 1(0)$, if $0 < p$ (otherwise); $Z_{method}(K_i) = \{Y : d_n(K_i, M_i) < d_n(Y, M_i), \text{ for any kmer } Y\}$, and M_i is the motif model constructed by remainings after picking out the binding site K_i from the original motif. In this case, we say the frequency of ‘MISCORE is better than PWM’ is 978.

3.2 Recognizability

Once a motif model is built, it can be used for motif detection to find novel sites associated with the same protein. This subsection investigates the potential of MISCORE to recognize true binding sites in real sequences. We extracted eleven binding sites associated with transcription factor GAL4 from [\[1\]](#) to construct a PFM. The consensus for GAL4 motif is CGGNNNNNNNNNNNNCCG. The value of MMS R_s for this motif is 7.22 with standard deviation 0.64. The rule described

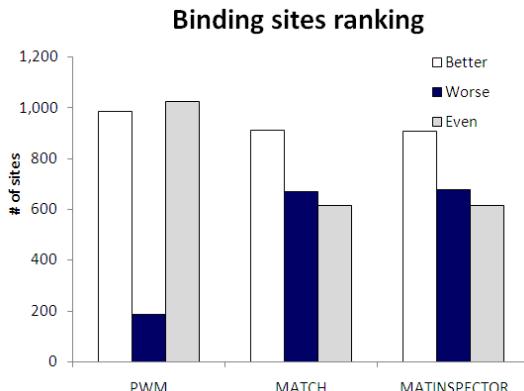


Fig. 3. Comparisons of motif sites ranking using an artificial sequence. We prepared a 2.5mbp artificial sequence generated by a 3rd order Markov chain human background model. We used the leave-one-out procedure in this experiment. For each true binding site in turn, we computed similarity scores using different methods for the PFM model constructed by the remainings. These scores were then ranked using (7) against the scores obtained for random kmers. Using 2,199 binding sites from 72 human motifs, we counted the frequencies of better, even and worse false positive rates of MISCORE compared to three other tools.

in (6) with all weighting factors as 1 was used to search for true binding sites in the *Saccharomyces cerevisiae* genome.

A total of 34 and 104 predicted binding sites are returned using λ values 0 and 0.032 respectively. After carefully removing the double strand overlapped sites, 24 and 83 predicted binding sites are left in each set. From the 83 sites, we found 10 of it are true positives and the remaining are false positives (mostly they are located in the coding regions). The results are shown in Table II.

4 Concluding Remarks

This paper presents a mismatch-based matrix similarity score (MISCORE) for motifs detection in DNA sequences. A crisp detection rule is constructed using the PFM model and the MISCORE. With such a rule and some datasets, we evaluate the effectiveness of our proposed similarity score in terms of separability, recognizability and robustness. Results show that MISCORE performs favorably compared to the three well-known similarity scores used in motif discovery algorithms. From robustness analysis, the motif model is quite reliable provided that over 50% true binding sites can be collected to construct its PFM. It should be noticed that the proposed techniques in this paper is only applicable for conserved motifs detection. For weak motifs, detection rules can be more complicated but our MISCORE will be still useful.

Two important issues related to MISCORE need further studies, that is, assignment of the weighting factors in (4) and optimization of the cut-off value in

(6). It is believed that an appropriate cost function will be used to solve such an optimization task.

References

1. Bram, N.F., et al.: A gal family of upstream activating sequences in yeast: roles in both induction and repression of transcription. *The EMBO Journal* 5(3), 603 (1986)
2. Cherry, J.M., et al.: Sgd: *Saccharomyces* genome database. *Nucleic Acids Res.* 26(1), 73–79 (1998)
3. Doniger, S., et al.: Identification of functional transcription factor binding sites using closely related *saccharomyces* species. *Genome Research* 15, 701–709 (2005)
4. Fawcett, T.: An introduction to roc analysis. *Pattern Recognition Letters* 27(8), 861–874 (2006)
5. Gama-Castro, S., et al.: Regulondb (version 6.0): gene regulation model of *escherichia coli* k-12 beyond transcription, active (experimental) annotated promoters and textpresso navigation. *Nucleic Acids Res.* 124, D120–D124 (2008)
6. Harbison, C., et al.: Transcriptional regulatory code of a eukaryotic genome. *Nature* 431(7004), 99–104 (2004)
7. Kel, A.E., et al.: Match: A tool for searching transcription factor binding sites in dna sequences. *Nucleic Acids Res.* 31(13), 3576–3579 (2003)
8. Moses, et al.: Position specific variation in the rate of evolution in transcription factor binding sites. *BMC Evolutionary Biology* 3(1), 19 (2003)
9. Osada, R., et al.: Comparative analysis of methods for representing and searching for transcription factor binding sites. *Bioinformatics* 20(18), 3516–3525 (2004)
10. Quandt, K., et al.: Matlnd and Matlnspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucl. Acids Res.* 23(23), 4878–4884 (1995)
11. Sandelin, A., et al.: Jaspar: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* 94(Database issue), D91–D94 (2004)
12. Storno, G.D.: DNA binding sites:representation and discovery. *Bioinformatics* 1, 16–23 (2000)
13. Tomovic, A., Oakeley, E.J.: Position dependencies in transcription factor binding sites. *Bioinformatics* 23(8), 933–941 (2007)
14. Wingender, E., et al.: Transfac: a database on transcription factors and their dna binding sites. *Nucleic Acids Res.* 24(1), 238–241 (1996)
15. Zhu, J., Zhang, M.Q.: Scpd: a promoter database of the yeast *saccharomyces cerevisiae*. *Bioinformatics* 15(7-8), 607–611 (1999)

Ensembles of Pre-processing Techniques for Noise Detection in Gene Expression Data

Giampaolo L. Libralon¹, André C. Ponce Leon Ferreira Carvalho¹,
and Ana C. Lorena²

¹ICMC/USP - São Carlos - Caixa Postal 668
13560-970 - São Carlos, SP, Brazil

glibralon@gmail.com, andre@icmc.usp.br

²Center of Mathematics, Computation and Cognition (CMCC)
ABC Fed. Univ. (UFABC), Santo André, SP, Brazil
ana.lorena@ufabc.edu.br

Abstract. Due to the imprecise nature of biological experiments, biological data are often characterized by the presence of redundant and noisy data, which are usually derived from errors associated with data collection, such as contaminations in laboratorial samples. Gene expression data represent an example of noisy biological data that suffer from this problem. Machine Learning algorithms have been successfully used in gene expression analysis. Although many Machine Learning algorithms can deal with noise, detecting and removing noisy instances from data can help the induction of the target hypothesis. This paper evaluates the use of distance-based pre-processing techniques in gene expression data, analyzing the effectiveness of these techniques and combinations of them in removing noisy data, measured by the accuracy obtained by different Machine Learning classifiers over the pre-processed data. The results obtained indicate that the pre-processing techniques employed were effective for noise detection.

1 Introduction

Noise can be defined as an example apparently inconsistent with the remaining examples in a data set. The presence of noise in a data set can decrease the predictive performance of Machine Learning (ML) algorithms, by increasing the model complexity and the time necessary for its induction. Data sets with noisy instances are common in real world problems, where the data collection process can produce noisy data.

Data are usually collected from measurements related with a given domain. This process may result in several problems, such as measurement errors, incomplete, corrupted, wrong or distorted examples. Therefore, noise detection is a critical issue, specially in domains demanding security and reliability. The presence of noise can lead to situations that degrade the system performance or the security and trustworthiness of the involved information [6].

Several approaches and algorithms for noise detection can be found in the literature. This paper focus on the use of distance-based techniques in a pre-processing phase. This phase aims to identify possible noisy examples and remove them. In this work, ML algorithms were trained with the original data sets and with different sets of

pre-processed data produced by the application of distance-based noise detection techniques. By evaluating the difference of performance among classifiers generated over original (without pre-processing) and pre-processed data, the effectiveness of distance-based techniques in recognizing noisy cases can be estimated. Afterwards, in an attempt to improve the performance achieved in the pre-processing, the authors combine different pre-processing techniques in a committee (ensemble).

Details of the techniques used are presented in Section 2. The methodology, data sets and ML algorithms employed in the experiments are described in Section 3. The results obtained are presented and discussed in Section 4. Finally, Section 5 has the main conclusions from this work.

2 Noise Detection

There are different pre-processing techniques able to detect and remove noise. In this paper, distance-based techniques, where noise detection is based on the distance between an instance and its neighbors, are evaluated. This is the approach used in [7][10]. Distance-based techniques are simple to implement and do not make any assumptions about the data distribution. However, they have high requirements of memory space and computational time, resulting in a complexity directly proportional to data dimensionality and the number of examples [6].

These techniques use similarity measures to calculate the distance between instances and use this information to identify possible noise. The similarity measure used in the distances' calculations has a higher effect on the performance achieved by these techniques.

For high dimensional data sets, the typical used Euclidian metric is not adequate [1], since data is often sparse. The *HVDM (Heterogeneous Value Difference Metric)* metric is shown by [14] as suitable to deal with such problems and was then used in this paper. This metric is based on the distribution of the attributes in the data set, according to their output values, and not only in punctual values, as is observed in the Euclidian distance and other similar distance metrics. Please refer to [14] for further information about the HVDM metric.

The *k*-nearest neighbor (*k*-NN) algorithm was used for finding the neighbors of a given instance. This algorithm defines the classification of an instance according to the class of the majority of its *k* nearest neighbors. The value of the *k* parameter, which represents the number of nearest neighbors of a given instance, influences the performance of *k*-NN.

The techniques evaluated in this paper are: *Edited Nearest Neighbor (ENN)*, *Repeated ENN (RENN)*, *AllkNN*, *Decremental Reduction Optimization Procedures 2 to 5 (DROPs 2 to 5)* and *Decremental Encoding Length (DEL)* algorithms. The first three algorithms listed are considered noise detection filters, while the remaining are considered data reduction techniques, removing also redundant data.

Let T be the original training set and S be a subset of T , obtained by the application of any of the distance-based techniques evaluated. Now suppose that T has n instances $\mathbf{x}_1, \dots, \mathbf{x}_n$. Each instance \mathbf{x} of T (and also of S) has k nearest neighbors.

The *ENN* algorithm was proposed in [15]. Initially, $S = T$, and an instance is considered noise and then removed from the data set if its class is different from the class

of the majority of its k nearest neighbors. This procedure removes mislabeled data and borderlines. In the *RENN* technique, the *ENN* algorithm is repeatedly applied to the data set until all instances in it have the majority of its neighbors with the same class. Finally, *AllkNN* was proposed in [11] and is also an extension of the *ENN* algorithm. It proceeds as follows: for $i = (1, \dots, k)$, mark as incorrect (possible noise) any instance incorrectly classified by its i nearest neighbors. After the analysis of all instances in the data set, it removes the signalized instances.

DROPs 2 to 5 and DEL algorithms were proposed by [13]. Basically, they present two phases, a noise detection and removal phase and a redundant data detection and removal phase.

3 Experiments

All experiments performed employed the 10-fold cross validation method [8]. All selected data sets were presented to the pre-processing techniques investigated. Next, their pre-processed versions, resulting from the application of each technique, were presented to ML algorithms. Each of the original data sets analyzed was also presented directly to the ML algorithms, aiming to compare the performance obtained by ML algorithms with the original data sets and with their pre-processing versions. The error rate obtained by the ML algorithms is calculated by the mean of the individual errors obtained for each test data set partition. Each noise detection technique was applied 10 times, one for each training fold.

Those pre-processing techniques for which the pre-processed data improved the performance of the ML techniques, regarding the original data sets, were later combined in ensembles. The combination used a majority vote methodology, where a example is considered noise only if, given n techniques, at least m of them, with $m > n/2$, classify it as noise. If this occurs, the example is removed from the pre-processed data set. Otherwise, it is kept. The use of ensembles aims trying to improve the results obtained by the pre-processing techniques.

The experiments were run in a 3.0 GHz Intel Pentium 4 dual processor with 1.0 GB of RAM. For the noise detection pre-processing techniques evaluated, the code provided by [13] was used. The number of nearest neighbors (k parameter) was set as 1, 3 or 9, to follow a geometric progression including the value 3, the default value of the mentioned code.

The following ML algorithms were used in the experiments: J48, a version of the C4.5 [9]; Repeated Incremental Pruning to Produce Error Reduction (RIPPER) algorithm [2], both using the *Weka simulator* [5]; and Support Vector Machines (SVMs) [12], implemented by the *SVMTorch II* [3] software.

The experiments were performed using gene expression analysis data sets, particularly for gene expression classification. These data sets have a large number of attributes and are very susceptible to noise. The gene expression data sets used in the experiments are described in Table 1. For each data set, it has the number of instances, the number of attributes and the existent classes. The experimental results were evaluated using the statistical test of Friedman and Bonferroni-Dunn's multiple comparisons post-hoc test [4]. The statistical tests were applied with 95% of confidence level.

Table 1. Description of the data sets analyzed

Data set	Instances	Attributes	Classes
ExpGen	207	79	B, H, T, R, P
Golub	72	3571	ALL, AML
Leukemia	327	271	BCR, E2A, HYP, MLL, T-ALL, TEL, OTHERS
Lung	197	1000	AD, SQ, COID, NL
Colon	62	2000	normal, tumor

4 Experimental Results

Each pre-processing technique removed a different amount of data. The AllKNN, ENN and RENN techniques, referred from now on as Group 1, removed less examples than DROPs 2 to 5 and DEL techniques, named from now on as Group 2. For Group 1, the reduction was between 20% to 30%, except for the Colon data set, with reductions between 30% to 40%. For Group 2, the reductions varied between 80% to 95%. Most of the examples removed by Group 1 were also removed by Group 2.

The running time for each pre-processing techniques was less than one minute for a given data set. It is important to mention that the pre-processing phase is applied only once for each data set analyzed, generating a pre-processed data set that can be used several times by different ML algorithms. As expected, more complex data sets had longer processing times.

The noise detection techniques investigated were evaluated measuring how they affected the performance of the ML algorithms regarding training time, model complexity and classification error rates. Due to space limitation, we only show here the average classification error rates.

4.1 Individual Noise Detection Techniques

Table 2 illustrates the average error rates for all five data sets and each ML algorithm. The first row has the average error rates for the original data sets. The other rows show these results for the seven pre-processing techniques using different values for k . The best results for each ML technique are shown in **bold** and error rates similar to the best ones for each data set are shown in *italic*.

According to these results, for the SVMs, there was an improvement in the classification accuracy for the *Lung* data set using the Group 1 techniques. Only few improvements can be seen for the other data sets. It is worth noting that Group 2 techniques performed significantly worse than Group 1 techniques.

The RIPPER algorithm presented similar error performance for the original and pre-processed data for the *Leukemia*, *ExpGen* and *Colon* data sets. In the last two data sets, some results were improved by the pre-processing. For the remaining pre-processed data sets, *Lung* and *Golub*, some improvements were also obtained. In general, Group 1 techniques performed significantly better in the noise removal task than Group 2 techniques. In the particular case of the *Colon* data set for DROp3 with $k=9$, the number of removed instances was so high that RIPPER algorithm could not build a model.

The error rates of the C4.5 algorithm for the pre-processed data sets *Leukemia*, *Lung* and *Golub* were either similar or lower than those obtained using their original data sets.

The error rates for the *ExpGen* data set were always similar to those obtained using the original data set. For the *Colon* data set, on the other hand, they were always worse.

For all three ML algorithms investigated, the complexity was reduced when the pre-processed data sets were used. For the SVMs, the smallest the pre-processed data set produced by the noise detection techniques, the lowest the number of SVs obtained and, consequently, the complexity of the model. For the C4.5 algorithm, the model complexity decreased until a lower bound from which even if the pre-processed data set presented less examples, the complexity could not be reduced.

For the RIPPER algorithm, the final models also had a lower complexity, but the reduction was smaller. For some of the data sets, the same complexity obtained by the original data set was obtained using their pre-processed versions.

The time taken by SVMs, C4.5 and RIPPER algorithms to build the induced hypothesis, when applied to pre-processed data sets, was always reduced regarding the original data sets, reaching at maximum 1 second. For the SVMs, the processing time was only slightly reduced when compared with the processing time for the original data sets.

4.2 Ensembles of Noise Detection Techniques

According to the criterion of combining only pre-processing techniques, which resulted in better ML performance after the pre-processing phase, three ensembles of pre-processing techniques were selected for each ML algorithm.

- SVMs:
 - *Leukemia*: ENN & RENN with $k=1$ and AllkNN with $k=9$;
 - *Golub*: DROPs 3 & 4 with $k=3$ and DROPS with $k=9$; and,
 - *ExpGen*: DROPS with $k=3$ and DEL with $k=1$ & $k=3$.
- C4.5:
 - *Leukemia*: ENN with $k=1$ & $k=3$ and RENN with $k=1$, $k=3$ & $k=9$;
 - *Lung*: AllkNN, ENN and RENN with $k=1$; and,
 - *Golub*: ENN and DROPs 2 & 5 with $k=9$.
- RIPPER:
 - *Lung*: AllkNN, ENN and RENN with $k=1$;
 - *Golub*: AllkNN, ENN and RENN with $k=1$; and,
 - *Colon*: AllkNN with $k=1$ and ENN & DEL with $k=3$.

The results obtained in the experiments using these ensembles can be seen in Table 3, which presents, for each ML technique, the data sets analyzed, the number of pre-processing techniques of each ensemble and the average error rates for each ensemble (Error Ensembles column). The column ‘Error Best technique’ indicates the best error rate obtained by the pre-processing techniques when applied individually. The names of these pre-processing techniques are indicated in the column ‘Best technique’.

For all experiments, the ensembles obtained better results than the application of the ML algorithms directly to the original data sets, but not all ensembles developed performed better than some pre-processing techniques applied alone. Besides, the ensembles’ results were close to the best results obtained in the previous experiments. It should be observed that for C4.5, ensembles performed better. For SVMs and RIPPER, they could not achieve the best results for the majority of the data sets analyzed.

Table 2. Error rates and standard deviation for the original and pre-processed data sets and different ML techniques

k		SVM						DT						RIPPER					
		Leukemia	Lung	Golub	ExpGen	Colon	Leukemia	Lung	Golub	ExpGen	Colon	Leukemia	Lung	Golub	ExpGen	Colon			
Original data	7.1⁺4.4	29.4⁺3.7	29.3⁺10.8	7.7⁺6.8	35.7⁺13.6	17.2⁺7.0	9.1⁺5.2	16.4⁺8.1	8.2⁺4.6	21.0⁺13.5	19.3⁺5.8	11.1⁺8.4	15.5⁺12.5	13.0⁺6.7	24.3⁺18.0				
AIKNN	1	8.0 ⁺ 3.9	29.4⁺3.7	32.1 ⁺ 13.9	7.7⁺5.1	6.1⁺5.2	17.9⁺8.9	8.2⁺7.5	26.6⁺21.4	21.5⁺7.5	7.6⁺5.4	12.5⁺12.1	9.6⁺5.9	24.1⁺13.1					
	3	8.0 ⁺ 3.9	29.4⁺3.7	34.8 ⁺ 13.9	8.2⁺3.9	35.7⁺13.6	17.5⁺5.5	13.6⁺12.7	8.7⁺4.9	24.5⁺11.9	20.0⁺9.7	13.5⁺7.4	13.0⁺7.4	26.2⁺13.9					
	9	7.9 ⁺ 3.0	29.4⁺3.7	34.8 ⁺ 13.9	8.7⁺3.8	35.7⁺13.6	18.7⁺5.6	7.6⁺4.2	12.0⁺14.2	12.5⁺5.1	30.7⁺5.3	22.4⁺10.7	9.6⁺5.8	16.6⁺12.9	15.4⁺8.8	30.7⁺5.3			
ENN	1	7.3 ⁺ 3.7	29.4⁺3.7	32.1 ⁺ 13.9	8.7⁺4.3	35.7⁺13.6	17.4⁺4.8	5.6⁺5.0	16.4⁺10.5	9.2⁺6.1	21.4⁺19.4	20.5⁺7.9	6.6⁺4.1	12.5⁺12.1	11.5⁺6.7	27.4⁺15.1			
	3	8.3 ⁺ 3.6	29.4⁺3.7	34.8 ⁺ 13.9	7.7⁺3.3	35.7⁺13.6	16.8⁺7.1	8.1⁺4.8	13.6⁺12.7	12.0⁺7.5	30.0⁺17.5	20.5⁺4.5	8.6⁺6.2	15.2⁺12.2	13.5⁺8.3	25.7⁺15.8			
	9	11.0 ⁺ 4.8	29.4⁺3.7	34.8 ⁺ 13.9	9.6⁺5.5	42.4⁺18.6	17.4⁺6.2	8.1⁺6.3	11.1⁺12.7	12.5⁺7.1	32.9⁺19.3	21.4⁺7.8	9.1⁺7.0	15.4⁺18.1	15.4⁺9.9	32.6⁺14.0			
RENN	1	7.7 ⁺ 3.9	29.4⁺3.7	32.1 ⁺ 13.9	8.7⁺4.3	35.7⁺13.6	17.7⁺4.9	6.1⁺4.6	16.4⁺10.5	9.7⁺6.3	28.6⁺23.5	22.4⁺8.1	6.1⁺4.0	12.5⁺12.1	13.4⁺8.0	27.6⁺13.9			
	3	11.0 ⁺ 5.1	29.4⁺3.7	34.8 ⁺ 13.9	7.7⁺3.3	43.3⁺19.4	16.8⁺6.7	7.6⁺4.9	13.6⁺12.7	12.5⁺7.1	40.0⁺14.9	19.6⁺7.4	10.6⁺4.7	13.8⁺11.2	13.0⁺8.0	39.1⁺20.0			
	9	13.4 ⁺ 6.3	29.4⁺3.7	34.8 ⁺ 13.9	9.6⁺5.5	42.4⁺18.6	17.4⁺6.2	8.1⁺6.3	11.1⁺12.7	12.5⁺7.1	35.5⁺13.6	22.7⁺9.9	12.1⁺7.1	30.5⁺17.3	13.9⁺8.2	35.7⁺13.6			
DROP2	1	11.3 ⁺ 3.8	77.9 ⁺ 25.8	32.0 ⁺ 19.7	7.7 ⁺ 5.1	35.7 ⁺ 13.6	37.4 ⁺ 5.4	50.1 ⁺ 19.9	37.7 ⁺ 21.0	26.9 ⁺ 21.1	46.9 ⁺ 16.2	42.8 ⁺ 8.2	65.5 ⁺ 27.7	47.0 ⁺ 18.7	55.3 ⁺ 16.6	41.7 ⁺ 20.6			
	3	14.1 ⁺ 6.5	78.5 ⁺ 27.0	24.8 ⁺ 17.7	8.6 ⁺ 6.7	46.7 ⁺ 20.0	37.6 ⁺ 5.1	47.7 ⁺ 20.4	20.6 ⁺ 15.3	42.2 ⁺ 22.5	45.2 ⁺ 11.6	50.9 ⁺ 23.4	50.2 ⁺ 17.7	55.3 ⁺ 13.1	41.4 ⁺ 15.4				
	9	8.9 ⁺ 5.1	69.3 ⁺ 29.6	23.6 ⁺ 19.7	9.2 ⁺ 6.3	39.1 ⁺ 16.7	29.9 ⁺ 7.4	44.6 ⁺ 18.9	10.5 ⁺ 13.1	27.4 ⁺ 14.4	24.4 ⁺ 14.1	32.4 ⁺ 7.1	56.8 ⁺ 15.0	17.5 ⁺ 13.3	43.5 ⁺ 9.5	27.6 ⁺ 15.9			
DROP3	1	13.4 ⁺ 5.8	79.1 ⁺ 24.7	32.0 ⁺ 16.4	8.2 ⁺ 5.1	35.7 ⁺ 13.6	42.0 ⁺ 10.9	62.4 ⁺ 17.7	33.8 ⁺ 22.4	36.1 ⁺ 20.7	48.8 ⁺ 17.7	48.6 ⁺ 10.4	74.0 ⁺ 25.5	32.0 ⁺ 13.4	64.3 ⁺ 19.6	39.1 ⁺ 20.0			
	3	11.3 ⁺ 3.2	78.1 ⁺ 26.8	20.7 ⁺ 15.1	8.2 ⁺ 6.0	35.7 ⁺ 13.6	32.5 ⁺ 7.2	41.4 ⁺ 20.5	24.7 ⁺ 11.7	35.7 ⁺ 14.4	46.6 ⁺ 27.9	43.7 ⁺ 10.1	50.1 ⁺ 25.7	32.1 ⁺ 15.1	45.0 ⁺ 15.0				
	9	14.4 ⁺ 4.7	61.7 ⁺ 35.1	48.0 ⁺ 22.2	8.7 ⁺ 4.9	53.3 ⁺ 20.0	27.8 ⁺ 6.1	35.0 ⁺ 5.7	19.5 ⁺ 14.6	21.1 ⁺ 16.4	43.3 ⁺ 18.0	30.5 ⁺ 5.3	38.5 ⁺ 15.9	31.3 ⁺ 17.5	43.0 ⁺ 14.9				
DROP4	1	14.1 ⁺ 5.6	72.6 ⁺ 28.3	32.0 ⁺ 16.4	7.4 ⁺ 5.7	39.1 ⁺ 16.7	36.6 ⁺ 7.1	60.4 ⁺ 16.8	33.8 ⁺ 22.4	34.6 ⁺ 22.1	43.1 ⁺ 23.0	49.8 ⁺ 14.3	68.6 ⁺ 27.2	32.0 ⁺ 13.4	62.8 ⁺ 18.8				
	3	13.5 ⁺ 7.3	71.2 ⁺ 30.5	21.1 ⁺ 13.7	9.6 ⁺ 6.8	46.7 ⁺ 20.0	32.2 ⁺ 9.8	47.3 ⁺ 15.6	19.1 ⁺ 11.8	45.1 ⁺ 14.7	43.3 ⁺ 10.6	43.5 ⁺ 10.3	46.2 ⁺ 26.0	33.4 ⁺ 15.3	51.3 ⁺ 16.9				
	9	8.9 ⁺ 5.3	11.1 ⁺ 28.0	33.9 ⁺ 25.5	8.2 ⁺ 5.5	35.7 ⁺ 13.6	31.5 ⁺ 7.9	44.7 ⁺ 16.7	26.4 ⁺ 14.2	23.0 ⁺ 17.4	32.4 ⁺ 17.2	37.0 ⁺ 8.9	44.7 ⁺ 19.9	31.4 ⁺ 22.7	39.2 ⁺ 12.8				
DROP5	1	11.0 ⁺ 5.5	42.3 ⁺ 26.7	22.3 ⁺ 19.3	7.7 ⁺ 6.4	46.7 ⁺ 20.0	42.5 ⁺ 5.6	43.4 ⁺ 21.2	26.6 ⁺ 15.2	38.5 ⁺ 21.2	40.6 ⁺ 18.3	51.4 ⁺ 10.6	34.2 ⁺ 21.9	41.8 ⁺ 18.9	73.4 ⁺ 24.5				
	3	12.8 ⁺ 7.4	36.3 ⁺ 20.9	29.5 ⁺ 22.0	6.7 ⁺ 5.1	35.7 ⁺ 13.6	36.2 ⁺ 11.3	38.5 ⁺ 12.8	31.0 ⁺ 20.3	44.8 ⁺ 24.0	20.7 ⁺ 17.3	50.1 ⁺ 6.9	35.6 ⁺ 7.5	32.0 ⁺ 16.4	71.6 ⁺ 15.1				
	9	8.3 ⁺ 2.5	48.1 ⁺ 30.9	21.3 ⁺ 15.5	8.2 ⁺ 5.0	35.7 ⁺ 13.6	31.5 ⁺ 9.2	31.8 ⁺ 12.2	11.1 ⁺ 11.3	25.5 ⁺ 15.2	44.7 ⁺ 23.2	34.0 ⁺ 8.1	34.1 ⁺ 13.5	20.5 ⁺ 11.0	47.3 ⁺ 20.0				
DEL	1	13.4 ⁺ 5.9	83.9 ⁺ 21.3	45.4 ⁺ 26.2	6.2 ⁺ 6.8	40.0 ⁺ 17.3	45.0 ⁺ 10.7	49.1 ⁺ 16.1	16.1 ⁺ 16.1	69.2 ⁺ 23.6	46.6 ⁺ 25.0	53.8 ⁺ 7.4	60.1 ⁺ 13.8	53.0 ⁺ 19.9	83.1 ⁺ 8.2	38.8 ⁺ 16.0			
	3	14.3 ⁺ 6.0	73.3 ⁺ 30.7	36.4 ⁺ 26.6	5.7 ⁺ 4.4	42.4 ⁺ 18.6	35.2 ⁺ 11.2	51.7 ⁺ 13.8	34.8 ⁺ 20.0	40.2 ⁺ 21.0	31.9 ⁺ 17.9	51.4 ⁺ 6.9	53.3 ⁺ 22.0	44.6 ⁺ 22.5	67.6 ⁺ 12.0				
	9	9.2 ⁺ 5.0	52.5 ⁺ 31.3	36.4 ⁺ 27.4	8.2 ⁺ 5.0	45.7 ⁺ 19.8	30.9 ⁺ 5.2	40.7 ⁺ 12.1	21.5 ⁺ 18.2	29.9 ⁺ 19.4	32.7 ⁺ 21.2	33.3 ⁺ 8.7	40.9 ⁺ 25.6	38.0 ⁺ 25.6	32.8 ⁺ 14.6				

Table 3. Results obtained with ensembles developed for SVMs, C4.5 and RIPPER algorithms

ML algorithm	Data Set	# techniques	Error Ensembles	Error Best technique	Best Technique
SVM	Leukemia	3	7.3+-3.7	7.3+-3.7	ENN $k=1$
	Golub	3	30.7+-26.1	21.1+-13.7	DROP4 $k=3$
	ExpGen	3	6.7+-6.5	5.7+-4.4	DEL $k=3$
C4.5	Leukemia	5	16.2+-6.7	16.8+-5.0	RENN $k=9$
	Lung	3	5.1+-4.1	5.6+-5.0	ENN $k=1$
	Golub	3	13.6+-12.3	11.4+-11.3	DROP5 $k=9$
RIPPER	Lung	3	8.1+-5.3	6.1+-3.9	RENN $k=1$
	Golub	3	12.3+-10.0	12.5+-12.1	AIKNN, ENN, RENN $k=1$
	Colon	3	26.9+-17.7	19.5+-21.7	DEL $k=3$

Another important point is the reduction they imposed to the initial problem complexity. As in the previous experiments, the complexity of the ML models was reduced for the pre-processed data sets, as well as their training times.

The analysis of experiments presented in this paper suggests that the ML techniques investigated improved their classification accuracy when they used pre-processed data sets, particularly for the C4.5 and RIPPER algorithms. For the SVMs, the new results were slightly better with lower complexity. It is also possible to note that the results vary according to the noise detection technique, data set and ML algorithm used.

Therefore, the pre-processing techniques investigated were able to induce simpler classifiers while keeping the most expressive patterns of the data sets.

5 Conclusions

This paper investigated the application of distance-based pre-processing techniques in different gene expression problems. Three ML algorithms, trained over the original and pre-processed data sets, were employed to evaluate the power of the presented techniques in keeping the most informative patterns.

The results observed indicate that the noise detection techniques employed were effective in the noise detection process, specially the Group 1 techniques. The Group 2 techniques presented mainly higher error rates, possibly because they removed a large amount of data from the initial data sets.

For the Group 1 techniques, experiments resulted not only in simplifications of the induced classifiers but also in lower error rates, specially for the C4.5 and RIPPER algorithms, where another advantage was higher comprehensiveness.

Ensembles of pre-processing techniques also presented good performance, some of them even better than the best results obtained by the pre-processing techniques alone. One advantage of using ensembles is to obtain satisfactory results, very similar or even better than the best results obtained by the pre-processing techniques alone, reducing the problem complexity and eliminating the need to test and evaluate which pre-processing technique is better for a given data set.

The authors are now investigating new techniques for noise detection and developing other approaches for the ensembles, aiming to further improve the gains obtained by the identification and removal of noisy data.

Acknowledgments. The authors acknowledge the support from FAPESP and CNPq.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 420–434. Springer, Heidelberg (2000)
2. Cohen, W.W.: Fast effective rule induction. In: Proc. 12th Int. Conf. on Machine Learning, pp. 115–123 (1995)
3. Collobert, R., Bengio, S.: SVM Torch: Support vector machines for large-scale regression problems. *J. Machine Learning Res.* 1, 143–160 (2001)
4. Demsar, J.: Statistical comparisons of classifiers over multiple datasets. *J. Machine Learning Research* 7, 1–30 (2006)
5. Frank, E., Witten, I.H.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2005)
6. Hodge, V., Austin, J.: A survey of outlier detection methodologies. *Artificial Intelligence Review* 22, 85–126 (2004)
7. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. *The VLDB Journal* 8(3-4), 237–253 (2000)
8. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)
9. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
10. Tang, J., Chen, Z., Fu, A.W., Cheung, D.: A robust outlier detection scheme in large data sets. In: Proc. 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (2002)
11. Tomek, I.: Two modifications of CNN. *IEEE Transactions on Systems, Man and Cybernetics* 7(11), 769–772 (1976)
12. Vapnik, V.N.: The Nature of Statistical Learning Theory, 2nd edn. Springer, Heidelberg (1995)
13. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. *Machine Learning* 38(3), 257–286 (2000)
14. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. *J. Artificial Intelligence Research* 6(1), 1–34 (1997)
15. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics* 2(3), 408–421 (1972)

FES Position Control of Forearm Using EOG

Ken Suetsugu¹, Yoshihiko Tagawa¹, Tomohisa Inada¹, and Naoto Shiba²

¹ Kyushu Institute of Technology, Japan

g344208k@tobata.isc.kyutech.ac.jp

² Kurume University, Japan

Abstract. In recent years, the number of individuals with disabled motor functions has increased. However, restoring motor functions is possible by providing electrical stimulation to the peripheral nerves. Flexible operation methods are desirable for people with central nerve disorders as they can be used in daily life despite limited residual motor functions. Various examples using electromyogram (EMG), joint movement or breathing to control functional electrical stimulation (FES) have been reported. The present work investigates eye direction recognition experiments and the possibility of controlling the angle of the forearm using electrical stimulation. The target angle for the forearm is determined using electrooculogram (EOG) to allow more natural operation. In addition, PD controller and disturbance observer are applied to realize forearm movement and controlled stimulation.

1 Introduction

In recent years, as the number of cerebrovascular diseases has increased, so has that of individuals with disabled motor functions. People with spinal cord injuries also have disabled functions and they are unable to move their extremities voluntarily. They need long-term therapy or rehabilitation to recover and even then we cannot expect there to be a perfect recovery. Countermeasures for disuse syndrome due to function loss are challenging issues for rehabilitation in hospitals or at home.

Disuse syndrome covers ailments such as muscular atrophy, articular contracture, and circulatory disorders. Functional electrical stimulation (FES) can contribute to reducing the severity of disuse syndrome by contracting individual muscles and also by restoring lost motor functions. A wide range of people with central nerve disorders can use FES to restore motor functions, it is therefore necessary for FES control to be flexible and adaptable to many situations. There are many works in which electromyogram (EMG), joint movement, or breathing have been applied for FES control [1, 2]. For example, when people want to catch an object, they watch it carefully, and then catch it by hand. Using electrooculogram (EOG) flexible operation to control individual upper limbs is another means to achieve movement of the upper limbs in any position.

This paper proposes a new method to get the targeted elbow angle using eye direction recognition. We attempted to apply the forearm targeted angle to a desired value of FES control. It is necessary to reduce the impact of the upper arm movement on the

forearm, which is a reaction due to the FES of the forearm and it can be recognized as one of disturbances. A disturbance observer is applied to cancel out the disturbance as fast as possible and to compensate for the uncertainty between the control model and the actual limbs. Experimental results suggested that several control methods were useful and EOG was a suitable method to give effective control commands.

2 Practical Position Control for Paralyzed Upper Arm

Individuals with paralyzed upper limbs who want to grasp an object in front of them can recognize how far their hands are located from the object and make some strategies to control the position of their hands using an FES system. Since such a situation can occur anywhere, a control system that can use portable instruments would be indispensable.

A method of EOG processing not dependent on the environment is a promising approach for giving control commands. The distance between hand and object is monitored, and the hand is controlled to move closer to the object inch by inch. The processed EOG could have several modes such as forward – backward, up – down, and right – left. Control commands corresponding to each mode could be sent to muscles in the upper limbs through FES instruments. This paper will propose a basic principle for controlling the forearm position.

2.1 EOG

The EOG has been applied to the input device of a computer and the handling of a wheelchair [3, 4]. In this paper the EOG is used to decide the target angle of the elbow joint of individuals who have paralyzed upper limbs. The advantages of the EOG are easy to quantify: the measurement device is inexpensive and this method causes little fatigue. On the other hand, influences of artifacts such as EEG, blinking, saccade and the drift of the baseline are some major defects which need to be addressed.

Figure 1 shows the pasting position of the surface electrodes to measure the EOG, where A is the ground, B and C are electrodes to detect the EOG of the vertical direction, and D and E electrodes detect the EOG of the horizontal direction. In our system, the signals are sampled 100 times per second. DSP is used for signal processing and Ag-AgCl electrodes are used as surface electrodes.

2.2 Processing EOG

The electric potential of the EOG is about 50 to 3500 μ V with a frequency from DC to 100 Hz. EOG signal processing was performed as shown in the flow chart of Fig. 2.

The first step is filtering processing. The low and high pass filters have a fourth order to remove the offset and to limit the range of frequency, respectively.

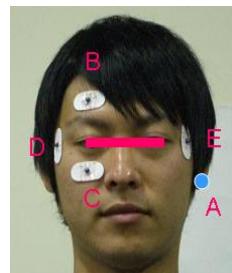


Fig. 1. Attached surface electrodes to detect the EOG

The second step is the removal of the "blink" signals using the blanking processing [5]. we remove the blink signal's large voltage when the derivation EOG is measured. The blink EOG level is about four times bigger than the normal EOG. Signals more than the threshold are therefore considered as input. The blanking process for blinking signals is used and the outputs are 0 V for 0.6 s. The characteristic of the blink EOG is used for determination and cancellation of the target angle.

The third step is to convert the differential calculus EOG into the rectangle EOG. When the EOG is more than +1 V (upward gaze), the output is +4 V. When input is equal or less than -1 V (downward gaze) the output is +2 V.

The fourth step is the blanking process to remove noise which occurs several times in a row during the short period of time when the rectangle EOG signal is input. Noise occurs with eyeball movements. In this experiment, input of 0 V is held for 0.6 s after noise occurs in order to detect only the first movement and limit other input.

The fifth and final step is to decide and permit the target angle of the input with the blink EOG which was removed in the second step. The system admits input when blinks are input 4 times in 1.5 s. Angle inputs in the meantime are recognized as the targeted angle. The targeted angle is reset when blinks are input 5 times in 1.5 s.

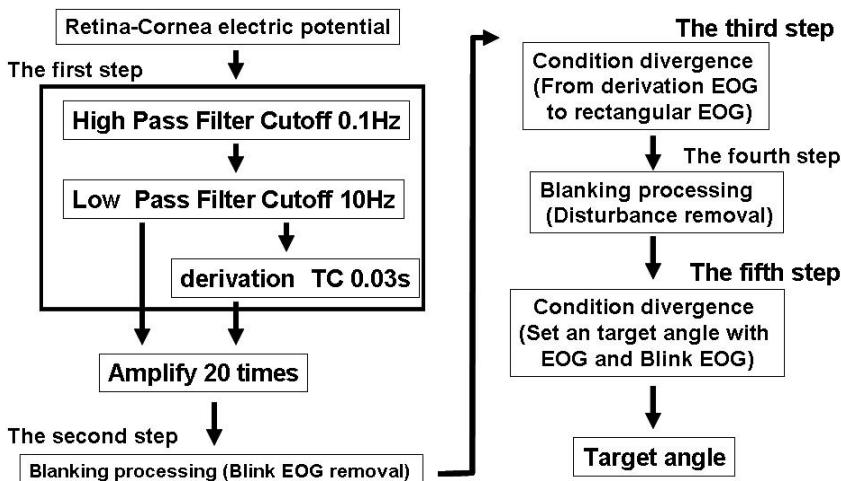


Fig. 2. Flow of EOG processing

3 Control of the Musculoskeletal System

Modeling the human body is a complex task, because many parameters are needed and some cannot be measured directly. Therefore the musculoskeletal model is simplified using the parameters which we can estimate non-invasively. The human body was controlled in the experiment using this musculoskeletal model.

3.1 Relationship between Stimulation Voltages and Muscle Forces

The musculoskeletal model of the upper limbs is derived from the relationship between electrically induced muscular contraction forces and stimulation intensity. The input is stimulation voltage [V] and the output is tension [Nm]. The experiment is performed under the condition of a fixed upper arm in a horizontal plane. The torque output is measured for three postures of the elbow joint: (1) at a 180 degree angle (2) at a 135 degree angle (3) at a 90 degree angle. The musculoskeletal model is developed as linearly approximated data of the average of these three data. Electrical stimulation is carried out on the biceps brachii muscle, brachial muscle and the triceps brachii muscle, and the torque is measured for each extension and flexion. Electrical stimulation is carried out for each muscular threshold, and voltage is increased up to a painful stimulation level. The electrical stimulation generator used in this study utilizes a voltage control method which was developed by one of the authors of this paper. In this experiment, a rectangular wave with a carrier frequency of 5000 Hz and a burst-wave of a stimulation frequency of 20 Hz are used. The maximum generated voltage of the stimulation generator is 63.5 V [6].

3.2 Disturbance Observer

PD control using a disturbance observer [7] is adopted in this system. The disturbance observer estimates error terms, disturbance and modeling uncertainty from control input and feedback. The disturbance observer also decreases the influence of error through feedback. PID, adding integration action to the PD, is another reasonable means to compensate for lack of information due to subtracting the output from the target value. However, PID control is not able to design both the target value and disturbance responses independently. To improve the reduction of disturbance response, the proportional gain of the PID controller has to be a large value. A PID system easily becomes unstable when it is disturbed. When using a disturbance observer, the controlled system (i.e., the human forearm) is replaced by a nominal model. The observer can estimate external forces, modeling error and parameter changes. Therefore we utilized a disturbance observer.

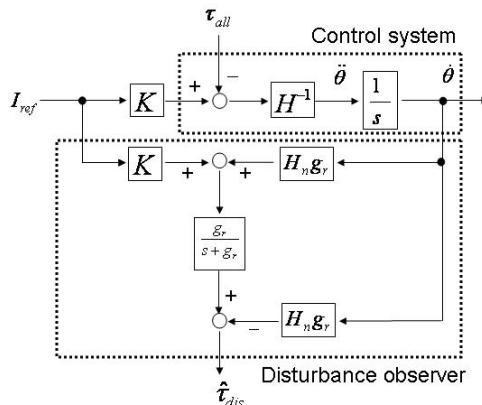


Fig. 3. Disturbance observer

Figure 3 shows the control system including disturbance observer. The parameters of the figure are as follows: τ_{alt} is the disturbance torque, H is the inertia of plant, $\ddot{\theta}$ is the angular acceleration, $\dot{\theta}$ is the angular velocity, K is the torque coefficient, and I_{ref} is the reference value. The disturbance observer includes a low pass filter, and the disturbance can be estimated as follows:

$$\hat{\tau}_{dis} = \frac{g_r}{s + g_r} (I_{ref} K + H_n g_r \dot{\theta}) - H_n g_r \dot{\theta} \quad (1)$$

LPF is set as a first-order delayed system. From the feedback of estimated disturbance provided by the disturbance observer, a robust angle control system can be developed for the human body (in this case a nominal model thereof).

3.3 Control System of a Forearm Flexion Using EOG

The goal of this study is to input the target angle using oculomotor information (EOG), and to control the elbow joint angle. The concept of the control system is shown in Fig. 4. First, the EOG is measured by the controlled system (human forearm). The information is sent to the computer (DSP). Then, the stimulation intensity is calculated by a disturbance observer and PD control after angle information is measured using an encoder attached to the elbow joint. The stimulation is accomplished with conductive gel electrodes which are placed over the motor points of the stimulated muscles. The controlled system is a human forearm (biceps brachii muscle, brachial muscle and triceps brachii muscle). Because the actuator is human muscle, a manipulated variable has 35 V and is limited for safety.

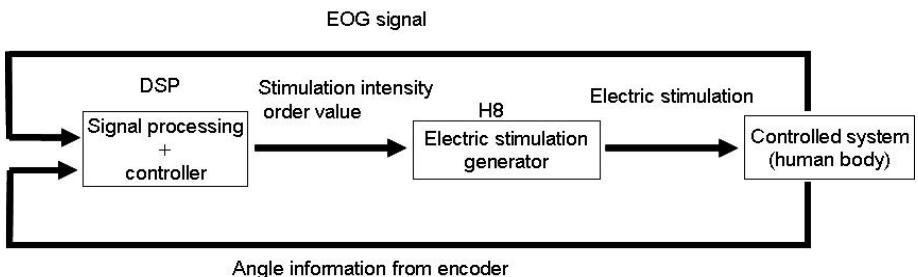


Fig. 4. System scheme

Figure 5 shows the block diagram of the control system. The main symbols are as follows: r is the target angle from the EOG; d is the disturbance torque, \hat{d} is the estimated disturbance torque; θ is the output angle of elbow joint; Q is the first order low pass filter having $1/g_r = 0.003$; T is the torque equation obtained from the experiment and the equations of flexion and extension are τ_F and τ_E , respectively:

$$\begin{aligned}\tau_F &= -0.14V + 1.01 \\ \tau_E &= 0.11V - 2.03\end{aligned}, \quad (2)$$

where V is the stimulation intensity; and τ is the muscular torque. The controlled system is the human body forearm and H_n is the nominal value of the forearm equal to 0.86. The parameters of the PID are determined using CHR method from the step response of the forearm to some stimulation voltages in the horizontal plane, $K_P = 0.57$, $K_I = 6.57 \times 10^{-4}$, and $K_D = 7.66$. For the PD control, the same K_P and K_D are used.

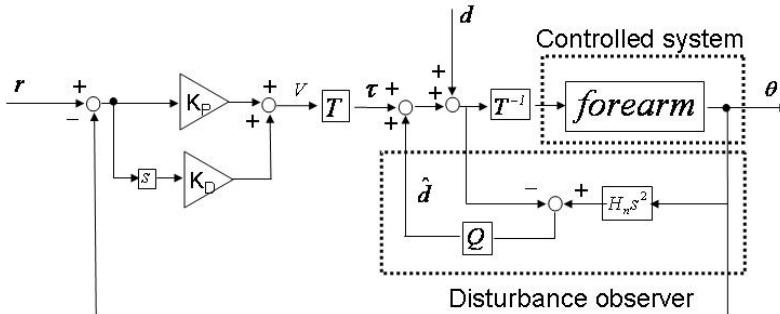


Fig. 5. Block diagram of PD control system with disturbance observer

4 Experimental Results

4.1 Definition of Forearm Flexion by EOG

To determine whether the input method using EOG can set the target angle, we perform a target angle input experiment. Electrodes for measurement are affixed as shown in Fig. 1 above. The EOG of the up and down direction is measured in order to detect the blink EOG. It is thought that a blink signal occurs as a result of the change in both the capacitance and the resistance caused by a crossover between the

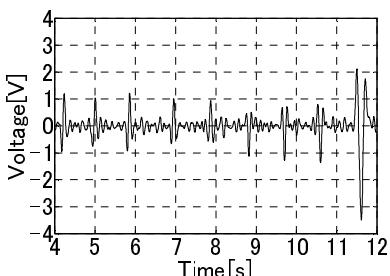


Fig. 6. EOG of processing step 1

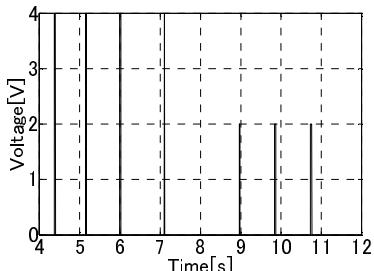


Fig. 7. EOG of processing step 4

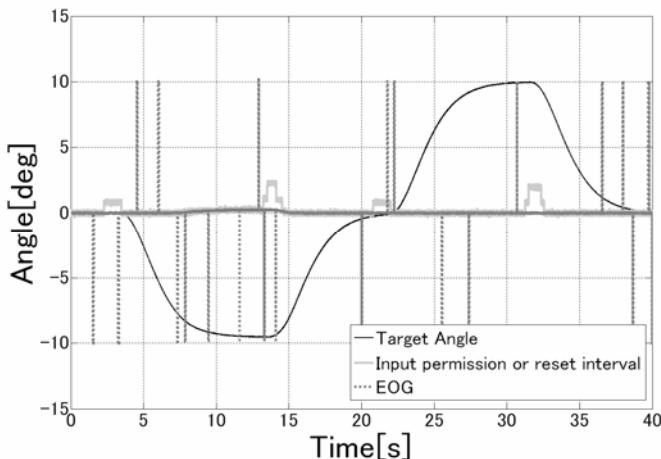


Fig. 8. Target angle input

electrodes measurement changes by a blink. Three sequential eyeball movements (bottom, top and bottom) are measured. The test subject is one normal healthy man. Figure 6 shows the result of the EOG using processing step 1, and Fig. 7 shows the EOG of processing step 4. From Fig. 6 it can be seen that the swing of the base line and big waveform caused by the blink EOG are confirmed. It is also found that only the EOG of the up and down direction can be extracted in Fig. 7.

Figure 8 shows the determination of the target angle using EOG processing and EOG blinking. A blink is an input permission switch, where +10 degree is input when we look up within the input permission interval and -10 degree is input when we look down. In addition, the target angle can be reset by blinking five times during the reset interval of 1.5 sec. The system does not recognize any input when the eyeballs are moved, excluding input permission intervals. The figure shows that the test subject decides upon the target angle using oculomotor activity and blinks.

4.2 Practical Control of Paralyzed Upper Limbs

Figure 9 shows our target angle follow-up control experiment on the elbow joint using the electrical stimulation. Electrical stimulations are carried out on biceps brachii, brachial muscle and triceps brachii muscle. During the experiment, the elbow is put on the table and the forearm moves in the vertical plane (flexural way is minus, and extension way is plus).



Fig. 9. Angle control of elbow joint using electrical stimulation

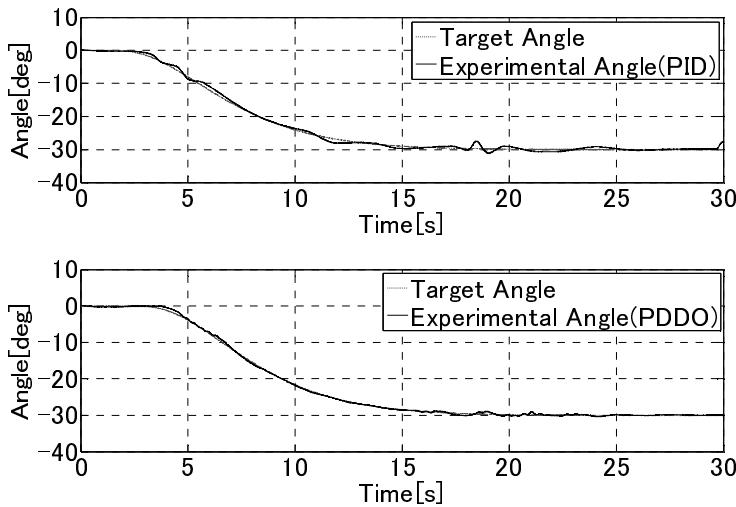


Fig. 10. Target angle follow-up control experiment on elbow joint in vertical plane using PID control and PDDO control

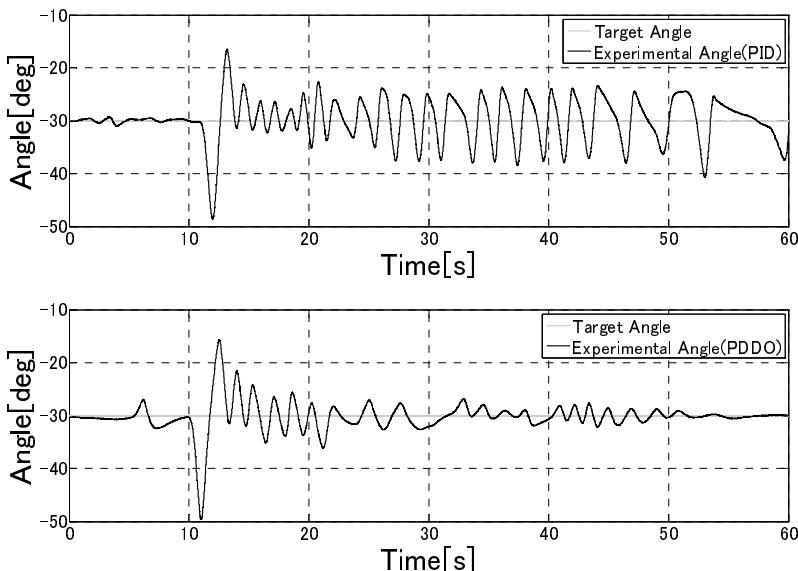


Fig. 11. PID and PD control output at time of disturbance input

To determine whether the PID control can regulate the output well under the disturbance, we compared the PID control to the PD control with disturbance observer (PDDO). Offset is expected when using the PD control. Therefore the elbow joint angle is controlled by the PID control and PDDO control. Figure 10 shows the

experimental results of the PDDO control and the PID control for the flexion of the elbow joint at a target angle of -30 degrees. Although the system comes under the influence of gravity, the difference between the target angle and the experimental results is minimal. In Fig. 11, an external force was applied as disturbance on the forearm at 10 s after having converged to the target angle of -30 degree. Unlike the PDDO, the response of the PID control became undamped continuous oscillation after the disturbance. Thus, the PDDO has greater robustness against disturbance and was therefore used in our control system.

When a test subject wants to move his forearm several degrees using the EOG, he moves his eyes up or down. Figure 12 shows the experimental result when the target angle is -10 degree and shows a good follow-up feature. Step desired values are modified into second forms to avoid a high intensity of electrical stimulation on the human muscles.

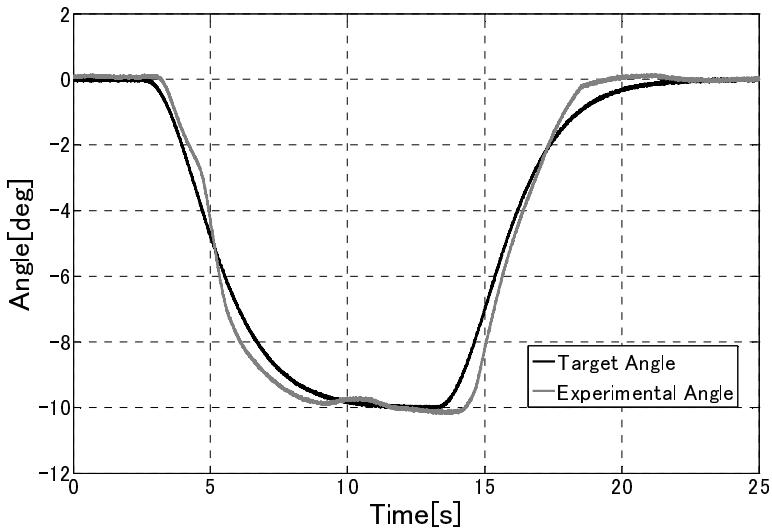


Fig. 12. Target angle input with EOG and angle control of elbow joint

5 Discussion and Conclusion

We developed a system of target angle follow-up control using electrical stimulation, and set the target angle using EOG. When a test subject blinked voluntarily, the system could recognize the input of the EOG. However, this system prevents input signals excluding the permission interval, as shown in Fig. 8. Thus, the blink input can determine the target angle. The distance from the subject's hand to an object can be shortened gradually with the EOG. The PD control with disturbance observer realized the desired elbow angle and maintained that angle against disturbance. In contrast, the experiment with the PID control took much longer to converge when a large disturbance was input to the control system. The controller using a disturbance observer reduced the effect of disturbance much more effectively. Experimental results suggest

that a control strategy using a disturbance observer is effective even if the controlled object has uncertainties. In this system, a zero-order disturbance was assumed, and higher order LPF will be needed to improve performance of the disturbance observer. The design of the low pass filter is also of great important to repress disturbance.

In conclusion, this system using the electrical stimulation to control the elbow joint and set the target angle using the EOG operated well. In future work, we will control two degree of freedom systems, which include not only the elbow joint, but also a shoulder joint, and will also investigate a better control method.

Acknowledgments

This study was partly supported by a Japanese Society for the Promotion of Science Grant-in Aid for scientific research (No.20360118).

References

1. Handa, T., Kameyama, J., Takahasi, H., Handa, Y., Hosimiya, N.: Practical use and control commands of the FES system. *Biomechanism* 11(25), 285–292 (1992)
2. Handa, Y., Hosimiya, N.: Restoration of upper extremity function by electrical neuromuscular stimulation. *Biomechanism* 9(6), 75–82 (1988)
3. Fang, W., Itakura, N.: Evaluation for measurement of continual eye-gaze shift for the eye-gaze input interface using electro-oculogram amplified by AC coupling -comparison with limbus tracking method, The institute of electronics, information and communication engineers (IEICE)Technical report, pp. 41–44 (2006)
4. Rafael, B., Luciano, B., Mznuel, M., Elena, L.: System for assisted mobility using eye movement based on electrooculography. *Transactions on neural systems and rehabilitation engineering (IEEE)* 10(4), 209–240 (2002)
5. Merletti, R.: Electrically Evoked Myoelectric Signals. *Critical Reviews in Biomedical Engineering* 19, 293–340 (1992)
6. Ito, T., Tagawa, Y., Shiba, N., et al.: Development of practical and effective hybrid exercise for use in weightless environment. In: The 26th annual conference of the IEEE EMBS, San Francisco, CA, USA, pp. 4252–4255 (2004)
7. Natori, K., Tsuji, T., Ohnishi, K., Ales, H., Karel, J.: Robust bilateral control with internet communication. In: The 30th annual conference of the IEEE industrial electronics society, Busan, Korea, pp. 2321–2326 (2004)

Reduction of FPs for Lung Nodules in MDCT by Use of Temporal Subtraction with Voxel-Matching Technique

Yoshinori Itai¹, Hyoungseop Kim¹, Seiji Ishikawa¹,
Shigehiko Katsuragawa², and Kunio Doi³

¹ Kyushu Institute of Technology, Japan

itai@kimlab.cntl.kyutech.ac.jp

² Kumamoto University, Japan

³ The University of Chicago, USA

Abstract. Detection of subtle lesions on computed tomography (CT) images is a difficult task for radiologists, because subtle lesions such as small lung nodules tend to be low in contrast, and a large number of CT images must be interpreted in a limited time. One of the solutions to the problem, a temporal subtraction technique, has been introduced in the medical field as 2D visual screening. The temporal subtraction image is obtained by subtraction of a previous image from a current one, and can be used for enhancing interval changes on medical images by removing most of the normal background structures. In this study, we have developed a method for computerized detection of lung nodules by using temporal subtraction with a voxel-matching technique in multidetector-row CT (MDCT) images. First, the candidates for nodules were detected by use of a multiple threshold technique based on the pixel value in the temporal subtraction image obtained by the voxel-matching technique. Next, a number of features were quantified, and some false positives were removed by a rule-based method with an artificial neural network. We applied our computerized scheme to 6 chest MDCT cases including 94 lung nodules. Our scheme for detecting lung nodules provided a sensitivity of 71.2 % for lung nodules with sizes under 20 mm, with 9.8 and 11.5 false positives per scan on the consistency test and validation test, respectively.

Keywords: Computer-aided diagnosis, Lung nodule, MDCT, Temporal subtraction, ANN.

1 Introduction

Detection of subtle lesions on computed tomography (CT) images is a difficult task for radiologists, because subtle lesions such as small lung nodules tend to be low in contrast, and a large number of CT images must be interpreted in a limited time. One of the solutions to the problem, a temporal subtraction technique, has been introduced in the medical field as 2D visual screening. The temporal subtraction image is obtained by subtraction of a previous image from a current one, and can be used for enhancing interval changes on medical images by removing most of the normal background structures [1]. Temporal subtraction of chest radiographs has been

developed and shown to be effective in improving the accuracy and speed of interpretation of lung nodules in numerous observer tests [2]. Moreover, some computer-aided diagnosis (CAD) systems with temporal subtraction of time series in chest radiographs have been developed. In addition, temporal subtraction of CT can be useful for detecting and analyzing the interval changes of lung nodules (such as shapes of new lesions and temporal changes in existing nodules). In the temporal subtraction of CT that we have proposed, it became possible to obtain a temporal subtraction image in which subtraction artifacts due to displacement and/or deformation of normal structures such as lung vessels were substantially removed by use of a voxel-matching technique [3]. However, some subtraction artifacts still remained, and it was necessary to detect only lung nodules from the temporal subtraction image. So far, computerized detection of lung nodules with time series in CT images has not been proposed.

In this paper, we have developed a method for computerized detection of lung nodules by using the temporal subtraction images in multidetector-row CT (MDCT). First, the candidates for nodules were detected by use of a multiple-threshold technique based on the pixel value in the temporal subtraction image, which was obtained by use of temporal-subtraction with the voxel-matching technique [3]. Next, a number of features were quantified, and some false positives were removed by a rule-based method with an artificial neural network (ANN). The rest of this paper is organized as follows. In section 2, computerized schemes of nodule detection are presented. In section 3, experimental results are summarized. Finally, a discussion and concluding remarks are given in section 4.

2 Overall Scheme for the Detection of Lung Nodules in MDCT

2.1 Computerized Scheme of the Detection of Lung Nodules in MDCT

Figure 1 shows the overall scheme for detection of lung nodules in this study. Our computerized scheme for detection of nodules included three steps, mainly: applying temporal subtraction with the voxel-matching technique for enhancing nodules and suppressing vessels, detecting nodule candidates from the temporal subtraction image, and refining the nodule candidates. We describe the details of each step in the following sections.

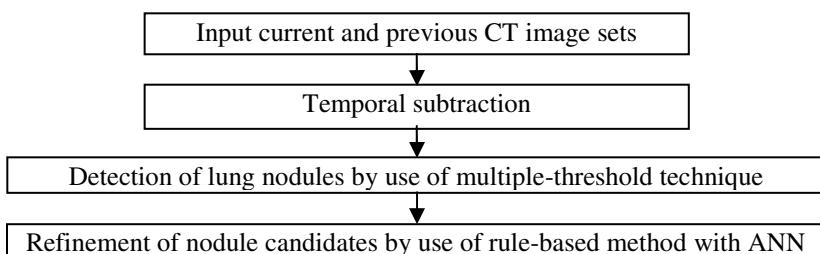


Fig. 1. Overall scheme for detection of lung nodules

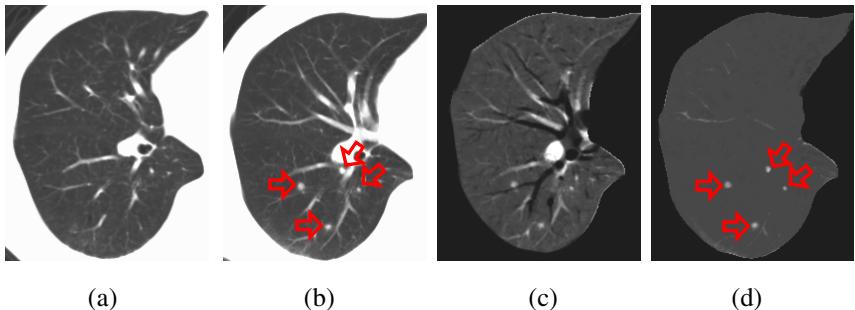


Fig. 2. Temporal subtraction for MDCT image. Four areas arrows show nodules in the current CT image in (b). ((a) and (b) show original previous and corresponding current CT image, respectively. (c) and (d) show temporal-subtraction image without and with image registration by use of our previous method [3], respectively.)

2.2 Temporal Subtraction of Chest MDCT

In a temporal subtraction, image registration between the current and the previous image is the most important process, because mis-registration of normal structures can cause subtraction artifacts and false-positive lung nodules. In [3], we proposed a new temporal-subtraction method, and we achieved enhancement of nodules with substantial removal of subtraction artifacts from temporal-subtraction images by deforming the entire lung region, pleura, air tube, and lung vessels with the voxel-matching technique. By use of the temporal-subtraction image for detection of lung nodules, the number of false positives can be reduced. In this paper, first, we applied temporal subtraction [3] for time-series MDCT images for enhancing nodules and suppressing vessels.

The CT value under the pleura tends to be high because blood and lymph are gathered there by the gravity effect. These inhomogeneous background components might cause some subtraction artifacts in the temporal-subtraction image. For eliminating inhomogeneous background components, a background trend correction was applied on the original current and previous images by use of an opening operation of the morphologic filter before subtraction. In the opening operation, we used a circle with a size of 15 pixels in diameter as a kernel. By applying opening operation, background component such as the structures with size over kernel is remained. Therefore, inhomogeneous background components can be removed by subtraction of the processed images from the original images.

Figure 2 shows temporal subtraction images obtained from an MDCT image. In Fig. 2, (a) and (b) show a previous CT image and the corresponding current CT image, respectively. (c) and (d) show the temporal-subtraction image without and with registration, obtained from current and previous CT image sets. As shown in Fig. 2, most of the lung vessels were eliminated from the temporal-subtraction image, and the temporal changes in nodules (indicated by arrows) were enhanced. However, it was found that some subtraction artifacts still remained in the temporal subtraction image (Fig. 2 (d)). For detecting only lung nodules in the temporal-subtraction image, the following procedures were performed.

2.3 Detection of Lung Nodules by Use of Multiple-Threshold Technique

Lung regions were segmented in the temporal-subtraction image by use of a threshold technique based on the CT value in the original current image and morphologic operation [4] before detection of the final lung nodules. For detection of lung nodules from temporal-subtraction images, multiple threshold technique and shape feature analysis were applied based on the assumption that nodules newly appearing in the current image set might have a high CT value and look like a circle on the temporal-subtraction image. In this technique, some binary images were obtained from a temporal-subtraction image by use of thresholds. In this study, the thresholds were the CT values corresponding to the top 0.5, 1.0, 1.5, 2.0, and 2.5 percentages of area of histogram obtained from temporal-subtraction image. Then three features, area, circularity, and complexity, were quantified for each connected component in the binary image, and nodule candidates were detected by use of a rule-based method. In the rule-based method, we set the thresholds for three features and each thresholds were difference in the detection of lung nodules with sizes over and under 20 mm. In the detection of lung nodules with size over 20 mm, the detected nodule candidates might include many false positives due to subtraction artifacts. To eliminate these false positives, we quantified some features for each candidate, and refinement by use of a rule-based method with ANN was performed.

2.4 Refinement of False Positives from Nodule Candidates

Before applying feature analysis, we segmented the candidates for lung nodules to identify their shape. Many segmentation methods for lung nodules have been proposed for measuring the size and shape features and estimating their location. In this study, for simplification, the candidates were segmented by use of a threshold technique based on the CT value in the current CT image and distance transformation [5]. In the previous step of our algorithm, some candidates for lung nodules were detected from the temporal-subtraction image. In this step, at first, the gravity point of a candidate was calculated in the temporal subtraction image and was determined as a seed point in the current image. With use of this seed point, the candidates were segmented by use of the threshold of the CT value and distance transformation. Figure 3 shows some results of segmentation of nodule candidates. As shown in Fig. 3, nodule candidates including lung nodules and lung vessels were segmented correctly.

Eight features, including four intensity features and four shape features, were quantified for segmented nodule candidates. Four intensity features consisted of the average, variance, maximum value, and averaged maximum value. On the other hand, four shape features consisted of circularity, complexity, elongation [5], and the output of dot enhancement filter [6]. Generally, lung vessel might be elongated and look like a line. On the other hand, a nodule might be a circle and look like a ball. Therefore, the output of dot enhancement filter in vessels may show a low value and that in nodules may show a high value. The output of 2-D dot enhancement filter is related to eigenvalues of a 2-D Hessian matrix of the pixel in an image. Figure 4 shows relationship between circularity, complexity and elongation of true positives

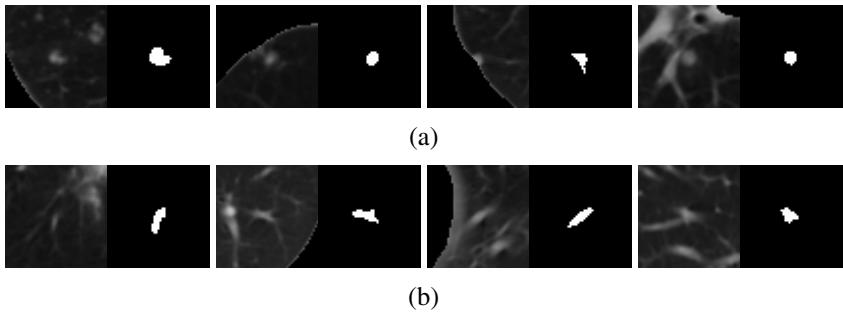


Fig. 3. Results of segmentation of nodule candidates. (a) and (b) show results of segmentation of nodule candidates of lung nodules and lung vessels, respectively. In each pair, left and right figure show original CT image and segmented nodule image, respectively.

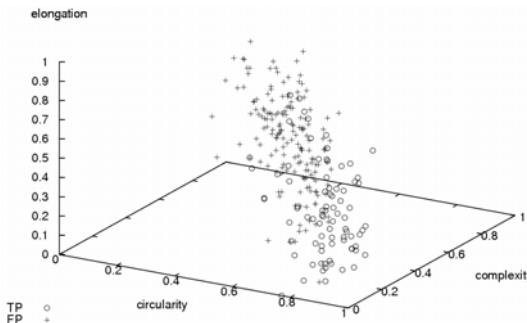


Fig. 4. Relationship among circularity, complexity, and elongation of true positive (TP) and false positive (FP)

and false positives. As shown in Fig. 4, it can be possible to separate true positives and a part of false positives. In the following step, we attempted to remove some false positives from nodule candidates in the feature space.

To eliminate false positive, that is lung vessels, from nodule candidate, firstly, rule-based method was applied. In the rule-based method, upper and lower threshold were determined for each features empirically, respectively. If all features of a candidate are in between each lower and upper threshold, then the candidate is classified as a lung nodule. Otherwise, the candidate is classified as a false positive and eliminated from nodule candidates.

Moreover, false positives were removed from nodule candidates by use of three-layer ANNs with a back-propagation algorithm. Eight features including four intensity features and four shape features were used as input data for the ANN. The structure of the ANN consisted of eight input units, 4 hidden units, and one output unit. The number of hidden units was determined empirically. The output shows the likelihood of nodules. If the value of the output is lower than 0.5, the candidate is classified as a false positive and eliminated from the nodule candidates. Training and testing of the ANN were performed by use of the same database on a consistency test. On the other hand, the training and testing of the ANN was performed by use of the

leave one-case-out method on a validation test. In this method, all of the cases, excluding one case in the database, were used for training, and the one case not used was applied in the testing of the trained ANN.

3 Experimental Results

We used MDCT images obtained with multi-slice CT scanners (Light Speed QXi, GE, Milwaukee, USA), with four row detectors to develop a new method for the detection of lung nodules by using temporal subtraction. The matrix size for each slice image was 512 x 512 pixels, and the voxel size was 0.67 mm on average on the x and y axes, and 5.00 mm on the z axis. We applied our computerized scheme of detection of lung nodules to 6 thoracic MDCT cases. In all of the cases used in this study, lung nodules newly appeared in the current CT image sets. The total number of lung nodules was 94, and the diameters ranged from 3.3 mm to 57.6 mm. The interval of CT scanning was 3 months on average. In our experiment, consistency and validation tests were performed on refinement of nodule candidates by use of ANNs.

Table 1. Detection result for nodule candidate from temporal-subtraction image

	TPF [%]	FPs/scan
Large nodules over 20 mm	96.9 (31 / 32)	1.2
Small nodules under 20 mm	87.3 (76 / 87)	50.0

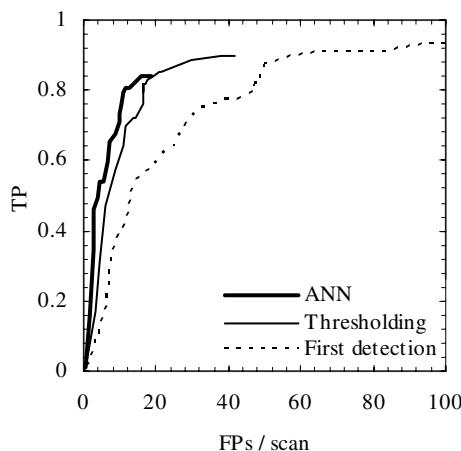


Fig. 5. Relationship between FP and TP on nodule detection

First, the results of nodule detection from the temporal-subtraction image are shown in Table 1. As shown in Table 1, a relatively good performance in the detection of nodules with size over 20 mm was achieved, where the sensitivity was 96.9 % and the false-positives were 1.2 per scan. On the other hand, for nodules with

Table 2. Detection result for lung nodules

	TPF [%]	FPs/scan
Detection of nodule candidates	79.3 (69 / 87)	47.2
Refinement by use of threshold technique	79.3 (69 / 87)	16.5
Refinement by use of ANN	79.3 (69 / 87)	11.0

Table 3. Detection result for nodule candidate on consistency and validation tests

	TPF [%]	FPs/scan
Consistency test	71.2 (62 / 87)	9.8
Validation test	71.2 (62 / 87)	11.5

size under 20 mm, many false-positives were detected; the sensitivity was 87.3 % and the false positive were 50.0 per scan.

Second, the results of refinement of nodule candidates are shown in Fig. 5. In Fig. 5, the distribution of true positives and false positives of the ANN was obtained by consistency test. As shown in Fig. 5, the performance of nodule detection was improved by use of the threshold technique with ANN. Table 2 shows a comparison of the true positives and false positives between the first detection of nodule candidates and refinement of nodule candidates by use of the threshold technique and ANN in Fig. 5. As shown in Table 2, false positives were reduced to 11.0 per scan by use of the threshold technique with ANN with a sensitivity of 79.3 %.

Finally, the results of a comparison between the consistency and validation tests are shown in Table 3. As shown in this table, our computerized scheme provided a sensitivity of 71.2 %, with false positives of 9.8 and 11.5 per scan on the consistency test and validation test.

4 Discussion and Conclusions

In the detection of nodule candidates from a temporal-subtraction image, it was found that the performance of detection for lung nodules with sizes under 20 mm was worse than size over 20 mm, as shown in Table 1. Many false positives (50.0 per scan) were detected, this means that many subtraction artifacts caused by mis-registration of small lung vessels remained in the temporal-subtraction image. On the other hand, some lung nodules could not be detected in the temporal-subtraction image. Some examples of false negatives of lung nodules are shown in Fig. 6. As shown in Fig. 6, nodules which were small and came in contact with the pleura and lung vessels could not be detected in this study. Upon setting the parameter in the multiple-threshold technique to detect these small nodules, a lot of false positives were detected accidentally. On the other hand, small nodules attached to the pleura were removed from the temporal-subtraction image because of mis-registration of the lung surface and mis-segmentation of the lung region. Moreover, in the detection of nodules attaching to lung vessels, segmentation of nodules could not be performed, and accurate shape features could not be calculated. For overcoming these problems, improvement of the registration of small lung vessels and the lung surface is needed.

As shown in Table 3, we could reduce false positives to 11.5 per scan with a sensitivity of 71.2 % by use of the rule-based method with ANN. However, some lung vessels, running along the axial direction, were detected as nodules, as shown in Fig. 7. These vessels were like a circle in the 2-D image, and some of their shape features showed similar values to nodules. To eliminate these false positives, features related to 3-D shape information of a nodule candidate of a 3-D image are needed in feature analysis.

In conclusion, we have developed a computerized scheme of detection for lung nodules from MDCT by use of temporal subtraction. We believe that the temporal-subtraction images would be useful to radiologists for detection of interval changes on MDCT images.

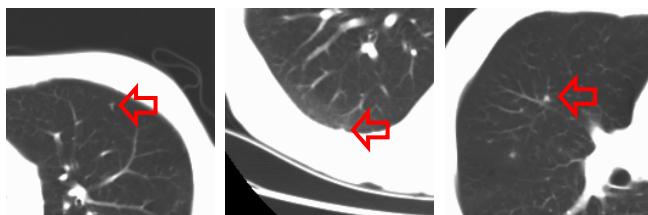


Fig. 6. Examples of false negatives

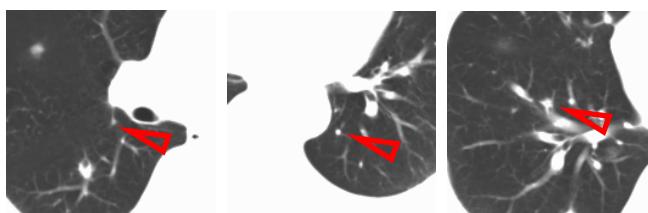


Fig. 7. Examples of false positives

Acknowledgement

This work was partly supported by Tateisi Science and Technology, and a Grant-In-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology of Japan under grant number 20560397, which is hereby gratefully acknowledged. We are grateful to Mrs Elisabeth Lanzl for improving this manuscript. This work is also in part supported by USPHS grant CA 98119.

References

1. Kano, A., Doi, K., MacMahon, H., Hassell, D.D., Giger, M.L.: Digital image subtraction of temporally sequential chest images for detection of interval change. *Med. Phys.* 21(3), 445–461 (1994)

2. Kakeda, S., Nakamura, K., Kamada, K., Watanebe, H., Nakata, H., Katsuragawa, S., Doi, K.: Improved detection of lung nodules by using a temporal subtraction technique. *Radiology* 224(1), 145–151 (2002)
3. Itai, Y., Kim, H., Ishikawa, S., Katsuragawa, S., Doi, K.: A new registration method with voxel-matching technique for temporal subtraction images. In: Proc. SPIE, vol. 6915, p. 691531 (2008)
4. Awai, K., Murao, K., Ozawa, A., Komi, M., Hayakawa, H., Hori, S., Nishimura, Y.: Pulmonary Nodules at Chest CT: Effect of computer-aided diagnosis on radiologists' detection performance. *Radiology* 230(2), 347–352 (2004)
5. Sugiyama, A., Kamano, S., Yamamoto, S., Matsumoto, M., Tateno, Y., Iinuma, T., Matsumoto, T.: Reduction of false-positive shadows in computer-aided diagnosis system for chest X-ray CT images. *Med. Imag. Tech.* 17(3), 217–227 (1999) (Japanese)
6. Li, Q., Sone, S., Doi, K.: Selective enhancement filters for nodules, vessels, and airway walls in two- and three-dimensional CT scans. *Med. Phys.* 30(8), 2040–2051 (2003)

Improved Mass Spectrometry Peak Intensity Prediction by Adaptive Feature Weighting

Alexandra Scherbart¹, Wiebke Timm^{1,2}, Sebastian Böcker³,
and Tim W. Nattkemper¹

¹ Biodata Mining & Applied Neuroinformatics Group,
Faculty of Technology, Bielefeld University

{ascherba,wtimm,tnattkem}@techfak.uni-bielefeld.de

² Intl. NRW Grad. School of Bioinformatics & Genome Research, Bielefeld University

³ Bioinformatics Group, Jena University
boecker@minet.uni-jena.de

Abstract. Mass spectrometry (MS) is a key technique for the analysis and identification of proteins. A prediction of spectrum peak intensities from pre computed molecular features would pave the way to a better understanding of spectrometry data and improved spectrum evaluation. The goal is to model the relationship between peptides and peptide peak heights in MALDI-TOF mass spectra, only using the peptide's sequence information and the chemical properties. To cope with this high dimensional data, we propose a regression based combination of feature weightings and a linear predictor to focus on relevant features. This offers simpler models, scalability, and better generalization. We show that the overall performance utilizing the estimation of feature relevance and re-training compared to using the entire feature space can be improved.

1 Introduction

Mass spectrometry (MS) is a key technique for the analysis and identification of proteins and peptides. Matrix-assisted laser desorption ionization (MALDI) is one of the most often used technique for the analysis of whole cell proteomes in high-throughput experiments. In quantitative proteomics, approaches are desired to allow a comparison of protein abundances in cells. Proteins are quantitatively characterized in a complex sample or protein abundances across different samples are compared. Opposed to labeling methods, label-free methods directly use peak heights (referred to as intensities) to estimate peptide abundances. There are different applications of MALDI-MS where the prediction of peak intensities in spectra are needed for further improvements. The identification is commonly done by comparing the peak's masses from a spectrum (protein mass finger print, PMF), to theoretical PMFs in a data base, generating a score for each comparison. Different tools are available for this purpose. For an overview see [1]. These tools rarely use peak intensities, because there is no model to calculate the theoretical PMFs directly. The use of peak intensities could improve the reliability of protein identification without lowering the error rate, as was shown by [2] for tandem MS.

For the prediction of MALDI PMF there has been one study so far by [3] who applied different regression and classification algorithms. [4] used multi-layer neural networks to predict peptide detectabilities (i.e. the frequency with which peaks occur in spectra) in LC/MS ion trap spectra which is a related problem.

An algorithmic approach for peak intensity prediction is a non-trivial task because of several obstacles: The extraction of PMF from spectra is a signal processing task which can not be done perfectly. Data from this domain is always very noisy and contains errors introduced by preprocessing steps in the wet lab as well as in signal processing. Misidentifications may even lead to wrong sequences. Intensity values can be distorted due to the unknown scale of spectra. It is nearly impossible to come by a large enough dataset from real proteins where the content is known, i.e. there is no perfect gold standard, because of the not reproducible and non-unique peptide/intensity relation.

To model the relationship of peptide sequences and the peak intensities, as regression task, numerical feature vectors have to be calculated building the feature space as input for the learning architectures. Intensities for different peptide sequences differ even if they have the same abundance, because of their different chemical properties. We propose to use chemical properties of the peptides derived from the peptide sequences. Since these high-dimensional vectors are supposed to contain redundant and interdependent information, a method is needed, that can cope with this data and is able to reduce dimensionality of feature space. The task of finding a suitable subset of features is well-known as feature selection and is one of the central problems in machine learning [5]. Focusing on relevant features that contribute to model the peptide/peak intensity relation offers simpler models, scalability, and better generalization [6]. Additionally, it reduces computational costs.

To overcome these obstacles to predict peak intensities in MALDI-TOF spectra based on a training set of peptide/peak intensity pairs, we considered an artificial neural net architecture, namely the Local Linear Map [7], since it combines unsupervised (a) and supervised (b) learning principles, with comparable results to those obtained by ν -Support Vector Regression (SVR) [8]. The LLM can learn global non-linear regression functions by fitting a set of *local* linear functions to the training data. It is very efficient (i.e. fast training and adaptation to addition data, and low memory-usage) and offers transparency. Other than for example SVR it can be used for data mining once adapted in a straight forward manner. Both architectures have been proposed to model the non-linear relationship between peptide and peak intensities. In previous experiments, we have modelled the relationship between with a positive correlation of $r^2 = 0.46$. But the reduction to a set of 18 features chosen in an ad-hoc forward feature selection brought no significant performance gain.

In this paper, the regression approach is extended by combining feature weightings with a linear predictor. We propose a two-step regression approach, which includes adaptive regression based feature weighting by m learning architectures individually, to ensure different kind of regression behavior and abstraction levels. Subsequently, the estimated feature weightings of d features are

used for scaling the input vectors in a retraining step with an independent predictor. The performance when utilizing a weighting of features and subsequent re-training is compared to an approach integrating weighting and filtering.

2 Materials and Methods

2.1 Data

In this study we use two datasets **A** and **B** of peptides of MALDI mass spectra. The first one, **A**, consists of 66 spectra of 29 different proteins, with 16 of these proteins being present in multiple spectra, whereas **B** consists of 200 spectra of 137 different proteins with 39 of these proteins occurring multiple times.

Peak extraction steps include soft filtering, baseline correction, peak picking and isotopic deconvolution in the corresponding raw spectra. The resulting list of peaks is matched against masses derived from a theoretical tryptic digestion. These steps for **A** (and **B** respectively) result in 857 (1631) matched peaks corresponding to 415 (1135) different peptides. In the remainder, the intensities refer to scaling the original intensity by “mean corrected ion current”, where I_p^{orig} is the original intensity after peak extraction, I_i is the raw value at index i , B_i the baseline, and N_i the noise determined in the denoising step:

$$I_p^M = \ln \left(\frac{I_p^{\text{orig}}}{\sum_{i=1}^N I_i - B_i - N_i} + 1 \right).$$

Feature Sets. We derive numerical representations of the peptides from the peptide sequences to contribute different properties of peptides. The feature vectors are built by amino acid frequencies (20 monomers, *mono*), typically used in bioinformatics, and by physico-chemical information about the amino acids constituting the peptide. The latter feature vectors are attributes taken from the amino acid index database [9] (*aaindex*) extended by peptide length, mass, and numbers and fractions of acidic, basic, polar, aliphatic and arginine residues, yielding 531-dimensional vectors.

Most of the peptides in the dataset occur multiple times in different spectra with different intensity values. Due to limitation of training data, we eliminate outliers (potential noisy peptides) by mapping each peptide to one unique value, the α -trimmed mean of all intensities per distinct peptide with $\alpha = 50\%$. The α -trimmed mean is defined as the mean of the center 50% of an ordered list. In the case of less than 4 peptides in the list a simple mean is taken.

2.2 Feature Selection/Weighting

Classic feature selection steps include heuristic search (forward selection, backward elimination, stepwise selection) and successively adding or eliminating attributes by an adequate strategy. The approaches can be divided into filters and wrappers. Filter approaches use general characteristics (e.g. correlation) of the data provided to select a subset of features, independently of the chosen learner. Wrappers score subsets of features by a metric according to the estimated accuracy of a given learning machine.

To reflect feature importance as a numerical value, an estimation of the feature relevance is done by different regression models which are trained on entire feature space. Each of the applied architectures comprises an internal model dependent metric as measure of quality such that the feature weights are set proportionally to assessed change in accuracy (via correlation, MSE) e.g. to decrease in error when permuting features. The greater the decrease in performance when leaving a certain feature out, the higher is the assigned degree of relevance of the feature. The model dependent metrics evaluation as offered by the R [10] package [11] and yield m estimations interpreted as feature weightings $Z \in \mathbb{R}^{m \times d}$ and are calculated according to:

- Linear Models (**LM**): absolute value of t-statistic for each model parameter.
- Random Forest (**RF**): [12]: “the average increase in squared OOB (out-of-bag) residuals when the variable is permuted”.
- Partial Least Squares (**PLS**): weights are proportionally to decrease in the sums of squares.
- Bagged Trees (**BT**): total importance over all bootstrapped trees.
- Boosted Trees (**GBM**): total sum of importance over all boosting iterations.

2.3 Local Linear Map

Motivated by the Self-Organizing Maps (SOM) [13], an LLM consists of a set of n_l regular ordered nodes $\mathbf{v}_i, i = 1, \dots, n_l$, which are connected to each other via a two-dimensional grid structure, defining a neighborhood between the nodes and a topology in feature space. Each node consists of a triple $\mathbf{v}_i = (\mathbf{w}_i^{\text{in}}, \mathbf{w}_i^{\text{out}}, \mathbf{A}_i)$. The vectors $\mathbf{w}_i^{\text{in}} \in \mathbb{R}^{d_{\text{in}}}$ are used to build prototype vectors adapting to the statistical properties of the input data $\mathbf{x}_\xi \in \mathbb{R}^{d_{\text{in}}}$. The vectors $\mathbf{w}_i^{\text{out}} \in \mathbb{R}^{d_{\text{out}}}$ approximate the distribution of the target values $\mathbf{y}_\xi \in \mathbb{R}^{d_{\text{out}}}$. The matrices $\mathbf{A}_i \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ are locally trained linear maps from the input to the output space. In the unsupervised training phase, the prototype vectors \mathbf{w}_i^{in} are adapted following the SOM learning rule: the vectors \mathbf{w}_i^{in} are pulled towards the input pattern \mathbf{x}_ξ according to the distance between the input pattern and the corresponding closest prototype in input space $\mathbf{w}_\kappa^{\text{in}}$, with $\kappa = \operatorname{argmin}_i \{\|\mathbf{x}_\xi - \mathbf{w}_i^{\text{in}}\|\}$. After unsupervised adaptation and tessellation of the input space, an input feature vector is mapped to an output by the corresponding local expert: $\mathcal{C}(\mathbf{x}) = \mathbf{w}_\kappa^{\text{out}} + \mathbf{A}_\kappa (\mathbf{x}_\xi - \mathbf{w}_\kappa^{\text{in}})$. The weights $\mathbf{w}_i^{\text{out}}$ and the linear map \mathbf{A}_i are changed iteratively by the gradient descent learning rules.

The concept of approximating nonlinear functions by fitting simple models to localized subsets of the data is related to other regression approaches like Locally-Weighted Regression [14] and to radial basis functions [15].

2.4 Evaluation

Performance assessment is done by 10-fold cross-validation (CV) for each regression model. It was ensured that peptides from one spectrum as well as peptides occurring in more than one spectrum are found in only one of the portions.

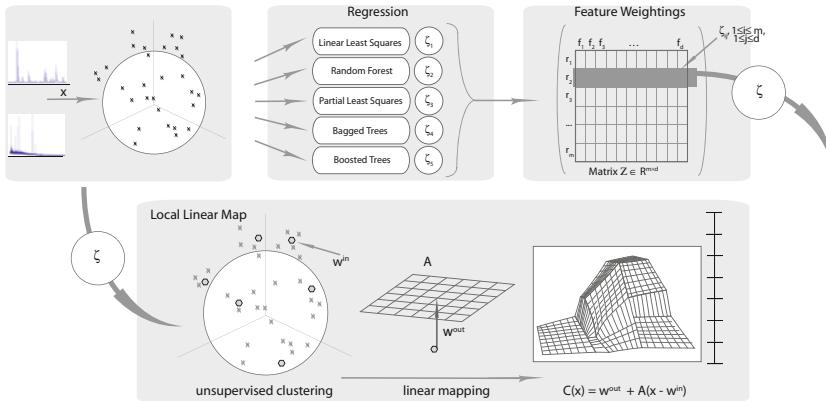


Fig. 1. Graph depicting the proposed architecture

Model selection: Grid search over the parameter space is performed to determine optimal parameters for learning. For every point in the parameter space the prediction accuracy for every training/test set is determined by squared pearson-correlation coefficient r^2 and root mean square error of the test set. The choice of the best parameter set is made by the best mean r^2 over all 10 test sets.

Model assessment: The final model is built by retraining the whole dataset with the optimal parameters chosen in the previous step. To validate its generalization error on new data, the other dataset respectively is used.

Feature Weighting: The final models **{LM, RF, PLS, BT, GBM}** are evaluated by the corresponding model specific metric for estimating the contribution of each feature.

Filtering of Feature Weightings: A filtering approach discards feature weightings below a certain threshold. The cut-off value is determined by the $t\%$ most important features (AUC), where $t \in \{70, 90, 95\}$ and hence these features are chosen $\max_i \arg T_i \leq t$ for a fixed method k and ω_k . sorted in descending numerical order, where $T_j = \sum_{i=1}^j \omega_{ki}, 1 \leq j \leq d$.

Retraining of LLM model: These feature weightings Z (scaled up to be in range of $[0, 1]$) are then used to scale the input variables to $\tilde{\mathbf{x}} = \mathbf{w} \cdot \mathbf{x}$. A new model of LLM-type is built based on the optimal parameters determined in the model selection step with the input vectors $\tilde{\mathbf{x}}$.

3 Results

In this study, we compare the performance of a predictor when applying different approaches of weighting and filtering of features. The feature importance or degree of feature relevance is estimated by 5 regression architectures. We evaluate the peak intensity prediction on two different datasets (**A**, **B**) for two

different feature sets (**aaindex** (531-dimensional vectors) and **mono** (20-dim.)). To make the results comparable, exactly one predictor is used as reference learning architecture, namely the LLM, for retraining on the derived feature space. Performance and generalization performance is assessed via across dataset prediction as described in 2.4. The presented results are restricted to these of dataset **A**, trained, parameter-tuned (CV) and validated on **B** (GV).

3.1 Non-weighted Feature Space

The regression models of the learning architectures are evaluated in terms of prediction and generalization performance (r^2) as well as their estimation of the contribution of features to the corresponding final model taking the entire, non-weighted feature space as input. The reference results are summarized in Tab. II for the **aaindex** and the **mono** feature space and the regression architectures **LM**, **LLM** and **SVR** given in the last three columns.

It can be observed that the applied learning architectures differ strongly in their performance. This trade-off may be due to two main reasons: First of all, the number of variables (531) even exceeds the number of available peptides in the dataset (415). Best prediction and generalization performance is observed for the **SVR**, while the **LLM** shows only slight worse accuracy regarding *mono* feature space. The linear model (**LM**) shows a clear overfitting and lacks of generalization on new peptides across datasets. A comparison between the two peptide features representations shows better performance for the small **mono** feature space in general. These observations suggest a non-linear relationship between peptide feature vectors and the intensity target values.

3.2 Weighted Feature Space

We utilize the estimated contributions of features from the model specific metrics and perform a retraining of the LLM models by replacing the input variables: $\tilde{\mathbf{x}} = \mathbf{w} \cdot \mathbf{x}$. The model of LLM-type is rebuilt based on the optimal parameters determined in the model selection step with the input vectors $\tilde{\mathbf{x}}$.

A comparison in Tab. II of the resulting performances when introducing a feature weighting shows clearly the increase in accuracy for all derived estimations of variable importance applied to the **LLM** and *aaindex* feature space.

While an improvement of prediction performance can be observed for the **LLM** compared to standard **LLM** trained on the entire *aaindex* feature space in all cases, the same holds for generalization on new data, despite of **LLM_{LM}** and **LLM_{RF}** models. Though using a linear predictor as a feature weighting method, the **LLM_{LM}** keeps up to extract and separate the underlying characteristics of the peptides in CV. The feature weighted-LLM trained on the high-dimensional *aaindex* feature space beats the non-weighted models of **LLM**-type trained on *aaindex* feature space as well as for the *mono* feature space in general. It outperforms the SVR in terms of prediction performance and it is only slightly worse in generalization case. The **LLM_{RF}** and **LLM_{PLS}** models trained on *aaindex* feature space yield the major improvement of prediction performance.

Table 1. Prediction accuracy of regression models of **LLM**-type - retrained on **aaindex** feature space (531 dim) incorporating input vectors scaled by feature weightings. The three last columns are given as reference accuracy in feature spaces without scaling. Results are also given for the *mono* feature space (20 dim).

<i>aaindex</i>	LLM_{LM}	LLM_{GBM}	LLM_{RF}	LLM_{PLS}	LLM_{TB}	LLM	SVR	LM
CV	0.33	0.48	0.46	0.47	0.49	0.27	0.44	0.36
GV	0.18	0.26	0.01	0.39	0.35	0.27	0.42	0.02
<i>mono</i>	LLM_{LM}	LLM_{GBM}	LLM_{RF}	LLM_{PLS}	LLM_{TB}	LLM	SVR	LM
CV	0.42	0.46	0.44	0.42	0.45	0.41	0.46	0.27
GV	0.29	0.30	0.31	0.30	0.30	0.33	0.44	0.20

3.3 Weighted and Filtered Feature Space

There is a small number of features considered as highly relevant, while a few features were estimated to contribute to the predictor model to very low degree. Hence, they might be overweighted or this is due to the number of other features slightly correlated to each other. It seems reasonable to apply a filtering on the resulting estimated feature weightings prior to retraining the **LLM**. A filtering approach following discards feature weightings below a certain threshold. The cut-off value is determined by the $t\%$ most important features (AUC).

With a filtering of the feature weightings the performance is not increased for all applied feature weighting methods. With a rising filtering threshold, prediction performance keeps constant, while the generalization performance decreases. The corresponding results are given exemplarily regarding the **LLM_{TB}** for the different values of in terms of CV {0.49, 0.47, 0.46} and in terms of GC {0.31, 0.3, 0.3}. The corresponding number of features excluded were {225, 274, 384}. The precedent filtering of feature weights leads to a slight worse prediction performance of the subsequent applied **LLM**.

4 Conclusions

The learning architectures of Local Linear Map-type [16] and ν -Support Vector Regression (SVR) [8] have been proposed to model the non-linear relationship between peptide and peptide peak heights in MALDI-TOF mass spectra. High-dimensional numerical feature vectors are derived from the peptide sequence building the feature space as input for the learning architectures. These features are supposed to differ in relevance. For this purpose, we focus on the issue of using relevant features in modelling the non-linear relationship between peptides and peptide peak heights. The regression architecture of **LLM**-type is extended by assigning features degrees of relevance according to their estimated contributions to different predictor models. We propose a regression based combination of estimated feature weightings and a linear predictor offering simpler models, better generalization and reduced computational costs. A comparison between the two peptide feature representations shows better performance for the high-dimensional *aaindex* feature space in general. We got the major improvement in

performance of regression models when retraining with weighted features based on estimated feature relevance by Partial Least Squares and Bagged Trees. These model dependent feature weightings methods perform a skillful scoring of the features in combination with the **LLM**. Though many features were supposed by the applied methods to be relevant to low degree, integrating a filtering of the feature weightings prior to retraining the LLM led to decrease of performance. The most relevant *aaindex* features found amongst others were estimated gas-phase-basicity, fractions of arginine residues, acidic, basic and polar.

References

- Shadforth, I., Crowther, D., Bessant, C.: Protein and peptide identification algorithms using MS for use in high-throughput, automated pipelines. *Proteomics* 5(16), 4082–4095 (2005)
- Elias, J.E., Gibbons, F.D., King, O.D., Roth, F.P., Gygi, S.P.: Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nat. Biotechnol.* 22(2), 214–219 (2004)
- Gay, S., Binz, P.A., Hochstrasser, D.F., Appel, R.D.: Peptide mass fingerprinting peak intensity prediction: extracting knowledge from spectra. *Proteomics* 2(10), 1374–1391 (2002)
- Tang, H., et al.: A computational approach toward label-free protein quantification using predicted peptide detectability. *Bioinformatics* 22(14), 481 (2006)
- Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. *Artificial Intelligence* 97(1-2), 245–271 (1997)
- Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3, 1157–1182 (2003)
- Ritter, H.: Learning with the self-organizing map. In: Kohonen, T., et al. (eds.) *Artificial Neural Networks*, pp. 379–384. Elsevier Science Publishers, Amsterdam (1991)
- Timm, W., Böcker, S., Twellmann, T., Nattkemper, T.W.: Peak intensity prediction for pmf mass spectra using support vector regression. In: Proc. of the 7th International FLINS Conference on Applied Artificial Intelligence (2006)
- Kawashima, S., Ogata, H., Kanehisa, M.: AAindex: Amino Acid Index Database. *Nucleic Acids Res.* 27(1), 368–369 (1999)
- R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Stat. Comp., Austria (2008) ISBN 3-900051-07-0
- Kuhn, M.: caret: Classification and Regression Training, R package v. 3.16 (2008)
- Liaw, A., Wiener, M.: Classification and regression by randomforest. *R News* 2(3), 18–22 (2002)
- Kohonen, T.: Self-organized formation of topologically correct feature maps. In: *Biological Cybernetics*, vol. 43, pp. 59–69 (1982)
- Cleveland, W.S., Devlin, S.J.: Locally-weighted regression: An approach to regression analysis by local fitting. *J. of the American Stat. Assoc.* 83, 596–610 (1988)
- Millington, P.J., Baker, W.L.: Associative reinforcement learning for optimal control. In: Proc. Conf. on AIAA Guid. Nav. and Cont., vol. 2, pp. 1120–1128 (1990)
- Scherbart, A., Timm, W., Böcker, S., Nattkemper, T.W.: Som-based peptide prototyping for mass spectrometry peak intensity prediction. In: WSOM 2007 (2007)

An Improved Genetic Algorithm for DNA Motif Discovery with Public Domain Information

Xi Li^{1,2} and Dianhui Wang¹

¹ Department of Computer Science and Computer Engineering,
La Trobe University,
Melbourne, VIC, 3086, Australia
dh.wang@latrobe.edu.au

² Department of Primary Industries,
Bioscience Research Division, Victorian AgriBiosciences Centre,
1 Park Drive, La Trobe Research and Development Park,
Bundoora, VIC, 3083, Australia

Abstract. Recognition of transcription factor binding sites (TFBSs or DNA motifs) to help with understanding the regulation of gene expression is one of the major challenges in the post-genomics era. Computational approaches have been developed to perform binding sites discovery based on brute-force search techniques or heuristic search algorithms, and numbers of them have achieved some degrees of success. However, the prediction accuracy of the algorithm can be relatively influenced by the natural low signal-to-noise ratio of the DNA sequence. In this paper, a novel DNA motif discovery approach using a genetic algorithm is proposed to explore the ways to improve the algorithm performance. We take account of the publicly available motif models such as Position Frequency Matrix (PFM) to initialize the population. By considering both conservation and complexity of the DNA motifs, a novel fitness function is developed to better evaluate the motif models during the evolution process. A final model refinement process is also introduced for optimizing the motif models. The experimental results demonstrate a comparable (superior) performance of our approach to recently proposed two genetic algorithm motif discovery approaches.

1 Background

The short segments (usually ≤ 30) of DNA sequences where Transcription Factors (TFs) bind to are named as Transcription Factor Binding Sites (TFBSs). One believes that the interaction between TF and TFBS dominants the regulation of gene expression. Most of the time TFBSs are located within the promoter region (100–1000 bp) of a gene's Transcription Start Site (TSS), while they can also be found in gene's downstream and coding regions. A DNA Motif refers to a common conserved appearance shared by a group of TFBSs recognized by the same TF. The DNA motif discovery problem can be regarded as extracting some short (5–30 letters) unknown motifs from a set of genes of interest.

The traditional binding sites identification methods, such as DNase footprinting [4], can provide trustable identification results, whereas they are labor-intensive and time-consuming. Nowadays, as massive genomic sequence and gene expression profiles become available, numerous computational tools have been developed to offer an alternative way to explore the DNA motif find problem. Many literature studies classify the search algorithms into exhaustive approaches and heuristic approaches. The exhaustive approaches guarantee to find the most over-represented words by enumerating the search space of counting the frequencies of all possible occurrences, such as CONSENSUS [13]. In the other hand the heuristic approaches are often designed by using the probabilistic models, which initially estimate the motif model and iteratively optimize the parameters of the estimated model to maximize the likelihood, such as MEME [1]. Tompa et.al. [12] evaluated 13 motif discovery tools using the eukaryotic datasets and as a complement Hu et.al. [5] reported the performance assessment of 5 representative tools on the prokaryotic data. The results from both studies show that current motif discovery tools are still lack of accuracy and far from perfect.

Genetic Algorithm (GA) has been introduced to the motif discovery problem ([2], [3], [7], [14]). Recently proposed GA algorithms have shown the evidences of improvement. For example, Wei and Jensen [14] presented a GA-based cis-element discovery framework named as GAME, which employs a Bayesian-based posterior distribution as the fitness function to evolve the population. In 2008, Chan et.al. [2] proposed another GA approach called GALF-P that employed a local filtering operator and an adaptive post-processing to reduce the false-positive rate. Comparative studies carried out by Chan et.al. [2] on both synthetic datasets and experimental datasets show that GALF-P achieves a better overall performance than GAME as well as other 3 commonly used motif finding tools.

In this work, we present an improved approach to identification of the DNA motif by using a genetic algorithm (IGAMD). Our method attempts to incorporate the experimental valid Position Frequency Matrix (PFM) for initializing the population. We employ information content and complexity measure as fitness function to effectively evaluate the candidate solutions. A new genetic operator named as substitution is introduced to avoid the local maximum solution. We also develop a final model refinement process to improve the model quality. Comparative studies of the promoter regions of 302 co-regulated genes containing 580 experimental verified TFBSS from 10 *E.coli* TF families demonstrate the advantageous performance of our approach over two GA-based methods GAME and GALF-P.

2 The Proposed Approach

In our method, a prior knowledge represented as PFM is introduced to support for constructing the initial population. During the evolution process, the fitter individuals have better chance to reproduce or to be kept in the next generation. We also explore to use a novel fitness function and a combination of genetic

operators for improving the algorithm performance. A final motif model refinement process is given to optimize the predicted results.

2.1 Representation

An individual represents a potential solution to optimize the problem. In DNA motif finding domain, an individual can be regarded as a motif model extracted from a set of overrepresented short DNA sequences (denoted as k -mers where k is the length of the motif), thereby the target of our GA approach is to find the optimal individual. In [2] and [14], the individual is represented by a string of integers storing the start position of a k -mer in each target sequence. Comparing with the position-based representation, the individual in our study is represented as a vector $\{v_1, v_2, \dots, v_n\}$, where v_m is a k -mer from the m -th sequence and n is the number of the input sequences.

2.2 Fitness Score Function

It is crucial to define a suitable fitness function that measures the quality of our proposed motif model representation. In previous GA approaches, some of them applied the information theory as fitness function, such as [7] and [14], some chose the distance-oriented functions to score individuals [3], or a combination of both [2]. Recently Congdon et.al. [3] presented a benchmark to access the GA performance by applying 1) Information Content (IC) and 2) distance-oriented separately as fitness function as well as 3) some variations of the combination of the two methods. The results show IC fails from the competition against other two attempts [3]. We believe a combination of metrics from different aspects together to evaluate the individual sounds a reasonable solution.

The information content (IC) computed using the relative entropy of the binding sites with respect to the background base distribution [11] is applied here to evaluate the conservation significance of a motif model. It can be regarded as an indication of how much deviation of a motif model from the background distribution. The IC value of a DNA motif model M is computed by:

$$IC(M) = \sum_{j=1}^k \sum_{b=A}^T f_b(j) \log \frac{f_b(j)}{p_b} \quad (1)$$

where $f_b(j)$ is the normalized frequency of nucleotide $b \in \Sigma = \{A, T, C, G\}$ (for DNA sequences) on the j -th position of all instances in M and p_b is a background base frequency estimated from the intergenic regions of the investigated genome.

A higher IC value can imply the higher conservation of a motif model, while a group of repetitive segments such as sequences of A, which can not be considered as possible binding sites, will produce a significant high IC value. If we only expect the individual with the highest IC as the optimal motif model, the vulnerability is obvious. Thus, a model complexity score function proposed by Mahony et.al. [9] can effectively measure the model compositional structure to exclude the models with low complexity scores. In our study, an individual

with a complexity score less than a threshold value will be removed from the population. The complexity measure is given by,

$$c(M) = \left(\frac{1}{4}\right)^k \prod_{b=A}^T \left(\frac{k}{\sum_{j=1}^k f(j, b)} \right)^{\sum_{j=1}^k f(j, b)} \quad (2)$$

To describe both the conservation property and compositional structure of a given motif model during the evolution process, we propose a new fitness function that is,

$$ICC(M) = \sum_{j=1}^k \sum_{b=A}^T f_b(j) \log \frac{k f_b(j)}{p_b \sum_{j=1}^k f(j, b)} + k \log \left(\frac{1}{4}\right) \quad (3)$$

2.3 Population Initialization

To initial the population, a verified PFM motif model K with a given length k is given. We adopt the formula used in roulette-wheel selection to generate a Position Cumulate Frequency Matrix (PCM) based on K , that is $p_{ij} = \frac{f_{ij}}{\sum_{z=A}^T f_{iz}}$.

Here $z, j \in \{A, C, G, T\}$ are for DNA sequence, $i \in \{1, 2, \dots, k\}$ is the i -th column, f_{ij} is the frequency of the nucleotide j at i -th column. Also the roulette-wheel selection is applied to produce a list of consensus sequences denoted as $\{C_1, C_2, \dots, C_s\}$ from the PCM, where s is the population size. The consensus can be regarded as a possible motif instance from K . Each of them is then used to scan the input sequences to collect a group of k -mers with the highest similarities as an individual. If more than one k -mers found from one sequence, the first appearance will be parsed and stored to the individual.

2.4 Evolution Process

The conventional roulette-wheel selection is used to choose individuals as parents for reproduction. Thus, the individual with higher fitness degree has a higher opportunity to be selected to reproduce than the others. For reproduction, a crossover operator, a mutation operator, and a substitution operator are proposed here.

When the crossover operator is chosen with a certain crossover probability, a single crossover point is randomly generated on both parents so that all k -mers beyond this point are exchanged between two parents. The one-point crossover is employ in both [14] and [2] as well. The promising experimental studies from them make us believe that the one-point crossover sounds to be a reasonable choice. When the mutation is performed with a given mutation probability, one randomly chosen k -mer _{i} from the selected parent will be replaced by a random k -mer from the sequence i . Here, we introduce a new operator named substitution, which can be regarded as the replacement of an individual I_{old} by I_{new} that derives from I_{old} itself. The substitution operator triggered with a given

probability starts with a randomly chosen individual I_{old} from the population to construct a consensus string C from I_{old} 's PFM. The way to construct C is to choose the nucleotides with the second-highest frequency from r columns and the nucleotides with the highest frequency from the rest of columns with equal probability. Instead of randomly generating the number r , we adopt a distribution formula proposed by [7], $P(r = k) = 1/2^k$, where $k \in [2, n(\text{number of input sequence})]$ and $P(r = 1) = 1 - \sum_{k=2}^n 1/2^k$. Unlike the mutation, a new set of k -mers one from each sequence, which have the highest similarities against C , will be grouped to form a new individual to replace I_{old} . We believe by using three operators together during the evolution process, it can provide a stable diversity level of the population to potentially avoid the local maxima solution. The reproduction will not be stopped until the size of the population doubled.

To keep the same population size across every generation, we apply two possible selections as the replacement strategy, which are winners-take-all selection and tournament selection [14]. Winners-take-all guarantees to keep the best candidate solutions in every generation and saves the computing time, while the population may get stabilized after a short number of generations, which has the potential risk to be trapped into the local optima solution. Thus, by default the tournament selection has a higher chosen probability than winners-take-all. Although some individuals with low fitness scores may be selected, the average fitness for the following generation increases smoothly to keep away from the premature convergence. The evolution process will be terminated when the number of generations reaches a fixed number or the optimal solution keeps the same for a given iteration times.

2.5 Final Model Refinement

After the evolution process meets the terminating criteria, a final model refinement is proposed to improve the overall algorithm performance.

For each model, a consensus pattern is composed of the dominant letter (highest frequency) from each column of its PFM. We merge all motif models that share the same consensus pattern and a small difference in information contents ($|IC_a - IC_b| \leq \gamma$) together as a new motif model. In such way, some mis-clustered k -mers may be grouped together. As the number of k -mers in one model potentially increases after merging, it will directly affect the model quality. To solve this problem, we carefully develop a false positive instance removing method.

For a given motif model M , the removing process starts with finding a k -mer k_{max} that can maximize the IC of M after removed from it. If the increase value of IC is larger than a predefined small constant value β , the k_{max} will be removed permanently and the IC will be re-calculated for next iteration. The iteration continues till no k -mer satisfies the removing criteria from M . The removing method applies to all the merged models. In such way, we believe that any k -mers which have large negative impact on the model quality will be removed, whereas some weak true-positive binding sites can be kept in the model.

A log-likelihood scoring function Maximum a Posteriori (MAP), which measures the model conservation by giving a statistical significance score proposed

by Liu et.al. [8], is employed here for ranking the merged motif model. The model with the highest MAP score is the final output motif model. The MAP score is given by,

$$MAP = \frac{\log(x_m)}{k} \left[\sum_{l=1}^k \sum_b f(l, b) \log(f(l, b)) - \frac{1}{x_m} \sum_{s_i} \log(p_0(s_i)) \right] \quad (4)$$

where x_m is the number of k -mers to build the PFM, k is the width of PFM and $p_0(s_i)$ is the probability of generating a k -mer s_i from the Markov background model.

By applying the model refinement stage, it extends the ability of identifying weak binding sites and potentially purifies the motif model by removing the false-positive instances to reach our goal of improving the overall algorithm performance.

3 Results

3.1 Experimental Datasets

The promoter regions of 302 co-regulated genes from 10 *E.coli* TF families are downloaded from RAST (<http://rsat.ulb.ac.be/rsat/>). The 302 promoter regions contain 580 experimental verified TFBSSs in total. The information of the 580 TFBSSs and 10 PFMs constructed by true binding sites are obtained from RegulonDB [6]. The datasets have covered different motif properties, such as width of motif from 8 to 23, total number of sites per set from 15 to 95 and total number of sequences per set from 7 to 121. In such case, we can demonstrate the stability of algorithm performance under different scenarios. The detailed description of the datasets is shown in Table 1.

Table 1. Details of the experimental datasets

TF Name	ArgR	CRP	Fis	Fur	IHF	LexA	Lrp	NarL	NtrC	PhoP
No. of seq.	8	121	42	21	46	12	11	19	7	15
No. of BS	18	185	103	54	65	17	33	73	17	15
Width of BS	19	23	16	20	14	21	13	8	18	18
Ave. No.of BS per seq.	2.25	1.52	2.5	2.6	1.4	1.4	3	3.8	2.4	1

To evaluate the proposed algorithm performance, we choose precision P and recall R from the field of information retrieval [10]. A single measurement called F -measure F is also applied here to indicate the performance accuracy, which is calculated as a weighted average of the precision and recall [10]. The best value of F-measure is 1 and the worst is 0.

3.2 Comparisons with GA Applications

In this section we evaluate our approach in details by carrying out the experimental comparisons with the two GA methods GAME [14] and GALF-P [2].

Table 2. The average results of IGAMD, GAME and GALF-P on the 10 datasets for 20 runs

TF	IGAMD			GAME			GALF-P		
	P	R	F	P	R	F	P	R	F
ArgR	0.91 ± 0.13	0.59 ± 0.14	0.71	0.53 ± 0.10	0.81 ± 0.17	0.64	0.50 ± 0.05	0.83 ± 0.07	0.63
CRP	0.45 ± 0.17	0.70 ± 0.09	0.55	0.75 ± 0.02	0.63 ± 0.04	0.68	0.67 ± 0.17	0.51 ± 0.13	0.58
Fis	0.40 ± 0.05	0.40 ± 0.10	0.40	0.28 ± 0.14	0.10 ± 0.05	0.15	0.38 ± 0.02	0.21 ± 0.03	0.27
Fur	0.83 ± 0.06	0.93 ± 0.02	0.87	0.85 ± 0.02	0.90 ± 0.02	0.87	0.90 ± 0.02	0.87 ± 0.03	0.88
IHF	0.28 ± 0.06	0.46 ± 0.03	0.35	0.08 ± 0.12	0.04 ± 0.07	0.05	0.22 ± 0.10	0.20 ± 0.06	0.21
LexA	0.93 ± 0.09	0.81 ± 0.04	0.87	0.82 ± 0.06	0.86 ± 0.03	0.84	0.95 ± 0.06	0.74 ± 0.05	0.83
Lrp	0.24 ± 0.06	0.24 ± 0.12	0.24	0.19 ± 0.11	0.16 ± 0.10	0.17	0.16 ± 0.06	0.16 ± 0.07	0.16
NarL	0.18 ± 0.05	0.14 ± 0.06	0.16	0.09 ± 0.07	0.02 ± 0.02	0.03	0.06 ± 0.05	0.03 ± 0.02	0.04
NtrC	0.94 ± 0.08	0.57 ± 0.10	0.71	0.58 ± 0.04	0.78 ± 0.07	0.67	0.60 ± 0.07	0.82 ± 0.08	0.69
PhoP	0.31 ± 0.03	0.71 ± 0.06	0.43	0.13 ± 0.14	0.17 ± 0.17	0.15	0.35 ± 0.08	0.56 ± 0.12	0.43
Ave	0.55	0.55	0.55	0.43	0.45	0.44	0.48	0.5	0.49

For each approach we adjusted the probabilities of genetic operators (from 0.1 to 0.9), the population size (from 500 to 2000), and the number of generation (from 500 to 2000), and kept all other parameters by default across 20-runs. The average of Precision, Recall and *F*-measure of 20-runs for each approach across 10 datasets as well as the standard deviation of Precision and Recall (followed by the ± symbol) are shown in Table 2. The best results are displayed as bold.

According to our results, IGAMD shows the comparable performance against the other two. In 7 of the 10 datasets, it has the best *F*-measure and the best precision score respectively. The number of best recall is 6 of 10. As a result, IGAMD outperforms the other two in terms of the average Precision, Recall and *F*-measure. We notice that for some datasets with high number of average binding sites per sequence such as Fis (2.45), Lrp (3) and NarL (3.8), all algorithms returned low prediction accuracy. From our point of view, it may be caused by the assumption made by all three algorithms which is one binding site per sequence during the evolution process. Although GAOMP shows the improvement of prediction accuracy comparing with GALF-P and GAME, the algorithm still failed to predict on some datasets which contain less conserved binding sites.

4 Conclusion

In this study, we propose a novel genetic algorithm based DNA motif discovery application IGAMD. Public domain information is introduced to construct the initial motif model for reducing the entire search space. Two model quality metrics are combined together as the fitness function to provide a stable evolution process. With a model refinement procedure, it can potentially increase the prediction precision. As the evidences shown in the result section, IGAMD demonstrates a comparable algorithm performance across 10 datasets against two GA approaches. The major issue for us is to further improve the reliability and the robustness of IGAMD for the low signal-to-noise data. A more comprehensive model representation along with a new fitness function is believed to enhance the predicting ability to handle less conserved binding sites.

Acknowledgments

The first author would like to thank Dr Tim Sawbridge at Victorian AgriBio-sciences Centre for his kind support on degree study.

References

1. Bailey, T.L., Elkan, C.: Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In: Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, pp. 28–36. AAAI Press, Menlo Park (1994)
2. Chan, T.-M., Leung, K.-S., Lee, K.-H.: TFBS identification based on genetic algorithm with combined representations and adaptive post-processing. Bioinformatics 24, 341–349 (2008)
3. Congdon, C.B., Aman, J.C., Nava, G.M., Gaskins, H.R., Mattingly, C.J.: An Evaluation of Information Content as a Metric for the Inference of Putative Conserved Noncoding Regions in DNA Sequences Using a Genetic Algorithms Approach. IEEE/ACM Trans. on Computational Biology and Bioinformatics 5, 1–14 (2008)
4. Galas, D.J., Schmitz, A.: DNase footprinting: a simple method for the detection of protein-DNA binding specificity. Nucleic Acids Res. 5, 3157–3170 (1978)
5. Hu, J., Li, B., Kihara, D.: Limitations and potentials of current motif discovery algorithms. Nucleic Acids Res. 33, 4899–4913 (2005)
6. Huerta, A.M., Salgado, H., Thieffry, D., Collado-Vides, J.: RegulonDB: a database on transcriptional regulation in *Escherichia coli*. Nucleic Acids Res. 26, 55–59 (1998)
7. Li, L.P., Liang, Y., Bass, R.L.L.: GAPWM: A Genetic Algorithm Method for Optimizing a Position Weight Matrix. Bioinformatics 23, 1188–1194 (2007)
8. Liu, X.S., Brutlag, D.L., Liu, J.S.: An algorithm for finding protein-DNA binding sites with applications to chromatin immunoprecipitation microarray experiments. Nature Biotechnology 20, 835–839 (2002)
9. Mahony, S., Hendrix, D., Golden, A., Smith, T.J., Rokhsar, D.S.: Transcription factor binding site identification using the Self-Organizing Map. Bioinformatics 21, 1807–1814 (2005)
10. Shaw Jr., W.M., Burgin, R., Howell, P.: Performance standards and evaluations in ir test collections: cluster-based retrieval models. Information Processing & Management 33, 1–14 (1997)
11. Stormo, G.D., Fields, D.S.: Specificity, free energy and information content in protein-DNA interactions. Trends in Biochemical Sciences 23, 109–113 (1998)
12. Tompa, M., Li, N., Bailey, T.L., et al.: Assessing computational tools for the discovery of transcription factor binding sites. Nature Biotechnology 23, 137–144 (2005)
13. van Helden, J., André, B., Collado-Vides, J.: Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. Journal of Molecular Biology 281, 827–842 (1998)
14. Wei, Z., Jensen, S.T.: GAME: detecting cis-regulatory elements using a genetic algorithm. Bioinformatics 22, 1577–1584 (2006)

A Hybrid Model for Prediction of Peptide Binding to MHC Molecules

Ping Zhang¹, Vladimir Brusic^{1,2}, and Kaye Basford¹

¹ The University of Queensland, School of Land, Crop and Food Sciences,
QLD 4072, Australia

² Cancer Vaccine Center, Dana-Farber Cancer Institute, Boston MA, USA
p.zhang2@uq.edu.au, vladimir_brusic@dfci.harvard.edu,
k.e.basford@uq.edu.au

Abstract. We propose a hybrid classification system for predicting peptide binding to major histocompatibility complex (MHC) molecules. This system combines Support Vector Machine (SVM) and Stabilized Matrix Method (SMM). Its performance was assessed using ROC analysis, and compared with the individual component methods using statistical tests. The preliminary test on four HLA alleles provided encouraging evidence for the hybrid model. The datasets used for the experiments are publicly accessible and have been benchmarked by other researchers.

1 Introduction

The concept of peptide vaccines uses the basic principle that T lymphocytes recognize antigens as peptide fragments that are generated from degradation of protein antigens. Peptides are subsequently bound to MHC class I or II molecules and displayed on the surface of the antigen presenting cells. Identification of epitopes and peptides that can bind MHC molecules is important for understanding the basis of cellular immune responses and the design of vaccines. Peptides that are presented by MHC molecules and recognized by T cells are termed T-cell epitopes. Human MHC is known as human leukocyte antigen (HLA).

Prediction of MHC binding peptides and T-cell epitopes has become one of the popular areas of bioinformatics applications in immunology. Statistical modeling and machine learning techniques have been used to build the prediction models based on the available databases. Recent analysis by Lin et al [1] showed that Matrix-based Models, Artificial Neural Networks (ANN) and Support Vector Machines (SVM) produce high accuracy in predicting the binding or nonbinding peptides to some MHC molecules. Because of the large number of different HLA molecules, where more than 2000 variants have been identified in humans, and variable performance of prediction methods for different HLA molecules, it is not possible to identify the best model for this type of prediction. Hybrid models were proposed, aimed at improvement of prediction performance. Moutaftsi et al [2] performed prediction of HLA binding peptides using the combination of prediction results from multiple matrices including Udaka [3], Parker [4], ARB [5,6] and SMM [7]. Bhasin and Raghava [8]

proposed a hybrid prediction model which combined a quantitative matrix based approach (QM) and a neural network (NN) approach. This method was implemented as nHLAPred server (<http://www.imtech.res.in/raghava/nhlapred>) and reported that the NN and QM can complement each other, leading to reduction in false prediction.

The performance of prediction models can be evaluated by several measures, including classification rate, sensitivity (SE) and specificity (SP). A common measure for assessing prediction accuracy is using receiver operating characteristic (ROC) curve. ROC curve plots SE vs 1-SP for a full range of decision thresholds. The area under the ROC plot (AUC) gives the overall evaluation of the model. By convention the AUC values range between 1.0 (perfect skill) and 0.5 (zero skill). The bigger the value is, the better overall performance of the model is represented. More details about ROC curve and its applications can be found in [9].

In this paper, we studied a hybrid method which combines the SVM and stabilized matrix method (SMM). The performance of the models was evaluated using ROC curve and compared with the single models.

2 SVM and Matrix Techniques for Prediction of MHC Binding Peptides

2.1 Support Vector Machine (SVM)

SVM is a kernel based machine learning technique that originated in modern statistical learning theory [10]. SVM can make data sets linearly separable by transferring them into a feature space of higher dimension. SVMs have excellent generalizing capability [11]. They use kernel functions to map the data; the most popular kernels use linear, polynomial, and Gaussian function. SVM can be applied for both classification and regression tasks. Some reports state that SVMs outperformed most other systems in a wide variety of applications [12]. SVMs have been used to predict MHC-binding peptides and they were reported as highly accurate. Dönnes and Elofsson [13] developed a model SVMHC based on support vector machines to predict the binding of peptides to MHC class I molecules. They compared its performance with two profile based methods, SYFPEITHI [14] and HLA_BIND [4] and claimed a slightly better result. Bhasin and Raghava [15] applied SVM for prediction of peptides binding with the MHC class II allele HLA-DRB1*0401. Their results showed good performance of SVM compared with other classification methods including Matrices, Motifs and ANN.

Zhang et al [16, 17] have also shown the advantages of SVM for predicting the MHC binding peptides. They reported a system based on SVM that outperformed ANN and Hidden Markov Model (HMM) [17], and then improved the model by adding the data representation of peptide/MHC interaction [16]. SVM regression models (SVR) were also used for prediction of peptide binding affinity to MHC molecules [18] and for binding/nonbinding peptide classification based on certain output cutoff values [19].

These models mainly used the binary coding of peptide sequences as the input variables. Zhao et al. [20] encoded each amino acid in the peptide sequences by ten factors which were obtained from 188 physical properties of 20 AAs via multivariate statistical analyses by Scheraga's group [21] for their SVM classification model. The SVM model was used for T-cell epitopes prediction and was also compared with

ANN and Z-Matrix based approaches [22]. Cui et al [23] also reported high performance of SVM for a variety of HLA alleles including 18 class I and 12 class II, and prediction of newly reported epitopes with high accuracy (11 out of 15).

In general, SVM has been reported by many researchers to perform well for prediction of MHC binding peptides and epitopes regardless of the kernels which were used with the models. This demonstrated the ability of SVMs to build effective predictive models when the dimensionality of the data is high and the number of observations is limited [11].

On the other hand, it has also been claimed that the performance of SVM strongly depends on both the quality and quantity of data. It fails if an unbalanced data set is used for learning. Riedesel et al [24] used the least square optimization method (LSM) with a weighting procedure to deal with asymmetric data sets with a small number of binding and a large number of non-binding peptides. This approach was expected to yield higher prediction accuracy than SVM. However, in their paper the assessment for the model performance using ROC curve showed better performance of SVM than LSM.

There are three important algorithms for training SVM: Chunking, Sequential Minimum Optimization, and SVM^{light} [25]. SVM^{light} is an implementation of an SVM learner which addresses the problem of large tasks. It decomposes the problem with many training examples into a series of smaller tasks. The main advantage of this decomposition is that it suggests algorithms with memory requirements linear in the number of training examples and in the number of support vectors [26].

2.2 Stabilized Matrix Method (SMM)

The Matrix methods assign scores for 20 amino acids (AAs) to all positions of MHC grooves. These scores are calculated based on the frequencies of the AAs appearing at the positions within a large number of binding experiments [27]. Brief reviews of methods for building matrix models can be found in [11] and [27]. Studies have shown that simple predictions using scoring matrices yield reasonably good results when little experimental data are available (typically tens to hundreds of peptides) compared with the size of the sequence space (20^9 for 9-mers). This indicates that the relationship between sequence and affinity can be approximated by the independent binding assumption, i.e. amino acids at different positions of a peptide contribute independently to the overall binding affinity of the peptide [5].

Peters et al. [5] developed a matrix-based algorithm called the Stabilized Matrix Method (SMM). First they create a matrix by using a frequency-based formula then modify the matrix to compensate for the errors contained in experimental data. This method was first tested on prediction of peptide binding HLA-A*0201 molecule using a set of 9-mer peptides. The performance of SMM was compared with three widely used matrix-based methods: BIMAS [4], SYFPEITHI [14] and the polynomial method (PM) [28]. SMM considerably outperforms the other methods on three independent test sets. This method was then less, peptide transport by the transporter associated with antigen presentation (TAP) and proteasomal cleavage of protein sequences [6]. The implementation of the algorithm successfully applied to predicting peptide binding to MHC molecules for creating the matrix has been made publicly available (<http://www.mhc-pathway.net/smm>).

3 Hybrid Model

3.1 Methodology

We propose a hybrid prediction model that combines SVM and SMM methods, as depicted in Figure 1. A matrix is built based on the SMM approach with the protein peptide sequences as the input. The output values of the SMM model are combined with the binary code of the sequences for the final classification of the peptides as the binding or nonbinding to the MHC-I molecules. The hybrid mode can be described in the following steps:

- 1) Use the training data to build the SMM matrix model and test on the test set. Save the output from both the training and test data.
- 2) Normalize the output from the matrix model. The normalization is performed using every output value for every peptide divided by the maximum output value of the training set.
- 3) Create the input vector for every peptide with the binary coding of the original sequence plus the output from the SMM matrix model. Each amino acid in the peptide can be encoded as a binary string of length 20 with one position set to “1” and the other positions set to “0”. The binary code of a nonamer peptide will be represented by the binary string of length 180. With the output from the matrix model added, the input number to the SVM will be 181.
- 4) Build the SVM model using the training data for final classification.
- 5) Test the model using test data.

3.2 Datasets

Peters et al. [29] presented a dataset that was used for benchmarking the MHC/peptide binding prediction models. The entire dataset can be downloaded from the publicly accessible website mhcbindingpredictions.immuneepitope.org (IEDB). They used this data set to compare performances of three prediction methods (ARB, SMM and ANN) developed in their lab. They also provided training and testing data sets used in their study. In this research the same training and testing 9-mer peptide datasets of four HLA class I alleles including HLA-A*0101, HLA-A*0201 HLA-A*0202 and HLA-A*0301 have been used in our experiments. The numbers of peptides included in the datasets are 1157, 3089, 1447 and 2094 respectively.

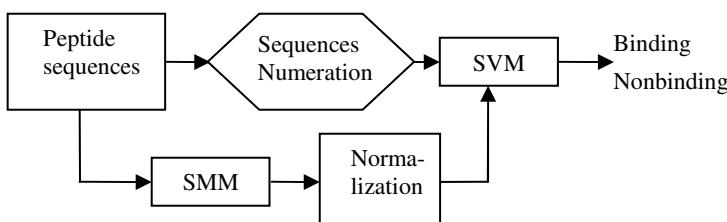


Fig. 1. The hybrid model with SMM matrix and SVM

3.3 Implementation and Experimental Results

In this research, we used SVM^{light} (http://www-ai.cs.uni-dortmund.de/svm_light) for all the SVM implementation. Additional C code to adjust the parameters was added to the original code for simulations. The SMM C code for UNIX was downloaded from the source and used to create the matrices using benchmark datasets.

For our experiments, the dataset of each allele were split into five equally sized subsets. In each partition, one subset was used for testing, while other subsets were combined and used for training. As a result, five prediction models were built for each allele.

To evaluate the performance of the models, the ROC curves were drawn for every test set (5 for each HLA variant) independently. To compare the hybrid model performance with the single SMM and SVM models, the experiments using SMM and SVM were run with the same datasets and the AUC value were calculated. Figure 2 shows the AUC values for the different models for the 5 datasets for each allele. The horizontal axis represents the dataset used for each allele, and the vertical axis is the AUC value. From the figure, we can see a small improvement in the performance of the hybrid model. However, the AUC values of the hybrid model are all higher than or equal to the values from the SVM, although the differences are not large. There was only one value (out of 20) from SMM higher than the corresponding hybrid value.

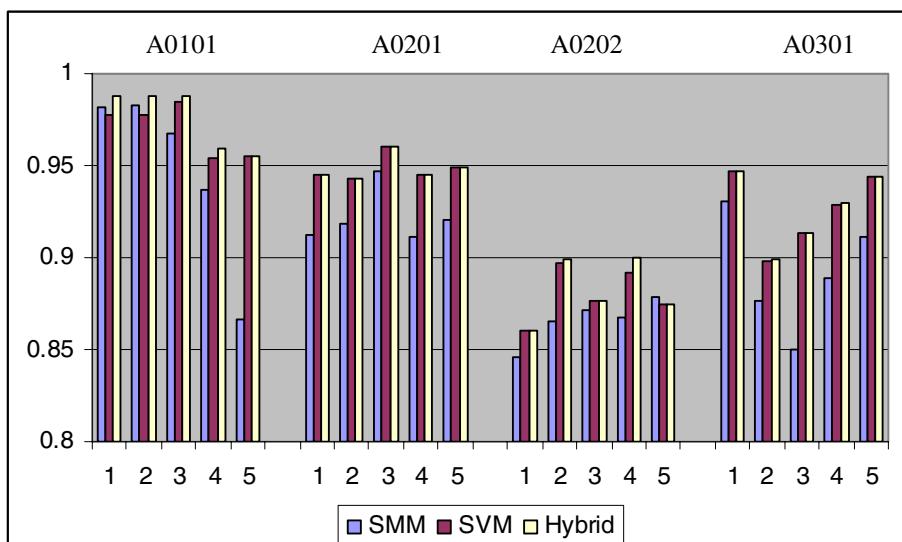


Fig. 2. AUC values of different models for each allele

Table 1 lists the average AUC values of the ROC curves for the three models for different HLA alleles. To determine whether the hybrid model performs better than the single methods, two-way analysis of variance (ANOVA) was performed for each allele. The p-values from the analysis are also shown in this table. The Least Significant Difference (LSD) was calculated to compare the model mean values. The results

are listed in the same table. The p values show that for 3 variants, the hybrid models have shown statistically significant improvement, while there was no difference for HLA-A*0101. The LSDs show the significant difference of the SMM from other two models, but do not show a significant difference between SVM and the hybrid mode.

Table 1. The average AUC values of each allele from the three models and ANOVA test results

Model	Allele			
	A0101	A0201	A0202	A0301
SMM	0.947 a	0.922 b	0.866 b	0.892 b
SVM	0.970 a	0.948 a	0.880 a	0.926 a
Hybrid	0.974 a	0.948 a	0.882 a	0.927 a
P-value (within allele)	0.1734	0.00003	0.0425	0.0011

Note. Within an allele, means followed by the same letter are not significantly different from one another at the 5% significance level.

4 Discussion and Conclusion

This research proposed the idea of combining the Matrix approach and the SVM. It used the SVM with the traditional kernels and the most used encoding for the input peptide sequences. The preliminary test of the hybrid model, based on a limited number of alleles, showed encouraging results. The ANOVA test on the model means did not show significant difference between the SVM and the hybrid model, however, the hybrid model did not perform worse than the SVM in any single test. This indicates that there is potential for further development of the hybrid model. In future research, new kernel functions such as kernels combining biological features [30] and more efficient encoding or input vectors can be tested with the hybrid system. Different matrix approaches can also be applied in the hybrid system. As the performance of SVM is dependent on the training data, the optimization of training data can also be considered in future work.

Acknowledgements

The authors acknowledge the support of the ImmunoGrid project, EC contract FP6-2004-IST-4, No. 028069.

References

- Lin, H.H., Ray, S., Tongchusak, S., Reinherz, E., Brusic, V.: Evaluation of HLA Class I Peptide Binding Prediction Servers: Applications for Vaccine Research. *BMC Immunol.* 9, 8 (2008)

2. Moutafts, M., Peters, B., Pasquetto, V., Tscharke, D.C., Sidney, J., Bui, H., Grey, H., Sette, A.: A Consensus Epitope Prediction Approach Identifies the Breadth of Murine TCD8+ Cell Responses to Vaccinia Virus. *Nature Biotechnology* 24(7), 817–819 (2006)
3. Udaka, K., Wiesmuller, K.H., Kienle, S., Jung, G., Tamamura, H., et al.: An Automated Prediction of MHC Class I - Binding Peptides Based on Positional Scanning with Peptide Libraries. *Immunogenetics* 51, 816–828 (2000)
4. Parker, K.C., Bednarek, M.A., Coligan, J.E.: Scheme for Ranking Potential HLA-A2 Binding Peptides Based on Independent Binding of Individual Peptide Side-Chains. *J. Immunol.* 152, 163–175 (1994)
5. Peters, B., Tong, W., Sidney, J., Sette, A., Weng, Z.: Examining the Independent Binding Assumption for Binding of Peptide Epitopes to MHC-I Molecules. *Bioinformatics* 19, 1765–1772 (2003)
6. Peters, B., Sette, A.: Generating Quantitative Models Describing the Sequence Specificity of Biological Processes with the Stabilized Matrix Method. *BMC Bioinformatics* 6, 132 (2005)
7. Bui, H.H., Sidney, J., Peters, B., Sathiamurthy, M., Sinichi, A., Purton, K.A., Mothe, B.R., Chisari, F.V., Watkins, D.I., Sette, A.: Automated Generation and Evaluation of Specific MHC Binding Predictive Tools: ARB Matrix Applications. *Immunogenetics* 57, 304–314 (2005)
8. Bhasin, M., Raghava, G.P.S.: A Hybrid Approach for Predicting Promiscuous MHC Class I Restricted T Cell Epitopes. *J. Biosci.* 32, 31–42 (2006)
9. Zweig, M.H., Campbell, G.: Receiver-Operating Characteristic (ROC) Plots: A Fundamental Evaluation Tool in Clinical Medicine. *Clinical Chemistry* 39(4), 561–577 (1993)
10. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
11. Tsurui, H., Takahashi, T.: Prediction of T-Cell Epitope. *Journal of Pharmacological Sciences* 105, 299–316 (2007)
12. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other Kernel-based Learning Methods (2000)
13. Dönnes, P., Elofsson, A.: Prediction of MHC Class I Binding Peptides, Using SVMHC. *BMC Bioinformatics* 3, 25 (2002)
14. Rammensee, H., Bachmann, J., Emmerich, N.N., Bachor, O.A., Stevanovic, S.: SYFPEITHI: Database for MHC Ligands and Peptide Motifs. *Immunogenetics* 50, 213–219 (1999)
15. Bhasin, M., Raghava, G.P.: SVM Based Method for Prediction HLA-DRB1*401 Binding Peptides in an Antigen Sequence. *Bioinformatics* 20, 421–423 (2004)
16. Zhang, G.L., Bozic, I., Kwok, C.K., August, J.T., Brusic, V.: Prediction of Supertype-specific HLA Class I Binding Peptides Using Support Vector Machines. *Journal of Immunological Methods* 320(1-2), 143–154 (2007)
17. Bozic, I., Zhang, G.L., Brusic, V.: Predictive Vaccinology: Optimisation of Predictions Using Support Vector Machine Classifiers. In: Gallagher, M., Hogan, J.P., Maire, F. (eds.) IDEAL 2005. LNCS, vol. 3578, pp. 375–381. Springer, Heidelberg (2005)
18. Lui, W., Meng, X., Xu, Q., Flower, D.R., Li, T.: Quantitative Prediction of Mouse Class I MHC Peptide Binding Affinity Using Support Vector Machine Regression (SVR) Models. *BMC Bioinformatics* 7, 182 (2006)
19. You, L., Zhang, P., Bodén, M., Brusic, V.: Understanding prediction systems for HLA-binding peptides and T-cell epitope identification. In: Rajapakse, J.C., Schmidt, B., Volkert, L.G. (eds.) PRIB 2007. LNCS (LNBI), vol. 4774, pp. 337–348. Springer, Heidelberg (2007)

20. Zhao, Y., Pinilla, C., Valmori, D., Martin, R., Simon, R.: Application of Support Vector Machines for T-Cell Epitopes Prediction. *Bioinformatics* 19, 1978–1984 (2003)
21. Kidera, A., Konishi, Y., Oka, M., Ooi, T., Scheraga, H.A.: Statistical Analysis of the Physical Properties of the 20 Naturally Occuring Amino Acids. *J. Protein Chem.* 4, 23–55 (1985)
22. Zhao, Y., Gran, B., Pinilla, C., Markovic-Plese, S., Hemmer, B., Tzou, A., Whitney, L.W., Biddison, W.E., Martin, R., Simon, R.: Combinatorial Peptide Libraries and Biometric Score Matrices Permit the Quantitative Analysis of Specific and Degenerate Interactions Between Clonotypic T-Cell Receptors and MHC–Peptide Ligands. *J. Immunol.* 167, 2130–3141 (2001)
23. Cui, J., Han, L.Y., Lin, H.H., Zhang, H.L., Tang, Z.Q., Zheng, C.J., Cao, Z.W., Chen, Y.Z.: Prediction of MHC-binding Peptides of Flexible Lengths from Sequence-derived Structural and Physicochemical Properties. *Mol. Immunol.* 44, 866–877 (2007)
24. Riedsel, H., Kolbeck, B., Schmetzer, O., Knapp, E.W.: Peptide Binding at Class I Major Histocompatibility Complex Scored with Linear Functions and Support Vector Machines. *Genome Informatics* 15(1), 198–212 (2004)
25. Dong, J., Suen, C.Y.: A Fast SVM Training Algorithm. *International Journal of Pattern Recognition and Artificial Intelligence* 17(3), 367–384 (2003)
26. Joachims, T. (ed.): *Making Large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge (1999)
27. Yu, K., Petrovsky, N., Schonbach, C., Koh, J.Y.L., Brusic, V.: Methods for Prediction of Peptide Binding to MHC Molecules: A Comparative Study. *Mol. Med.* 8, 137–148 (2002)
28. Gulukota, K., Sidney, J., Sette, A., DeLisi, C.: Two Complementary Methods for Predicting Peptides Binding Major Histocompatibility Complex Molecules. *J. Mol. Biol.* 267, 258–1267 (1997)
29. Peters, B., Bui, H.H., Frankild, S., Nielsen, M., Lundegaard, C., et al.: A Community Resource Benchmarking Predictions of Peptide Binding to MHC-I Molecules. *Plos Computational Biology* 2(6), 574–584 (2006)
30. Yang, Z.R., Johnson, F.C.: Prediction of T-cell epitopes Using Biosupport Vector Machines. *J. Chem. Inf. Model* 45(5), 1424–1428 (2005)

Part V

Special Session: Data Mining Methods for Cybersecurity

An Evaluation of Machine Learning-Based Methods for Detection of Phishing Sites

Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi

Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, Japan
`{daisu-mi,hiroa-ha,youki-k}@is.naist.jp`

Abstract. In this paper, we present the performance of machine learning-based methods for detection of phishing sites. We employ 9 machine learning techniques including AdaBoost, Bagging, Support Vector Machines, Classification and Regression Trees, Logistic Regression, Random Forests, Neural Networks, Naive Bayes, and Bayesian Additive Regression Trees. We let these machine learning techniques combine heuristics, and also let machine learning-based detection methods distinguish phishing sites from others. We analyze our dataset, which is composed of 1,500 phishing sites and 1,500 legitimate sites, classify them using the machine learning-based detection methods, and measure the performance. In our evaluation, we used f_1 measure, error rate, and Area Under the ROC Curve (AUC) as performance metrics along with our requirements for detection methods. The highest f_1 measure is 0.8581, the lowest error rate is 14.15%, and the highest AUC is 0.9342, all of which are observed in the case of AdaBoost. We also observe that 7 out of 9 machine learning-based detection methods outperform the traditional detection method.

1 Introduction

Phishing is a form of identity theft whose targets are users rather than computer systems. A phishing attacker attracts victims to a spoofed web site, a so-called “phishing site”, and attempts to persuade them to send their personal information.

To prevent a user from browsing phishing sites, there are two distinct approaches. One is URL filtering. It detects phishing sites by comparing the URL of a site a user visits with a URL blacklist composed of the URLs of phishing sites. However, it is difficult to build a perfect blacklist due to the rapid increase of phishing sites. According to trend reports published by the Anti-Phishing Working Group [1], the number of reported phishing sites was 23,917 in July 2007, far surpassing the 14,315 in July 2005.

The other approach is a heuristic-based solution. A heuristic is an algorithm to distinguish phishing sites from others based on users’ experience, that is, a heuristic checks if a site seems to be a phishing site. A heuristic-based solution employs several heuristics and converts results from each heuristic into a vector. Based on the vector, the heuristic-based solution calculates the likelihood of a

site being a phishing site and compares the likelihood with the defined discrimination threshold. Different from URL filtering, a heuristic-based solution has a possibility to identify new phishing sites.

Unfortunately, the detection accuracy of existing heuristic-based solutions is far from suitable for practical use [2], even if various studies [3, 4, 5] discovered heuristics. To improve the detection accuracy, both discovering innovative heuristics and refining the calculation algorithm of the likelihood are important.

In our previous work [6], we attempted to employ a machine learning technique to improve the detection accuracy. We employed AdaBoost, a machine learning technique, as a calculation method of the likelihood. Our preliminary evaluation showed the AdaBoost-based detection method can achieve higher detection accuracy than the traditional detection method.

Here, we present a performance evaluation of 9 Machine Learning-based Detection Methods (MLBDMs) including AdaBoost, Bagging, Support Vector Machines (SVM), Classification and Regression Trees (CART), Logistic Regression (LR), Random Forests (RF), Neural Networks (NN), Naive Bayes (NB) and Bayesian Additive Regression Trees (BART). In the evaluation, we used f_1 measure, error rate, Area Under the ROC Curve (AUC) as performance metrics along with our requirements for detection methods. Our requirements are (i) they must achieve high detection accuracy, (ii) they must adjust their detection strategies for web users.

We let all MLBDMs classify whether a site is a phishing site or not by using a dataset of 3,000 URLs, composed of 1,500 phishing sites and the same number of legitimate sites. We employ 8 heuristics presented in CANTINA [7] and measure their performance in a less biased way. The results show that the highest f_1 measure is 0.8581, the lowest error rate is 14.15%, and the highest AUC is 0.9342, all of which are observed in the case of AdaBoost.

The rest of this paper is organized as follows: In Section 2, we present our related work. In Section 3, we describe our evaluation conditions, and we show our experimental results in Section 4. We show our future work in Section 5 and summarize our contributions in Section 6.

2 Related Work

For mitigating phishing attacks, machine learning, which facilitates the development of algorithms or techniques by enabling computer systems to learn, has begun to garner attention. PFILTER, which was proposed by Fette et al. [8], employed SVM to distinguish phishing emails from other emails. According to [9], Abu-Nimeh et al. compared the predictive accuracy of several machine learning methods including LR, CART, RF, NB, SVM, and BART. They analyzed 1,117 phishing emails and 1,718 legitimate emails with 43 features for distinguishing phishing emails. Their research showed that the lowest error rate was 7.72% in the case of Random Forests. In [10], Ram Basnet et al. performed an evaluation of six different machine learning-based detection methods. They analyzed 973 phishing emails and 3,027 legitimate emails with 12 features, and showed that

the lowest error rate was 2.01%. The experimental conditions were different between [9] and [10], however, the machine learning provided high accuracy for the detection of phishing emails.

Aside from phishing emails, a machine learning method was also used to detect phishing sites. According to [11], Pan et al. presented an SVM-based page classifier for detection of phishing sites. They analyzed 279 phishing sites and 100 legitimate sites with 8 features, and the results showed that the average error rate was 16%.

Our previous work [6] employed AdaBoost for the detection of phishing sites. We checked 100 phishing sites and the same number of legitimate sites with 7 heuristics. Our performance evaluation showed that the average error rate was 14.7%.

We find that there are two problems in earlier research. One is that the number of features for detecting phishing sites is lesser than that for detecting phishing emails. It indicates that the detection of phishing sites is much difficult than that of phishing emails. The other is that no research contribution confirmed whether any kind of MLBDMs were available to distinguish phishing sites from legitimate sites. To the best of our knowledge, earlier research tested only one machine learning technique. In this paper, we evaluate 9 MLBDMs and show their performance by measuring the performance.

3 Evaluation Approach

In this paper, we evaluate the performance of MLBDMs. We let each machine learning method combine heuristics, perform supervised learning from the dataset, and distinguish phishing sites from other sites.

In this section, we define metrics of the performance evaluation along with our requirements for MLBDMs. We then decide the heuristics that we used in our evaluation, and describe how we construct a dataset for both training and testing. Finally, we explain the preliminary set-up of our experiments.

3.1 Evaluation Metrics

First, we defined metrics for evaluating performance along with requirements for detection methods. Our requirements were as follows.

1. Accuracy

An MLBDM must achieve high detection accuracy. User safety would obviously be compromised if phishing prevention systems labeled phishing sites as legitimate. Users would also complain if prevention systems labeled legitimate sites as phishing sites because of the interruption in browsing caused by prevention systems.

2. Adjustment Capability

An MLBDM must adjust its strategy for detecting phishing sites for web users. If a user is a novice, who is easily taken in by phishing attacks, phishing prevention systems should decrease false negative errors instead of increasing

Table 1. Test Result

	actual phishing sites	actual legitimate sites
predict phishing sites	tp	fp
predict legitimate sites	fn	tn

false positive errors. Conversely, if a user is a security expert, the system should focus on decreasing false positive errors.

For Requirement 1, we used the f_1 measure (higher is better) and the error rate (lower is better) as metrics to evaluate the detection accuracy. Statistically, f_1 measure has been used as an index of a test's accuracy. This measure is the harmonic mean of precision and recall, so calculated by $2 \cdot p \cdot r / (p + r)$, where p is the precision and r is the recall of the test. Given the test result as shown in Table 1, p is $tp / (tp + fp)$ and r is $tp / (tp + fn)$. The average error rate has been also a reasonable metric to indicate the detection accuracy. It is calculated by $(fp + fn) / (tp + tn + fp + fn)$, dividing the number of incorrectly identified sites by the number of all sites in the dataset.

For Requirement 2, we performed Receiver Operating Characteristic (ROC) analysis. Generally, detection methods calculate the likelihood of being phishing sites L and compare the likelihood with the defined discrimination threshold θ . In our experiment, MLBDMs distinguish a phishing site by checking if L is less or equal than $\theta (= 0)$. Imagine that θ was higher than 0. In this case, MLBDMs would tend to label a site as phishing rather than as legitimate. Conversely, MLBDMs would tend to label a site as legitimate if θ was lower than 0. Accordingly, we assumed that adjusting θ provides different detection strategies. Based on this assumption, we employed ROC analysis because it has been widely used in data analysis to study the effect of varying the threshold on the numerical outcome of a diagnostic test. We also used the Area Under the ROC curve (AUC; higher is better) as a metric to evaluate adjustment capability.

3.2 Heuristics

In our evaluation, we employ 8 heuristics, namely Age of Domain, Known Images, Suspicious URL, Suspicious Links, IP Address, Dots in URL, Forms, and TF-IDF-Final heuristics, all of which were employed in CANTINA [7]. To the best of our knowledge, the most successful tool for combining heuristics is CANTINA, which has achieved high accuracy of detecting phishing sites without using the URL blacklist. In CANTINA, the likelihood of the phishing site is calculated from weighted majority by using 8 heuristics. without using machine learning techniques.

3.3 Dataset

We then built a dataset with the criteria for choosing URLs. Based on the criteria in the original CANTINA, we collected URLs with the same number of

phishing sites and legitimate sites. All sites were English language sites because CANTINA does not work well if the sites are not written in English. First, we chose 1,500 phishing sites that were reported on PhishTank.com [12] from November, 2007 to February, 2008. Second, we also selected 227 URLs from 3Sharp's study of anti-phishing toolbars [13]. There were listed 500 URLs of legitimate sites in [13], however, we could not connect to many listed URLs. Third, we attempted to collect 500 URLs from Alexa Web Search [14] and observed 477 URLs. Finally, we gathered 796 URLs from yahoo random link [15].

Each site was checked with our implementation of heuristics, and was converted into a vector $\mathbf{x} = (x_1, x_2 \dots x_p)$, where $x_1 \dots x_p$ are the values corresponding to a specific feature. The dataset consisted of 8 binary explanatory variables and 1 binary response variable.

To perform our evaluation in a less biased way, we employed 4-fold cross validation. Furthermore, our cross validation was repeated 10 times in order to average the result.

3.4 Experimental Set-Up

We adjusted the parameters for MLBDMs to minimize the error rate in training. For decision tree-based machine learning techniques such as RF, we tested them using different numbers of trees, namely 100, 200, 300, 400, and 500 trees. The minimum error rate (14.27%) was observed when the number of trees was 300, followed by 200 and 400 (14.28%), 500 (14.30%), and 100 (14.37%). Thus, we set the number of trees to 300 for RF-based detection methods.

The iteration time was set to 500 in all of our experiments if the machine learning technique needed to analyze iteratively for reducing training errors. The minimum error rate (14.27%) was observed when the number of iterations was 500, followed by 300 and 400 (14.28%), 200 (14.30%), and 100 (14.31%). In addition, finding the optimal iteration number is important, however, the choice of the exact value of the optimal iteration number is not often a critical issue since the increase in test error is relatively slow.

For some types of machine learning techniques, we used threshold value to approximate the prediction output. For example, BART is designed for regression, not for classification. Therefore, BART gives quantitative value whereas we need an MLBDM to output binary value that indicates whether a site is a phishing site or not. In such cases, we employed threshold value and observed if the result of BART regression was greater than the threshold. We decided the threshold in the same fashion as the original CANTINA. In the case of CANTINA, the maximum likelihood of being a phishing site is -1 and that of being a legitimate site is 1; therefore, it employs the middle value 0 as the threshold value.

In SVM, we tested both linear and non-linear kernel functions. The average error rate in training by using *Radial Based Function* (RBF), one of the typical non-linear kernel functions, was 14.18%, less than 21.02% of linear kernel. Thus, we used RBF in our experiments.

In NN, we selected the number of units in the hidden layer, namely 1, 2, 3, 4, and 5 units, for finding the minimum average error rate. The minimum error rate

Table 2. Precision, Recall and f_1 measure, False Positive Rate(FPR), False Negative Rate(FNR), Error Rate, and AUC

	Precision	Recall	f_1 measure	FPR	FNR	ER	AUC
AdaBoost	0.8614	0.8551	0.8581	13.83%	14.49%	14.15%	0.9342
Bagging	0.8492	0.8573	0.8527	15.36%	14.27%	14.82%	0.9231
SVM	0.8629	0.8498	0.8562	13.57%	15.02%	14.29%	0.8926
CART	0.8330	0.8542	0.8384	18.16%	14.58%	16.37%	0.9062
LR	0.8510	0.8588	0.8548	15.10%	14.12%	14.60%	0.9172
RF	0.8566	0.8546	0.8554	14.37%	14.54%	14.45%	0.9296
NN	0.8633	0.8512	0.8570	13.54%	14.88%	14.21%	0.9310
NB	0.8464	0.8636	0.8547	15.74%	13.64%	14.69%	0.9215
BART	0.8567	0.8550	0.8555	14.39%	14.50%	14.45%	0.9321
CANTINA	0.9134	0.6519	0.7606	06.21%	34.81%	20.52%	0.9162

Table 3. FPR given FNR rate < 5.00%, and FNR given FPR rate < 5.00%

	FP(FN< 5.00%)	FN(FP< 5.00%)
AdaBoost	30.65%	26.15%
Bagging	34.28%	30.35%
SVM	62.10%	71.10%
CART	46.24%	36.77%
LR	33.90%	41.27%
RF	31.56%	25.51%
NN	31.56%	26.48%
NB	31.86%	35.55%
BART	31.35%	25.76%

(14.14%) was observed when the number of units was 5, followed by 4 (14.23%), 2 (14.46%), 3 (15.48%), and 1 (16.03%).

4 Evaluation

In this section, we evaluate the performance of all MLBDMs by measuring f_1 measure, error rate and AUC, and studying them comparatively. We also compare MLBDMs with the original CANTINA.

First, we measured the accuracy of all MLBDMs. We calculated Precision, Recall and f_1 measure for each pattern of dataset respectively, and also calculated their average as shown in Table 2. The highest f_1 measure was 0.8581 in AdaBoost, followed by NN (0.8570), SVM (0.8562), BART (0.8555), RF (0.8554), LR (0.8548), NB (0.8547), Bagging (0.8527) and finally CART (0.8384).

We then calculated the false negative rate, false positive rate and error rate in Table 2. The lowest error rate was 14.15% in AdaBoost, followed by NN (14.21%), SVM (14.29%), RF and BART (14.45%), LR (14.60%), NB (14.69%), Bagging (14.82%), and finally CART (16.37%). The lowest false negative rate was 13.64% in NB, and the lowest false positive rate was 13.54% in NN.

We also calculated AUC as shown in Table 2. The highest AUC was 0.9342 in AdaBoost, followed by BART (0.9321), NN (0.9310), RF (0.9296), Bagging (0.9231), NB (0.9215), LR (0.9172), CART (0.9062), and finally SVM (0.8956). Finally, we compared all MLBDMs with CANTINA’s detection method. We evaluated the performance of CANTINA in the same way as that described in Section 3, and observed f_1 measure was 0.7607, error rate was 20.52%, and AUC was 0.9162 as shown in Table 2, respectively. According to our comparison, 7 out of 9 MLBDMs, namely AdaBoost, Bagging, LR, RF, NN, NB, and BART-based detection methods, outperformed CANTINA.

5 Future Work

In our future work, we will implement a phishing prevention system according to the detection result. Within such a system, we should adjust the discrimination threshold for each web user, as we mentioned in Section 3.1.

Table 3 shows the false positive rate when the false negative rate was less than 5.00%, and the false negative rate when the false positive rate was less than 5.00%. The lowest false positive rate was 30.65% in the case of AdaBoost, and the lowest false negative rate was 25.51% in the case of RF. This indicated that if novices could accept 30.65% of false positive errors, 95.00% of phishing sites would be blocked as phishing sites. Similarly, if security experts could accept 25.51% of false negative errors, 95.00% sites of legitimate sites would be browsed normally. It is beyond of scope of this paper, however, we need to decide the optimal threshold for each user by both measuring each user’s knowledge for the detection of phishing sites and estimating the acceptable level of false positives and false negatives.

6 Conclusion

In this paper, we evaluated the performance of machine learning-based detection methods (MLBDMs) including AdaBoost, Bagging, Support Vector Machines (SVM), Classification and Regression Trees (CART), Logistic Regression (LR), Random Forests (RF), Neural Networks (NN), Naive Bayes (NB) and Bayesian Additive Regression Trees (BART). Because we assumed that the detection method must be accurate and must have adjustment capability, we used f_1 measure, error rate and AUC as performance metrics in the evaluation.

We employed 8 heuristics presented in [7] and analyzed 3,000 URLs, which were composed of 1,500 phishing sites and the same number of legitimate sites. We performed 4-fold cross validation 10 times and measured the average of f_1 measure, error rate, and AUC.

The result showed that the highest f_1 measure was 0.8581 in AdaBoost, followed by NN (0.8570), SVM (0.8562), BART (0.8555), RF (0.8554), LR (0.8548), NB (0.8547), Bagging (0.8527) and finally CART (0.8384). The lowest error rate was 14.15% in AdaBoost, followed by NN (14.21%), SVM (14.29%), RF and BART (14.45%), LR (14.60%), NB (14.69%), Bagging (14.82%), and finally CART (16.37%). The highest AUC was 0.9342 in AdaBoost, followed by

BART (0.9321), NN (0.9310), RF (0.9296), Bagging (0.9231), NB (0.9215), LR (0.9172), CART (0.9062), and finally SVM (0.8956). Additionally, we plotted the ROC curve and found that all MLBDMs could achieve both high true positive rates and low false positive rates.

Acknowledgments

This work was a part of "Research and Development on Traceback Technologies in the Internet" sponsored by the National Institute of Information and Communications Technology(NICT).

References

1. Anti-Phishing Working Group: Phishing Activity Trends Report (July 2007)
2. Zhang, Y., Egelman, S., Cranor, L., Hong, J.: Phinding Phish: Evaluating Anti-Phishing Tools. In: Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS 2007) (2007)
3. Kumar, A.: Phishing - A new age weapon. Technical report, Open Web Application Secuirtry Project (OWASP) (2005)
4. Tally, G., Thomas, R., Vleck, T.V.: Anti-Phishing: Best Practices for Institutions and Consumers. Technical report, McAfee Research (2004)
5. Van der Merwe, A., Loock, M., Dabrowski, M.: Characteristics and responsibilities involved in a phishing attack. In: Proceedings of the 4th International Symposium on Information and Communication Technologies (ISICT 2005) (2005)
6. Miyamoto, D., Hazeyama, H., Kadobayashi, Y.: A Proposal of the AdaBoost-Based Detection of Phishing Sites. In: Proceedings of the 2nd Joint Workshop on Information security (2007)
7. Zhang, Y., Hong, J., Cranor, L.: CANTINA: A Content-Based Approach to Detect Phishing Web Sites. In: Proceedings of the 16th World Wide Web Conference (WWW 2007) (2007)
8. Fette, I., Sadeh, N.M., Tomasic, A.: Learning to detect phishing emails. In: Proceedings of the 16th International Conference on World Wide Web (WWW 2007) (2007)
9. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: Proceedings of eCrime Researchers Summit (eCryme 2007) (2007)
10. Basnet, R., Mukkamala, S., Sung, A.H.: Detection of phishing attacks: A machine learning approach. Studies in Fuzziness and Soft Computing 226, 373–383 (2008)
11. Pan, Y., Ding, X.: Anomaly based web phishing page detection. In: Proceedings of the 22nd Annual Computer Security Applications Conference on Annual Computer Security Applications Conference (ACSAC 2006) (2006)
12. OpenDNS: PhishTank - Join the fight against phishing,
<http://www.phishtank.com>
13. Robichaux, P., Ganger, D.L.: Gone Phishing: Evaluating Anti-Phishing Tools for Windows, <http://www.3sharp.com/projects/antiphishing/gone-phishing.pdf>
14. Alexa Internet, Inc.: Alexa the Web Information Company, <http://www.alexa.com>
15. Yahoo!Inc.: Random Yahoo Link, <http://random.yahoo.com/fast/ryl>

Detecting Methods of Virus Email Based on Mail Header and Encoding Anomaly

Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi

Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, Japan
`{daisu-mi, hiroa-ha, youki-k}@is.naist.jp`

Abstract. In this paper, we try to develop a machine learning-based virus email detection method. The key feature of this paper is employing Mail Header and Encoding Anomaly(MHEA) [1]. MHEA is capable to distinguish virus emails from normal emails, and is composed of only 5 variables, which are obtained from particular email header fields. Generating signature from MHEA is easier than generating signature by analyzing a virus code, therefore, we feature MHEA as signature to distinguish virus emails. At first, we refine the element of MHEA by association analysis with our email dataset which is composed of 4,130 virus emails and 2,508 normal emails. The results indicate that the one element of MHEA should not be used to generate MHEA. Next, we explore a way to apply MHEA into detection methods against virus emails. Our proposed method is a hybrid of matching signature from MHEA(signature-based detection) and detecting with AdaBoost (anomaly detection). Our preliminary evaluation shows that f_1 measure is 0.9928 and error rate is 0.75% in the case of our hybrid method, which outperforms other types of detection methods.

1 Introduction

Email viruses have become one of the major Internet security threats today. An email virus is a malicious program which hides in an email attachment, and become active when the attachment is opened. Currently, the most successful computer viruses spread via email [2], and the main infection source of virus is via email [3]. Thus, distinguishing virus-attached emails, called “*virus email*” from other emails is required to protect computer from viruses. Current approaches for dealing with email viruses rely on anti-virus softwares. Usually, anti-virus vendors capture and analyze viruses and generate the signature of each virus. However, a traditional signature-based approach is ineffective against newly-discovered viruses. Even anti-virus vendors endeavor to add new signature, it is hard to catch up with new viruses because modern viruses usually have both self-propagating and obfuscation functions.

According to [1], virus emails are distinguished from normal emails with Mail Header and Encoding Anomaly(MHEA). The MHEA is easily extracted from an email without deep analysis. The elements of MHEA can be obtained from particular email header fields which are rarely affected by variety of email viruses. Thus, MHEA has a possibility to deal with increasing the number of new viruses.

In this paper, we refine the Arai's proposal by association analysis with our email dataset, at first. Our dataset is composed of 4,130 virus emails and 2,508 normal emails. Arai suggested 5 variables on the email headers and/or anomaly on MIME encoding [1], however, we assume 1 variable, namely Encoded-Words length, can be affected by variety of email viruses. According to our verification, generating MHEA without this variable performs better than generating MHEA with this variable.

Next, we attempt to develop MHEA-based methods for detection of virus emails. We divide our dataset into two samples; one is used for training and one is used for testing. The training dataset has 2,065 virus emails and 1,259 normal emails, which are randomly sampled without replacement, respectively. We then employ MHEA obtained from training dataset as signature, and check if each email in testing dataset is registered in the signature. The result shows that 2,859 emails in testing dataset are classified virus emails or normal emails without error. The remaining 460 emails cannot be classified because the MHEA of these emails are not registered in the signature. Generally, these emails are regard as normal, so we assume these emails are classified into normal emails. We observe that f_1 measure is 0.9910 and error rate is 1.11%.

Aside from signature-based detection, anomaly detection has a possibility of classifying the 460 emails. For building an anomaly detection method, we transform the elements of MHEA to the weak hypotheses in AdaBoost, a typical machine learning algorithm. We perform supervised learning with the training dataset, and evaluate the classification accuracy with the testing dataset. The result shows that f_1 measure is 0.8858 and error rate is 10.82%.

We then design a hybrid detection method. The hybrid detection system classifies emails based on the signature first, and then performs anomaly detection on unclassifiable emails. We observe that f_1 measure is 0.9928 and error rate is 0.75%. In comparison to other detection methods, a hybrid detection method can achieve the best detection accuracy.

2 Mail Header and Encoding Anomaly

In this section, we summarize MHEA proposed in [1], and refine the generation process of MHEA.

2.1 Summary

According to [1], Arai distinguished between virus emails and normal emails by using Mail Header and Encoding Anomaly(MHEA). He analyzed some MIME header fields in email dataset, which was composed of 33,003 virus emails and 52,601 normal emails. According to the result, he suggested 5 variables can be used as heuristics that were not affected by variety of viruses. The variables are as follows.

V1. Irregular line

“Irregular line” is a binomial variable that indicates the email includes irregular line in Encoded-Words (EW). According to MIME [4], the attachment

files should be encoded to base64 format, called EW, when a Mail User Agent (MUA) sends binary data. MIME also defines that each line in the email is limited to 76 letters. If the length of EW is more than 76 letters, the MUA begins a new line to set in 76 letters for each line. Thus, EW should always be 76 letters except for the last line. However, Arai found some virus emails did not get along with MIME. Even if the line was not the last line, the number of letters in the line was not 76 in the cases of such emails. Arai regarded such lines as one of the footprints for detecting virus emails.

V2. EW length

“EW length” is an integer variable, which is calculated by dividing 1,000 into the length of EW. For example, EW length is 2 when EW has 2,076 letters. According to the analysis, Arai considered that the change in the file size is within 1,000 bytes even if the file size increased or decreased in virus’s mutation process.

V3. MIME multipart

“MIME multipart” is a string variable, which is combined **Content-type header** filed. For example, MIME multipart is `text/us-ascii,image/jpeg` when an email contains text written in English and one jpeg-format image file.

V4. Extension header

“Extension header” is a string variable, which is combined particular extensional MIME header such as **X-Priority**, **X-MSMail-Priority**, **X-Mailer header** field. Arai suggested that virus emails would be sent from particular MUAs.

V5. Encoded MIME boundary

The string variable, which is encoded from MIME boundary [5]. Arai attempted encoding by the following steps: (i) alphabets and numbers are converted to the letter “X”, and (ii) if the letter is the same as the letter before, the letter is removed. For example, when the given MIME boundary is “ABCDEFG==”, Step (i) converts from the MIME boundary to “XXXXXXXX==”, and Step (ii) converts from “XXXXXXXX==” to “X==”.

MHEA is generated by concatenating these 5 variables. In [1], Arai reported that (i) in the case of virus emails, the number of unique MHEA is much lower than that of normal emails, and (ii) MHEA that were derived from virus emails were different from those of normal emails. He mentioned that MHEA of virus emails have many similarities in comparison to those of normal emails, thus, he concluded that MHEA is capable to classify virus emails.

2.2 Refinement

We assumed that MHEA has a weak point in its generating process. Basically, EW length is affected by variety of viruses; it is natural to assume that the file size of a virus program changes 1KB or more in its mutation process. Thus, we consider that MHEA should be generated by concatenating 4 variables except from EW length.

Table 1. Frequency Distribution using V1, V2, V3, V4, and V5

Uniqueness of MHEA	Virus emails	Normal email
$\log(x) < 1$	207	1129
$1 \leq \log(x) < 2$	351	454
$2 \leq \log(x) < 3$	479	296
$3 \leq \log(x) < 4$	325	298
$4 \leq \log(x) < 5$	600	169
$5 \leq \log(x) < 6$	1207	162
$6 \leq \log(x)$	961	0
	4130	2508

Table 2. Frequency Distribution using V1, V3, V4, and V5

Uniqueness of MHEA	Virus emails	Normal emails
$\log(x) < 1$	50	886
$1 \leq \log(x) < 2$	106	453
$2 \leq \log(x) < 3$	116	406
$3 \leq \log(x) < 4$	130	226
$4 \leq \log(x) < 5$	462	375
$5 \leq \log(x) < 6$	1220	162
$6 \leq \log(x)$	2046	0
	4130	2508

To check my idea, we performed association analysis. First, we collected 4,130 of virus emails, which were sampled from our university email server from September 2006 to January 2007. In the email server, an anti-virus scans every email and these 4,130 emails were detected as virus. We also collected 2,508 emails sampled from our mailbox. There are 202,508 emails in the mailbox, and 2,508 emails have one or more attachment files.

Second, we generated MHEA for each email by concatenating 5 variables. We found 299 unique MHEA in the case of the virus emails and 1,179 in the case of the normal emails. We also observed that every MHEA of the virus emails is different from the MHEA of the normal emails.

According to [1], many virus emails had the same MHEA. For verification, we employ the uniqueness of MHEA as an index of similarity. We simply defined uniqueness x ; if x emails have the same MHEA, the uniqueness of the MHEA of these emails is x . To perform our association analysis, we investigated the frequency distribution as shown in Table 1. We temporarily determined the classes of frequency distribution as follows: Uniqueness is quite high when $\log(x) < 1$, high when $1 \leq \log(x) < 2$, slightly high when $2 \leq \log(x) < 3$, neutral when $3 \leq \log(x) < 4$, slightly low when $4 \leq \log(x) < 5$, low when $5 \leq \log(x) < 6$, quite low when $6 \leq \log(x)$. We observed that there is difference between the distribution of MHEA derived from virus emails and that of normal emails. Based on this condition, Cramer's coefficient of association(C) was 0.607. It indicated that classifying virus emails with MHEA is appropriate.

We also generated MHEA by concatenating 4 variables except from EW length. We then drew the frequency distribution as shown in Table 2 and observed that C was 0.761. In comparison to the case of using 5 variables, removing EW length led to the stronger correlation between uniqueness of MHEA and types of emails(virus or not). Thus, we concluded that EW length was affected by variety of viruses, so employed 4 variables for generating MHEA in the following section.

3 MHEA-Based Virus Email Detection Methods

Generally, detection methods are categorized into three types: signature-based detection, anomaly detection, and hybrid detection. Based on MHEA, we develop these 3 types of detection methods and evaluate the detection accuracy for each method. In our evaluation, we use f_1 measure (higher is better) and error rate (lower is better) as the indices of the detection accuracy.

3.1 Signature-Based Detection

First, we developed a signature-based detection method which can check the MHEA of the issued email with MHEA databases.

We generated the MHEA by using training dataset. Our dataset was composed of 4,130 virus emails and 2,580 normal emails. We chose 2,065 emails from virus emails by random sampling and also chose 1,258 emails from normal emails, and constructed a training dataset with these 3,319 emails. The rest of the emails were used for testing. The number of unique MHEA was 619, in which 70 types of MHEA were derived from virus emails and 549 types were derived from normal emails.

We also performed a classification experiment employing the MHEA as signature. In the classification test, we took one email from the testing dataset and generated its MHEA, and checked if the same MHEA existed in the signature dataset. If so, we checked whether the MHEA in the signature dataset was derived from virus emails or not. When the MHEA was derived from virus emails, we classified the email as virus. Conversely, we classified the issued email as normal when the MHEA was derived from normal emails.

The results were shown in Table 3. We classified 2,859 of 3,319 emails without error. The remaining 460 emails were unclassifiable because the MHEA of these emails were not listed on the signature. In the typical signature-based detection systems, these emails were regarded as normal emails. Based on this fact, we

Table 3. Signature-based Detection allowing unclassifiable email

	Actual virus emails	Actual normal emails	
Predict virus emails	2028	0	2028
Predict normal emails	0	831	831
Unclassifiable emails	37	460	460
	2065	1254	3319

Table 4. Signature-based Detection

	Actual virus emails	Actual normal emails	
Predict virus emails	2028	0	2028
Predict normal emails	37	1254	1291
	2065	1254	3319

Table 5. Anomaly Detection

	Actual virus emails	Actual normal emails	
Predict virus emails	1781	75	1956
Predict normal emails	284	1179	1363
	2065	1254	3319

regarded these unclassifiable emails as normal. The results were shown in Table 4. We observed that f_1 measure was 0.9910, error rate was 1.11%, TP rate was 98.21%, and FP rate was 0%.

3.2 Anomaly Detection

To categorize unclassified emails, we developed an anomaly detection based on MHEA. We employed AdaBoost to construct an anomaly detection algorithm. AdaBoost, which was proposed by Freund and Schapire, is the most typical boosting algorithm. AdaBoost solves many of the practical difficulties of the earlier boosting algorithms, and its ensembles perform better than the generic ensemble methods.

The weak hypotheses used in AdaBoost were transformed from V1, V3, V4, and V5 as mentioned in Section 2 and were shown as follows:

- H1. Check if the email contains irregular line in Encoded Word
- H2. Check if the uniqueness of MIME multipart is lower
- H3. Check if the uniqueness of Extension header is lower
- H4. Check if the uniqueness of Encoded MIME boundary is lower.

As an index for uniqueness, we checked frequency of appearance and compare it with defined discrimination threshold (θ). Imagine if an issued email's MIME multipart was observed in other 99 emails. In this case, 100 emails had the same MIME multipart, thus, frequency of appearance was 100. Because we decided $\theta = e^4 (\neq 54.60)$, frequency of appearance was larger than given θ ; It denoted that this email was deemed to be a virus email. Notice that this threshold was temporarily decided, so we discuss the effectiveness of θ in Section 4.

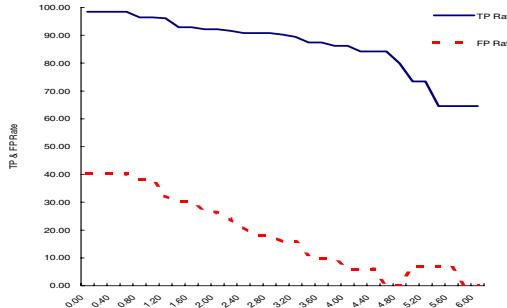
We evaluated the detection accuracy as shown in Table 5. By using training dataset, we performed supervised learning to construct the final hypothesis with the AdaBoost algorithm. We then classified emails in testing dataset with the final hypothesis. We observed that f_1 measure was 0.8858, error rate was 10.82%, TP rate was 86.25%, and FP rate was 5.98%. In comparison to a signature-based detection method, the accuracy of an anomaly detection method decreased.

3.3 Hybrid Detection

The key feature of our hybrid detection method was employing both signature-based detection and anomaly detection. In the context of virus email detection, a hybrid detection method first tried signature-based detection to check an email,

Table 6. Hybrid Detection

	Actual virus emails	Actual normal emails	
Predict virus emails	2060	20	2085
Predict normal emails	5	1234	1239
	2065	1254	3319

**Fig. 1.** Relationship between threshold and error rates

and then tried anomaly detection on the email only if the MHEA of the email was not listed on signature. In short, we applied the anomaly detection method to the unclassifiable emails for improving the detection accuracy.

The results were shown in Table 6. We observed that f_1 measure was 0.9928, error rate was 0.75%, TP rate was 99.76% and FP rate was 1.59%. In addition, 32 of 37 virus emails, which could not be detected by the signature-based detection, were detected as virus in the case of the hybrid detection method. By comparing the hybrid detection method with the signature-based detection method, f_1 measure was increased and error rate was decreased, although FP rate was increased. So, we assumed that the hybrid detection method can improve the detection accuracy.

4 Future Work

In this section, we discuss the index for measuring the uniqueness. Throughout the paper, it was difficult to decide whether the uniqueness was higher or lower, so we temporary determined the classes in Section 3 and the threshold in Section 3.2.

Figure 1 showed TP rates and FP rates by changing $\theta (= e^x)$ from e^0 to e^6 in increments x of 0.2, where the x axis denoted θ , and the y axis denoted the TP and FP rates in an anomaly detection method. A normal line denoted TP rate, and a broken line denoted FP rate. When given θ was 4.6, the TP rate was 84.21% and the FP rate was 0%. If we could employ 4.6 for θ for constructing a hybrid detection method, f_1 measure was 0.9985, error rate was 0.18%, TP rate was 99.71%, and FP rate was 0%.

However, it was difficult to determine θ in learning steps. In our future work, we need to explore a suitable way for adjusting θ . After a hybrid detection method classified an emails as a virus or not, a detection system would confirm whether the classification was correct or not, and should automatically investigate the suitable θ .

5 Conclusion

The main infection source of viruses is via email. Based on this fact, we developed virus email detection methods using Mail Header and Encoding Anomaly(MHEA).

We refined the generation process of MHEA by analyzing our dataset, which was composed of 4,130 virus emails and 2,058 normal emails, all of which have one or more attachment files. We found that using 4 variables, namely irregular line, MIME multipart, extension header, and encoded MIME boundary, performed better than using the original 5 variables.

In our preliminary evaluation, we observed that f_1 measure was 0.9910 and error rate was 1.11% in the case of the signature-based detection method. To decrease the unclassifiable emails, we employed the AdaBoost algorithm for constructing an anomaly detection method, and we also observed f_1 measure was 0.8858 and error rate was 10.82%. We finally constructed a hybrid detection method, which first tried signature-based detection to check an email, and then tried anomaly detection on the email only if the MHEA of the email was not listed on signature. We observed that f_1 measure was 0.9928 and error rate was 0.75%.

Acknowledgments

This work was a part of "Research and Development on Traceback Technologies in the Internet" sponsored by the National Institute of Information and Communications Technology(NICT).

References

1. Arai, T.: Computer virus detection method using mail form information. In: Symposium on Cryptography and Information Security (SCIS 2007) (2007)
2. Sophos Corporation: Top 10 viruses reported to sophos in 2005 (2005), <http://www.sophos.com/security/top-10/200512summary.html>
3. Information-technology Promotion Agency: Reporting status of computer virus - details for June 2008 (2008), <http://www.ipa.go.jp/security/english/virus/press/200806/documents/Virus0806.pdf>
4. Moore, K.: MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII text. RFC 2047, Internet Engineering Task Force (1996)
5. Freed, N., Borenstein, N.: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. RFC 2046, Internet Engineering Task Force (1996)

Faster Parameter Detection of Polymorphic Viral Code Using Hot List Strategy

Ruo Ando

National Institute of Information and Communication Technology,
4-2-1 Nukui-Kitamachi, Koganei,
Tokyo 184-8795 Japan
ruo@sfc.keio.ac.jp

Abstract. Polymorphic viral code with encrypted payload and obfuscated decipher routine is hard to detect by generic signature scan. In this paper we propose a faster parameter detection of polymorphic viral code using hot list strategy. Parameter detection is formulated as solving SAT problem using resolution and substitution by FoL (First order Logic) theorem prover. To make parameter detection faster, we discuss one of ATP (Automated Theorem Proving) strategies, called hot list. Experiment shows that with proper selection of hot list, we can make reasoning process faster with reduction rate of generated clauses from 60% to 80%.

1 Viral Polymorphism

Despite the short history, the techniques of writing malicious mobile code have become more sophisticated. Figure 1 shows three kinds of viral code. First, entry point is changed to point out viral code. Second, payload (viral part) is encrypted. Third, decipher routine is obfuscated. Besides encryption, polymorphic virus has obfuscated decipher routine, which makes it difficult to detect. Particularly, in the view of malware analysis (reverse engineering), parameter detection is important to cope with this kind of code. In this paper we propose a parameter detection technique using FoL (First order Logic) theorem prover and make it faster using a reasoning strategy called hot list.

2 Problem Formulation

Proposed method is divided into two steps: structure detection and parameter detection. Our detection method is SAT problem based, which is formulated between polymorphic code and the code pattern of viral code with decipher routine. In reasoning process, we have some code patterns of simplified polymorphic code. Let $f(x)g(y)$ be representation of polymorphic code. Also let $h(m,n)$ be representation of code pattern. First, our system apply resolution for $f(x)g(y)$ to $i(z)$. Then, the equality between $i(x)$ and $h(y)$ is evaluated.

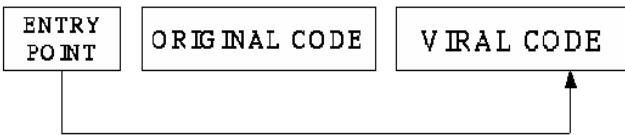
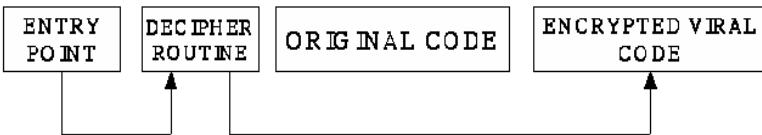
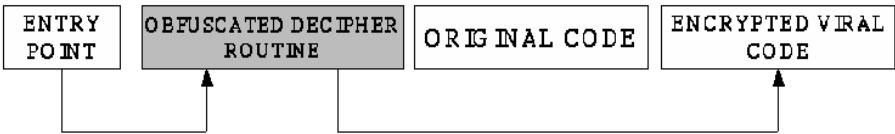
VIRAL CODE**VIRAL CODE WITH ENCRYPTION****POLYMORPHIC VIRAL CODE**

Fig. 1. Three kinds of viral code. Besides encryption, polymorphic code has obfuscated decipher routine.

$f(a,b)g(b,c) \rightarrow i(a,c)$: Resolution is applied.
 $i(a,c) = h(m,n)$: SAT is solved.

If $i(a,c)=h(m,n)$ is proved, it is meant that structure pattern (a decipher routine) is detected in the code. Then, our system is detecting parameters of the decipher routine. In parameter detection, equality substitution techniques called demodulation and substitution is applied. Let $i(x,y)$ be a pattern of decipher such as $(x \text{ XOR } y)$. SAT problem is represented on FoL reasoning process by Set of Support[1] and Unit conflict.

```

fact: c=X.
fact: i(a,c).
conclusion: i(X,c).
  
```

3 Parameter Detection and Hot List Strategy

3.1 Paramoudulation and Demodulation

In some cases of forensics, detecting parameters of decipher routine is required. We apply substitution techniques, paramodulation and demodulation[2]. Paramodulation is one of the techniques of equational reasoning. The purpose of this inference rule is to cause an equality substitution to occur from one clause to another. In the discussion of completeness and efficiency, paramodulation is often compared

with demodulation. Demodulation is designed for canonicalizing information and sometimes more effective than paramodulation. However, demodulation does not have power to cope with clauses as follows:

```
fact: f(g(x),x).
fact: equal(g(a),b).
conclusion f(b,a).
```

That is, paramodulation is a generalization of the substitution rule for equality. For searching parameters of obfuscated decipher routine, we need to use both paramodulation and demodulation.

```
fact: equal(data_16e,514Bh).
fact: mov(reg(ah),const(data_16e),63,time(1)).
conclusion : mov(reg(ah),const(514Bh),63,time(1)).
```

The clauses above is the application of demodulation to deal with constant number defined in the beginning of program. In obfuscating decipher routine, there's another way to hide parameter using mov instruction.

```
fact: mov(reg(ah),const(2Ch),
       162,time(1)).
fact: mov(reg(bx),reg(ah),300,time(1)).
fact: xor(reg(dx),reg(bx),431,time(1)).
/* decrypter */
```

In this case, we insert this clause to occur paramodulation.

```
-mov(reg(x),const(y),z,time(1)) | x=const(y,z).
conclusion:
decrypter(reg(dx),key(const(2Ch,162),431,time(1))).
```

Conclusion is deduced by paramodulation. By using paramodulation, we can detect the value of key, address of payload, loop counter (how many times the routine repeats), and entry point of decipher routine. Numerical output of theorem prover is shown in Table 5.

3.2 Hot List Strategy

In this paper we apply a heuristics to make paramodulation faster. Hot list strategy, proposed by Larry Wos[3], is one of the look ahead strategies. Look-ahead strategy is designed to enable reasoning program to draw conclusions quickly using a list whose elements are processed with each newly retained clause. Mainly, hot list strategy is used for controlling paramodulation. By using this strategy, we can emphasize particular clauses on hot list on paramodulation. In detail, readers are encouraged to read [3].

4 Experimental Result

In experiment, we generate four kinds of decipher routines as shown in Table 1 and 2. Our program generates random 300 executables with different forms.

Table 1. Three kinds of assembly code generated by SMEG

	Type A	Type B
loop I	set loop_start	loop_start
transfer I	mov data address	decrypt [address] key
decrypt I	decrypt data key	decrypt [address] key
transfer II	mov address data	decrypt [address] key
decrypt II	inc address	inc address
branch II	dec counter	dec counter
branch	test counter counter	test counter counter
loop II	jmp loop_start	jmp loop_start

Table 2. Three kinds of assembly code generated by SMEG

	Type C	Type D
loop I	set loop_start	set loop_start
transfer I	xchg address data	push data / pop data
decrypt I	decrypt data key	decrypt data key
transfer II	xchg address data	mov address data
decrypt II	inc address	inc address
branch II	dec counter	dec counter
branch	test counter counter	test counter counter
loop II	jmp loop_start	push / retf

Table 3. Hot list strategies for Type A. Paramodulation for detecting parameters into register E* is speeded up by hot list. We set 10 hot lists for each register and two groups.

Type A (no weighting)		Type A (with weighting)	
HOT LIST	all clauses	HOT LIST	all clauses
no heat	915	no heat	707
EAX	677	EAX	677
EBX	670	EBX	602
ECX	799	ECX	541
EDX	756	EDX	540
EDI	1078	EDI	822
ESI	1055	ESI	801
EBI	1055	EBI	801
EBP	1055	EBP	801
Group I	468	Group I	366
Group II	1510	Group II	1206

Table 4. Hot list strategies for Type B

Type B (no weighting)		Type B (with weighting)	
HOT LIST	all clauses	HOT LIST	all clauses
no heat	1592	no heat	769
EAX	915	EAX	605
EBX	1561	EBX	494
ECX	497	ECX	490
EDX	519	EDX	593
EDI	1921	EDI	1164
ESI	1724	ESI	843
EBI	1724	EBI	685
EBP	1724	EBP	685
Group I	463	Group I	242
Group II	2422	Group II	1807

Table 5. Hot list strategies for Type C

Type C (no weighting)		Type C (with weighting)	
HOT LIST	all clauses	HOT LIST	all clauses
no heat	976	no heat	604
EAX	1018	EAX	605
EBX	720	EBX	494
ECX	946	ECX	490
EDX	976	EDX	593
EDI	1592	EDI	1164
ESI	1272	ESI	843
EBI	1114	EBI	685
EBP	1114	EBP	685
Group I	738	Group I	463
Group II	2284	Group II	1807

Table 6. Hot list strategies for Type D

Type D (no weighting)		Type D (with weighting)	
HOT LIST	all clauses	HOT LIST	all clauses
no heat	1877	no heat	801
EAX	1444	EAX	587
EBX	1675	EBX	587
ECX	870	ECX	599
EDX	1877	EDX	737
EDI	7406	EDI	1462
ESI	2028	ESI	876
EBI	2028	EBI	876
EBP	2028	EBP	876
Group I	563	Group I	259
Group II	8186	Group II	1891

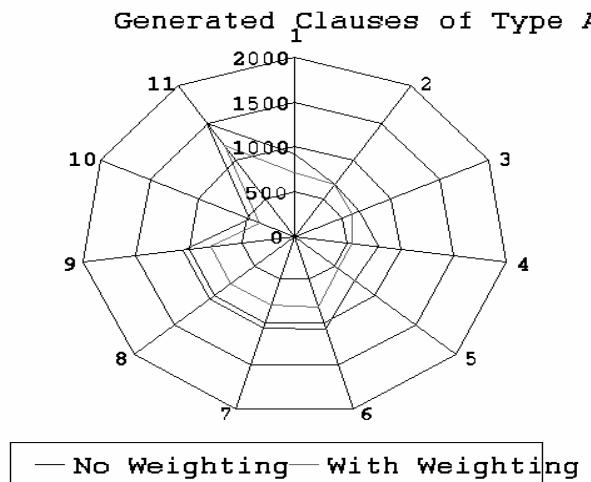


Fig. 2. The number of generated clauses of Type A according to Table 3. Line (group) 10 has the least number. Focus on register BX causes the fastest reasoning.

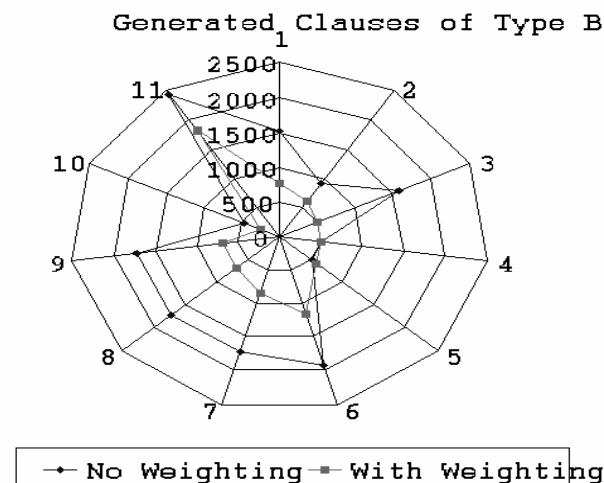


Fig. 3. The number of generated clauses of Type B according to Table 4. Line (group) 10 has the least number. Focus on register CX causes the fastest reasoning.

Type A and C uses mov and xchg (exchange) to transfer the encrypted data. Type B uses indirect addressing (xor [address] key) to execute payload transfer and decryption at the same time. In type D, stack operation is applied for data transfer (fetch) and loop II (push / reff). These four decipher routine are obfuscated by our program. Proposed system has been implemented on OTTER[4]

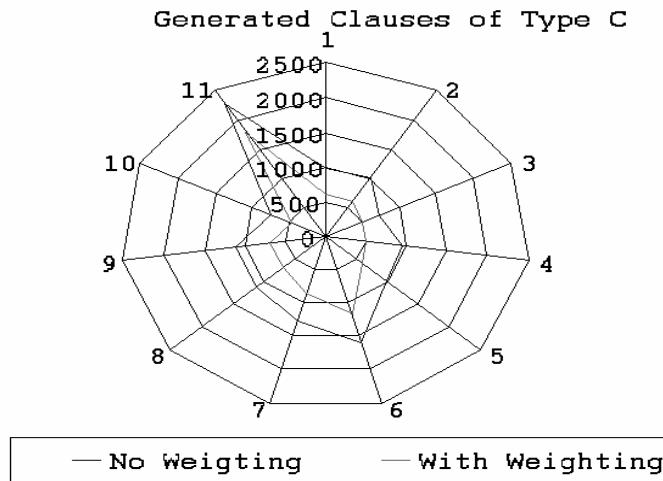


Fig. 4. The number of generated clauses of Type C according to Table 5. Line (group) 10 has the least number. Focus on register BX causes the fastest reasoning.

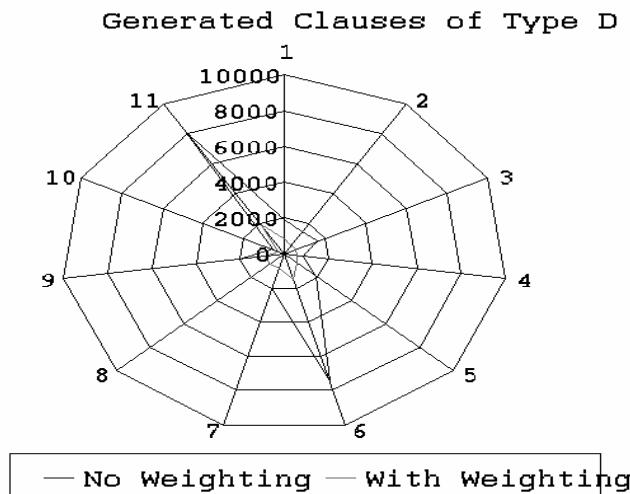


Fig. 5. The number of generated clauses of Type D according to Table 6. Line (group) 10 has the least number. Focus on register CX causes the fastest reasoning.

automated deduction system. To disassemble binary code, we are according to [5]. Figure 2,3,4 and 5 show graph representation of Table 3,4,5 and 6. Generally, hot list on calculation registers has better effect than ones of memory operation. Hot list has the best effect on Type D. we can make reasoning process faster with reduction rate of generated clauses from 60% to 80%.

Table 3, 4, 5 and 6 are result of applying hot lists for four types of SMEG generation. We make 10 hot list (list(hot)) according to eight registers and two groups {eax, ecx, ebx, edx} and {edi, esi, ebi, ebp}. Among 8 hot lists (eax, ecx, ebx, edx, edi, esi, ebi, ebp), which hot list increase performance best depends on types of generated code. As a whole, hot list group of calculation registers {eax, ecx, ebx, edx} results in good performance compared with group {edi, esi, ebi, ebp}. In some bad cases hot list of group II increase the number generated clauses compared with no hot list.

5 Conclusions

Polymorphic viral code with encrypted payload and obfuscated decipher routine is hard to detect by generic signature scan. In this paper we have proposed the faster parameter detection of polymorphic viral code using hot list strategy. Parameter detection is formulated as solving SAT problem using resolution and substitution by FoL (First order Logic) theorem prover. To make parameter detection faster, we have discussed the effectiveness of hot list strategy. Experiment shows that with proper selection of hot list, we can make reasoning process faster with reduction rate of generated clauses from 60% to 80%.

References

1. Wos, L., Robinson, G.A., Carson, D.F.: Efficiency and Completeness of the Set of Support Strategy in Theorem Proving. *Journal of Automated Reasoning* (1965)
2. Wos, L., Robinson, G.A., Carson, D.F., Shalla, L.: The Concept of Demodulation in Theorem Proving. *Journal of Automated Reasoning* (1967)
3. Wos, L., Pieper, G.W.: The Hot List Strategy. *Journal of Automated Reasoning* (1999)
4. OTTER automated deduction system, <http://www.mcs.anl.gov/AR/otter/>
5. Intel Corporation: IA-32 Intel® Architecture Software Developer's Manual, Volume 2A: Instruction Set Reference A-M (2004); vol. 2B: Instruction Set Reference N-Z (2004)

G-Means: A Clustering Algorithm for Intrusion Detection*

Zhonghua Zhao¹, Shangqing Guo^{1,***}, Qiliang Xu^{1,***}, and Tao Ban²

¹ School of Computer Science and Technology, Shandong University, Jinan, 250101, China

² Information Security Research Center, National Institute of Information and Communications Technology, Tokyo, 184-8795, Japan

{guoshangqing, xql}@sdu.edu.cn

Abstract. Coupled with the explosion of number of the network-oriented applications, intrusion detection as an increasingly popular area is attracting more and more research efforts, especially in anomaly intrusion detection area. Literature shows clustering techniques, like K -means, are very useful methods for the intrusion detection but suffer several major shortcomings, for example the value of K of K -means is particularly unknown, which has great influence on detection ability. In this paper, a heuristic clustering algorithm called G-means is presented for intrusion detection, which is based on density-based clustering and K -means and overcomes the shortcomings of K -means. The results of experiments show that G-means is an effective method for the intrusion detection with the high Detection Rate and the low False Positive Rate, as it can reveal the number of clusters in the dataset and initialize reasonably the cluster centroids, which makes G-means accelerate the convergence and obtain preferable performance than K -means.

1 Introduction

Intrusion detection system (IDS) is a key component of secure information systems. The goal of IDS is "to identify a set of intrusions that attempt to compromise the integrity, confidentiality or availability of a resource" [1]. Intrusion detection techniques generally fall into two categories: misuse detection and anomaly detection [2]. Misuse detection techniques recognize the signatures of known attacks, they match the observed behaviors with those known signatures and signal intrusions when there are matches. Misuse detection techniques are efficient and accurate in detecting known intrusions but cannot detect novel intrusions whose signature patterns are unknown; Anomaly detection techniques establish a profile of the subject's normal behaviors (norm profile), compare the observed behaviors of the subject with the norm profile and signal intrusions when the subject's observed behaviors differ significantly from the norm profile. Anomaly detection techniques can detect both known and unknown attacks.

* Foundation item: Supported by the Natural Science Foundation of Shandong Province (Y2007G37) and the Science and Technology Development Program of Shandong Province (2007GG10001012).

** Corresponding author.

In anomaly detection, clustering can be considered as one of the most important unsupervised learning algorithms and has been used to solve the problems of anomaly intrusion detection. The process of clustering is to organize instances into groups so that the members from the same cluster are quite similar, and the members from different clusters are quite different from each other. For clustering algorithm, K -means [3] is one of the simplest and most important unsupervised learning algorithms and partitions a set of data into K clusters. However, K -means suffers several major weaknesses which have great influence on detection ability for intrusion detection:

- 1) The results depend on the value of K , that is to say the value of K is very critical to the clustering result, but the ascertain of K is particularly troublesome. Obtaining the optimal K for a given data set is an NP-hard problem [8].
- 2) The way to initialize K clusters is not specified, which frequently results in suboptimal partitions.
- 3) The clustering may end with some empty clusters, and the category of the empty clusters is meaningless for the classification.

In this paper, we present G-means, a clustering algorithm for intrusion detection which is based on density-based clustering [5] and K -means clustering [3], and G-means overcomes the shortcomings of K -means.

2 Related Work

Leonid Portnoy [4] introduced a clustering algorithm to detect both known and new intrusion types without the need to label the training data. A simple variant of single-linkage clustering to separate intrusion instances from the normal instances was used. The algorithm starts with an empty set of clusters, and updates it as it proceeds. For each new instance, the algorithm computes the distances to the centroids of the clusters in the cluster set so far. The cluster with the shortest distance is selected, if the shortest distance is less than predefined constant W (cluster width), and then the instance is assigned to the cluster, or else a new cluster is created with the instance as the center. Each cluster center is then updated to the mean of the cluster members. Repeat the process above until all the instances have been assigned to the clusters. Though this algorithm overcomes the shortcoming of number of clusters dependency, it requires a predefined parameter of clustering width W which is not always easy to find. The assumption that "the normal instances constitute an overwhelmingly large portion (>98%)" is also too strong.

In [7], Qiang Wang introduced Fuzzy-Connectedness Clustering (FCC) for intrusion detection based on the concept of fuzzy connectedness which was introduced by Rosenfeld in 1979. FCC can detect the known intrusion types and their variants, but it is difficult to find a general accepted definition of fuzzy affinity which was used by FCC.

3 Methodology

G-means clustering algorithm can be divided into two phases: training phase to build clusters and detecting phase to detect intrusions. The main idea of training phase: firstly we run a density-based clustering procedure to produce the cluster-ordering of

the input dataset and the information which would be used to assign cluster memberships; Secondly, we extract the number and member of kernels from the density-based cluster-ordering, and the kernel is a part of every cluster; Finally, we run K -means clustering with extracted information by previous step. In detecting phase, we perform detection of intrusions.

Note that the density-based clustering is used as a preprocessing step for G-means but not suitable as a stand-alone tool for intrusion detection, because the density-based clustering produces the suboptimal partitions of the dataset and puts lots of instances into noises which are dropped out, which makes the Detection Rate and the False Positive Rate worse. The correlative data are provided in the posterior comparing experiment.

In G-means, there are three main parameters: the radius ε , the minimum number of instances $Minpts$ and extraction distance ε' . The radius ε determines the given radius of the neighborhood for a cluster. $Minpts$ is the minimum number of instances in the neighborhood of the radius ε . The extraction distance ε' is the maximum radius of the extracted clusters. Assume we have fixed ε , $Minpts$ and ε' . The details of the algorithm are listed as follows:

3.1 Training Phase

Step 1. Create the cluster-ordering of the input dataset and the information used to assign cluster memberships. The information contains the core-distance and a suitable reachability-distance for each instance, according to ε and $Minpts$. The algorithm is shown as follows:

- 1) Obtain an instance from the training dataset. Initialize its reachability-distance and core-distance, and then mark the instance with its clustering order.
- 2) If the instance is a core instance, gather all the instances which are directly density-reachable from the core instance into a set and order the instances in the set with their reachability-distance to the closest core instance from which they have been directly density reachable. Otherwise execute the step 5).
- 3) Allocate the instance having the smallest reachability-distance in the set as the current instance, calculate its core-distance, and then mark it with its clustering order. If the current instance is a core instance, expand the set with all the instances directly density-reachable from the current instance.
- 4) Repeat the step 3), until all the instances of the set are processed.
- 5) Repeat steps 1) and 2), until no instances are left in the training dataset.

Step 2. Obtain the cluster number K and the members of every kernel from the cluster-ordering with respect to $Minpts$ and the clustering-distance ε' by simply "scanning" the cluster-ordering. The procedure which performs this task is depicted as follows:

- 1) Obtain an instance from cluster-ordering of the dataset as the current instance. If the reachability-distance of the current instance is larger than the clustering-distance ε' , execute the step 2). Otherwise, jump to the step 3).

- 2) If the core-distance of the current instance is smaller than ε' , create a new cluster. Otherwise, assign the current instance to NOISE. Jump to the step 4).
- 3) Assign this instance to the current cluster.
- 4) Repeat the step 1), until no instances are left in the cluster-ordering.

Step 3. After obtaining the clustering information, the procedure continues as follows:

- 1) Calculate the centroids of K clusters with the members of every kernel obtained in Step2).
- 2) Assign each instance to the cluster that has the closest centroid from the instance. When all instances have been assigned, recalculate the centroids of K clusters.
- 3) Repeat the step 2) until the centroids no longer move.

3.2 Detecting Phase

In detecting phase, G-means perform detection of intrusions. Given an instance d , the algorithm proceeds as follows:

- 1) Normalize d with the normalization method based on the statistical information of the training set. Let d' be the instance after normalization.
- 2) Discover the cluster with the shortest distance to d' by calculating Euclidean distance.
- 3) Classify d' by the category of the closest cluster.
- 4) Classify d into the same category with d' .

3.3 Algorithm Analysis

The time complexity of creating the cluster-ordering in Step1 is $O(n^2)$, n is the number of instances in the training dataset, that of obtaining the cluster number K and the members of every kernel in Step2 is $O(n)$, that of obtain clustering information in Step3 is $O(n * K * t)$, t is the iterative time, K and t are usually much less than n , and that of detecting phase is $O(n * K)$. If we run creating the cluster-ordering in Step1 on top of a tree-based spatial index [9], however, the run-time of creating the cluster-ordering in Step1 is reduced to $O(n * \log n)$. Furthermore, if we have a direct access to the ε -neighborhood, e.g. if the instances are organized in a grid, the run-time of creating the cluster-ordering in Step1 is further reduced to $O(n)$ because in a grid the complexity of a single neighborhood query is $O(1)$ [5].

4 Experimental Results

4.1 Dataset Preparation

The network connection records used in the experiment are the KDD CUP 1999 Dataset from MIT's Lincoln Lab, which has been developed for IDS evaluation by DARPA. It consists of approximately 4,900,000 data instances, each of which is a vector of extracted feature values from a connection record obtained from the raw network data gathered during the simulated intrusions and a vector consisting of 41 various quantitative and qualitative features [6].

Four different categories of attack patterns are included:

- a) Denial of Services attack (DOS). Examples are Apache2, Land, Mail bomb, SYN Flood, Ping of death, Process table, Smurf and Syslogd.
- b) User to Super user or Root Attacks (U2R). Examples are Eject, Ffbconfig, Loadmodule, Perl, Ps and Xterm.
- c) Remote to User Attack (R2L). Examples are Dictionary, Ftp_write, Gest, Imap, Named, Phf, Sendmail , Xlock and Xsnoop.
- d) Probing: Probing is a class of attacks in which an attack scans a network of computers to gather information or find known vulnerabilities. Examples are Lpseep, Mscan, Nmap, Saint, Satan.

In the G-means algorithm, we use the Euclidean distance as the metric to calculate the distance between the instances and clusters. The Euclidean distance can only transact features with continuous values, so all the symbolic features need removing. We also normalize the data and make the values of every feature have the same range of [0, 1] to avoid the dominance of any feature in the distance calculation. The method of normalization is listed as follows [7]:

$$\text{new_instance}[i] = \begin{cases} \frac{\text{old_instance}[i] - \min(\text{feature}_i)}{\max(\text{feature}_i) - \min(\text{feature}_i)}, & \max(\text{feature}_i) \neq \min(\text{feature}_i) \\ 1, & \max(\text{feature}_i) = \min(\text{feature}_i) \end{cases}$$

4.2 Performance Measures

Several indicators of performance are used to evaluate G-means algorithm: the Detection Rate (DR), the False Positive Rate (FPR), and the Detection Rate of each major category intrusions.

The Detection Rate (DR) is defined as the number of intrusion instances correctly classified as one of major categories divided by the total number of intrusion instances present in the test set.

The False Positive Rate (FPR) is defined as the number of normal instances that are (incorrectly) classified as intrusions divided by the total number of normal instances in the test set.

The Detection Rate (DR) of each major category intrusions is defined as the number of intrusion instances which belong to the major category correctly classified as the homologous major category divided by the total number of the homologous major category instances in the test set.

4.3 Performance Analysis

In G-means, the values of three main parameters need fixing firstly. The three parameters are the radius ε , the minimum number of instances $Minpts$ and extraction distance ε' . We use a subset of the KDD data on the scale of 40,000 to run a series of tests with different values of these three variables, measuring the resulting performance. Through the parameters estimation experiment, we finally decide to choose 5 as the value for ε , 1 as the value for ε' and 8 as the value for $Minpts$. In contrast with

other clustering methods, we find that G-means is rather insensitive to the generating distance ε and the value for $Minpts$. Roughly speaking, the values have just to be "large" enough to yield a good result.

After all parameters are specified, we evaluate G-means by using five different subsets of the KDD data on different scales and its size is approximately 40,000 and the most size is approximately 200,000. The results of the experiments are shown in Table 1, 2.

Table 1. Detection Rate (DR) and False Positive Rate (FPR)

scale of the dataset	cluster number	DR	FPR
40,000	167	98.6741%	2.6230%
80,000	453	99.2843%	1.8159%
120,000	697	99.2074%	1.6259%
160,000	700	99.3282%	1.4206%
200,000	745	99.3189%	2.1575%

Table 2. Detection Rate (DR) of each major category intrusions

Scale of the dataset	cluster number	DOS	U2R	Probing	R2L
40,000	167	99.3782%	34.6154%	95.0085%	86.4121%
80,000	453	99.5097%	38.4615%	95.8364%	89.7869%
120,000	697	99.3845%	38.4615%	94.0833%	88.7211%
160,000	700	99.4450%	38.4615%	95.2764%	89.0764%
200,000	745	99.4389%	42.3077%	94.6433%	85.6128%

From the experiments, we may find that G-means has a higher DR of more than 98% and a lower FPR of about 2%. The DR of U2R category is approximately 40%. After the deep analysis, we find that the main cause is that the number of U2R instances in the training dataset is only 52 so that it is difficult for clustering of U2R. If more U2R instances are provided, we believe that the DR of U2R will also be high. From the experiments, we also find that the number of the empty clusters resulted in G-means is a little, so that we need not deal with them, e.g. the number of empty clusters on the scale of 200,000 is about 6. The reason with good performance for G-means is that the number of clusters is automatically set to a reasonable value and the initial cluster centroids of G-means are also set reasonably for getting perfect partitions.

4.4 Comparing with Density-Based Clustering and K-Means

We know that G-means is based on density-based clustering OPTICS [5] and K -means clustering [3]. In order to exhibit that G-means has the better performance than them, we separately run the three algorithms with the same training/testing subsets on the scale of 100,000. K -means is experimented on with different K values. The results of comparison are shown in Table 3, 4.

Table 3. The comparison in Detection Rate (DR) and False Positive Rate (FPR)

	G-means		<i>K</i> -means		OPTICS
clusters	621	5	23	300	600
DR	99.1246%	96.2579%	97.5292%	98.9820%	99.1953%
FPR	1.4107%	0.3264%	3.0951%	0.9001%	1.5054%
				700	621
				99.1166%	94.6456%
				99.1166%	94.6456%
				24.5815%	

Table 4. The comparison in Detection Rate (DR) of each major category intrusions

	G-means		<i>K</i> -means		OPTICS
clusters	621	5	23	300	600
DOS	99.5106%	99.0738%	99.3001%	99.3395%	99.6895%
U2R	38.4615%	0.0000%	0.0000%	0.0000%	38.4615%
Probing	95.2520%	0.0000%	43.1702%	94.9111%	94.8624%
R2L	85.3464%	0.0000%	1.3321%	81.8828%	84.4583%
				95.3007%	54.3462%
				85.7016%	57.1048%

Experimental results show that G-means directly obtains the preferable performance, but *K*-means usually needs multiple runs with different *K* values and chooses the best one, and therefore *K*-means is difficult to apply in practice directly.

Table4 shows that G-means is better than OPTICS. The DR of G-means is 99.1246% and is bigger than 94.6456% of OPTICS, at the same time, the FPR of G-means is only 1.4107% and that of OPTICS is 24.5815% which is worse. After the deep analysis, we find that the main cause of the difference is that OPTICS produces the suboptimal partitions of the dataset and puts lots of instances into noises which are dropped out.

5 Conclusions

In this paper, we present a clustering heuristic algorithm for intrusion detection, called G-means, which is based on density-based clustering and *K*-means. It overcomes the existed shortcomings of *K*-means and density-based clustering in the intrusion detection, accelerates convergence and obtains preferable performance than *K*-means by initializing the centroids of all the clusters in reason with the clustering information from the previous clustering procedure. The results of experiments show that G-means is an effective method for the intrusion detection with the high Detection Rate and the low False Positive Rate.

References

1. Denning, D.E.: An Intrusion-Detection Model. IEEE Transactions on Software Engineering 13(2), 222–232 (1987)
2. Wenke, L., Stolfo, S.J., Mok, K.W.: A Data Mining Framework for Building Intrusion Detection Models. In: The 1999 IEEE Symposium on Security and Privacy, pp. 120–132. IEEE Press, Oakland (1999)

3. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press, Berkeley (1967)
4. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion Detection with Unlabeled Data Using Clustering. In: ACM CSS Workshop on Data Mining Applied to Security, pp. 5–8. ACM Press, Philadelphia (2001)
5. Mihael, A., Markus, M.B., Hans-Peter, K., Jörg, S.: OPTICS: Ordering Points to Identify the Clustering Structure. In: ACM SIGMOD 1999 International Conference on Management of Data, pp. 49–60. ACM Press, Philadelphia (1999)
6. KDD Cup 1999 Data. University of California, Irvine,
<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
7. Qiang, W., Vasileios, M.: A Clustering Algorithm for Intrusion Detection. In: The SPIE Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, Florida, vol. 5812, pp. 31–38 (2005)
8. Hansen, P., Mladenovic, N.: J-means: A New Local Search Heuristic for Minimum Sum of Squares Clustering. Pattern Recognition, Biol. 34, 405–413 (2001)
9. Stefan, B., Daniel, A.K., Hans-Peter, K.: The X-Tree: An Index Structure for High-Dimensional Data. In: 22nd VLDB Conference, pp. 28–39. Morgan Kaufmann Publishers Inc., Mumbai (1996)

Anomaly Intrusion Detection for Evolving Data Stream Based on Semi-supervised Learning

Yan Yu^{1,*}, Shanqing Guo^{2,*}, Shaohua Lan¹, and Tao Ban³

¹ School of Computer Science and Technology,
Nanjing University of Science and Technology, 210094 Nanjing, P.R. China
{yuyan, lansh}@mail.njust.edu.cn

² School of Computer Science and Technology, Shandong University,
250101 Jinan, P.R. China
guoshanqing@sdu.edu.cn

³ Information Security Research Center, National Institute of Information and
Communications Technology, 184-8795 Tokyo, Japan
bantao@nict.go.jp

Abstract. In network environment, time-varying traffic patterns make the detection model not characterize the current traffic accurately. At the same time, the deficiency of training samples also degrades the detection accuracy. This paper proposes an anomaly detection algorithm for evolving data stream based on semi-supervised learning. The algorithm uses data stream model with attenuation to solve the problem of the change of traffic patterns, as while as extended labeled dataset generated from semi-supervised learning is used to train detection model. The experimental results manifest that the algorithm have better accuracy than those based on all historical data equivalently by forgetting historical data gracefully, as while as be suitable for the situation of deficiency of labeled data.

1 Introduction

Along with the development of computing and network technology, there has been a dramatic growth of interest in network security. In particular, Intrusion Detection Systems (IDS) have become important tools for ensuring network security. Intrusion detection is based on the assumption that intrusion activities are noticeably different from normal system activities and thus detectable. Earlier studies have utilized a rule-based approach for intrusion detection, but can not find the novel or unknown attacks[1,2]. As a result, various data mining techniques are used to detect attacks[3,4]. But there exist still some drawbacks, such as lack of labeled data, time-varying traffic patterns, low detection accuracy.

Traditional anomaly detection algorithms require a set of labeled data to train their intrusion detection models. But in network scenarios, there are a large number of unlabeled data but insufficient of labeled data since it will be expensive to generate.

* Corresponding author.

Semi-supervised learning[5,6] combines labeled and unlabeled data during training to improve performance. In semi-supervised clustering, some labeled data are used with unlabeled data to obtain better clustering. It is known that applying semi-supervised learning to anomaly detection can improve the detection accuracy.

The consecutive arrival of network traffic demands the detection model having well scalability. Besides, time-varying traffic patterns demand that the detection model should adapt to the change of those patterns. But previous attempts to build intrusion detection systems have been based on mining static, previously stored network traffic[3]. Such approaches are not suitable for the temporal nature of network traffic, which is a data stream. A data stream is a continuum of potentially infinite data elements, in an ordered sequence of arrival. So, a data stream has necessarily a temporal dimension, and the underlying process that generates the data stream can change over time[7,8].

In the light, we introduce an data stream model and semi-supervised learning into intrusion detection area. In this paper, we propose an anomaly detection algorithm for evolving data stream based on semi-supervised learning, SSAD. SSAD algorithm utilizes attenuation rule to decrease the effect of historical data on detection result, which can help the algorithm to learn from current data that characterize the traffic pattern more accurately. SSAD also use semi-supervised learning to extend labeled dataset as training dataset to do with the problem of lack of labeled data. The experimental result show that the algorithm not only improves the detection accuracy, but also be suitable for the situation of deficiency of labeled data.

The paper is organized as follows: section 2 introduces our semi-supervised learning approach to generate extended labeled dataset. Section 3 describes the data stream model for network traffic. Section 4 describes our SSAD algorithm. In section 5 we present the experimental results and analysis of SSAD algorithm in intrusion detection. The paper concludes with section 6.

2 Drawbacks of Existing Methods and Our Solutions

Existing intrusion detection approaches face with two problems mainly. Firstly, the traffic patterns change over time. Consequently, the detection model should characterize the nature of traffic time-varying. In other words, data in different stage should have different influence on the detection model. Secondly, it is well known that it is extremely difficult to obtain a large number of labeled data as the training dataset of intrusion model in network scenarios, which would degrade the detection accuracy heavily. Therefore, we will analyze the problems and propose our solutions.

2.1 Influences of Data in Different Stages and Our Solution

A data stream can be viewed as an infinite process consisting of data which evolve with time. Among of all data, current data must have more influence than historical data at the respect of detecting attacks. Therefore, intrusion detection algorithms based on data stream model should process network data discriminatively according to their arriving time. The influence of historical data should fade out as time goes on.

At the same time, storing all historical data is unrealistic. So, we use history information vector to represent history information which is the extension of [9] on time dimension. Several related concepts are formally defined as follows:

Definition 1. (History Information Vector): History information vector in anomaly intrusion detection is defined as a clustering information set $\{\vec{HI}_p\}_{p=1}^k$, which represents d -dimensional set of datapoints $\{\dots, \vec{X}_{i-1}, \vec{X}_i, \vec{X}_{i+1}, \dots\}$, where \vec{HI}_p denotes $(2d+1)$ tuple of the p -th cluster in history information. Namely, given d -dimensional datapoint set $\{\vec{X}_i | 0 < i \leq n\}$ in a cluster, there is $\vec{HI}_p = (\vec{S}_p, \vec{D}_p, n_p)$, where

$$\begin{aligned}\vec{S}_p &= \left[\sum_{i=1}^n x_i^{(1)}, \sum_{i=1}^n x_i^{(2)}, \dots, \sum_{i=1}^n x_i^{(d)} \right] \\ \vec{D}_p &= \left[\sum_{i=1}^n (x_i^{(1)})^2, \sum_{i=1}^n (x_i^{(2)})^2, \dots, \sum_{i=1}^n (x_i^{(d)})^2 \right]\end{aligned}\quad (1)$$

where n_p is the number of datapoints in cluster p .

Definition 2. (Information Attenuation): The influence of information on the result of intrusion detection will be fading out with time. This procedure is called information attenuation. Let $w_k^{(0)}$ denote the weight of cluster k in chunk, and α is attenuation coefficient. Then the weight will be changed to $w_k^{(t)} = e^{-\alpha t} w_k^{(0)}$ after period t , where α can be calculated by half-life ρ as follows: $e^{-\alpha\rho} = \frac{1}{2}$.

Attenuation Rule. Let $n_k^{(i)}$ denote the number of datapoints in cluster k which appear in the chunk at time t , and $c_k^{(t_0+t)}$ denotes one after period t . The attenuation rule can be defined as

$$\begin{cases} c_k^{(t_0)} = w_k^{(0)} n_k^{(t_0)} \\ c_k^{(t_0+t)} = e^{-\alpha} c_k^{(t_0+t-1)} + w_k^{(0)} n_k^{(t_0+t)} \end{cases}\quad (2)$$

Proof

Let t_0 be the initial time of algorithm. After every specific period, a data chunk will be generated. It is known that $w_k^{(t)} = e^{-\alpha t} w_k^{(0)}$ and $c_k^{(t_0)} = w_k^{(0)} n_k^{(t_0)}$. Then we have

$$\begin{aligned}c_k^{(t_0+t)} &= e^{-\alpha} c_k^{(t_0+t-1)} + w_k^{(0)} n_k^{(t_0+t)} \\ &= e^{-\alpha} (e^{-\alpha} c_k^{(t_0+t-2)} + w_k^{(0)} n_k^{(t_0+t-1)}) + w_k^{(0)} n_k^{(t_0+t)} \\ &= e^{-\alpha} (e^{-\alpha} (e^{-\alpha} c_k^{(t_0+t-3)} + w_k^{(0)} n_k^{(t_0+t-2)}) + w_k^{(0)} n_k^{(t_0+t-1)}) + w_k^{(0)} n_k^{(t_0+t)} \\ &= \dots \\ &= w_k^{(t)} n_k^{(t_0)} + w_k^{(t-1)} n_k^{(t_0+1)} + \dots + w_k^{(0)} n_k^{(t_0+t)}\end{aligned}$$

According to the definition of information attenuation, the datapoint number of cluster k after period t should be

$$c_k^{(t_0+t)} = w_k^{(t)} n_k^{(t_0)} + w_k^{(t-1)} n_k^{(t_0+1)} + \dots + w_k^{(0)} n_k^{(t_0+t)}$$

Attenuation rule is tenable.

2.2 Difficulty of Obtaining Large Number of Labeled Data and Our Solution

In network environment, it is extremely difficult to obtain a large number of labeled data. The deficiency of training samples degrades the detection accuracy. If we can extend the labeled data set to train detection model, we should build an accurate one. In clustering, an unlabeled dataset is partitioned into groups of similar examples, typically by optimizing an objective function. So, we consider to extend the labeled dataset using semi-supervised clustering to obtain more training samples.

There are both labeled and unlabeled data in original dataset. We can partition the data into different clusters by k-means algorithm. In k-means algorithm, objects in the same partition have high similarity. So, we can say that closer the distance between object and cluster center is, higher probability they have the same distribution with.

Therefore, we can extend labeled dataset by labeling some unlabeled data which is near to the labeled cluster center, according to the confidence $c\%$. The more the data are labeled, the more the training samples are. Accordingly, the noise introduced by labeling wrongly is more. So, the choice of confidence should balance between both of them. By this way, it is possible to obtain sufficient labeled data as training samples, which may improve the accuracy of learned detection model.

The concept of Extended Labeled Dataset is defined as following:

Definition 3. (Extended Labeled Dataset): Let $x \in S$ denote network data where $S = D_l \cup D_u$, $D_l = \{x_j, l | j = 1, \dots, n\}$ denotes labeled dataset, $D_u = \{x_j | j = n+1, \dots, n+m\}$ denotes unlabeled dataset with $m = |D_u| >> n$. Extending the label information of data in D_l to those in D_u according to the result of semi-supervised clustering. Then labeling the closest data to labeled cluster center based on confidence $c\%$, a new labeled dataset can be generated. The generated dataset is called extended labeled dataset on S , denoted by ED_l .

3 SSAD Algorithm

In SSAD algorithm, network data stream will be divided into chunks based on their arriving time. To every chunk of data and history information which has been process by attenuation, algorithm will apply semi-supervised clustering to extend labeled dataset which will be used as training dataset later.

Furthermore, SVM[10] has fast processing speed and good scalability, which have been proved to be suitable for intrusion detection[11]. Therefore, SVM algorithm is selected as classifier to build anomaly intrusion detection model. The main procedure of SSAD algorithm is described as follows.

Algorithm. SSAD

Input: data stream $X = \{D^{(t)}|t = 1, 2, \dots\}$, where $D^{(t)} = D_l^{(t)} \cup D_u^{(t)}$ for t as the chunk sequence number, in which $D_l^{(t)} = \{(x_1^{(t)}, l_1), \dots, (x_n^{(t)}, l_n)\}$ is labeled dataset, $D_u^{(t)} = \{x_{n+1}^{(t)}, \dots, x_{n+m}^{(t)}\}$ is unlabeled dataset, confidence $c\%$, and half-life ρ .

Output: Intrusion detection model $N^{(t)}(x)$.

1. Initialize data structures, $t = 0$, $w_k^{(0)} = 1$;
2. Cluster labeled dataset $D_l^{(t)}$, if $x_i, x_j \in S_h \wedge S_h \subseteq D_l^{(t)}$, then $l_i = l_j$;
3. Cluster iteratively until convergence:
 - 3a. As to unlabeled dataset $D_u^{(t)}$, assign its datapoint to closest cluster;
 - 3b. As to history information vector $\{\overrightarrow{HI}_p\}_{p=1}^k$, assign it to closest cluster, distance can be calculated as $dist(\overrightarrow{HI}_p, \mu_h) = \sqrt{\frac{1}{n_p} [n_p \mu_h^2 + \sum_{i=1}^{n_p} x_i^2 - 2\mu_h (\sum_{i=1}^{n_p} x_i)]}$;
4. Label clusters as $label_i$ with $0 \leq i \leq \gamma$, if labeled data exist according the class of labeled data, where γ is known as the class number of labeled dataset $D_l^{(t)}$;
5. Determine the classe ξ_h of clusters, where

$$\xi_h = \begin{cases} label_j, & \text{if } |label_i| \prec |label_j|, 1 \leq i \wedge j \leq \gamma \wedge i \neq j \\ new_attack, & S_h \cap D_l = \emptyset \end{cases}$$
 then distributing cluster class ξ_h to every member in the cluster;
6. Select $c\%$ data which are closest to their cluster center in each cluster, and put them into extended labeled dataset $ED_l^{(t)}$, where $c\%$ is the confidence;
7. Train SVM by $ED_l^{(t)}$, then generate intrusion detection model $N^{(t)}(x)$;
8. Calculate history information vector of current chunk as $\{\overrightarrow{HI}_p\}_{p=1}^k$ and its attenuation as $c_h^{(t+1)} \leftarrow e^{-\alpha} c_h^{(t)} + w_h^{(0)} |S_h^{(t)}|$;
9. $t = t + 1$, then goto step 2.

4 Experimental Results and Analysis

4.1 Dataset and Performance Measures

In order to evaluate the performance of SSAD algorithm, the dataset is generated randomly from a dataset in KDD[12], which contains 492,000 records having 41 features each. This dataset has five different classes including Normal traffic, DoS, Probe, R2L and U2R. In the experiment, small labeled data and plenty of unlabeled data are sampled randomly. All models are trained and tested with the same dataset.

4.2 Performance Measures

The effect of extended labeled dataset on intrusion detection is evaluated by precision, recall and F-value[13], which may be defined as follows: precision is defined as the number of correctly detected attacks divided by total number of true attacks. recall is

defined as the number of correctly detected attacks divided by total number of attacks. And F-value is defined as $(2 \times \text{recall} \times \text{precision}) / (\text{precision} + \text{recall})$.

The accuracy of SSAD algorithm is evaluated by detection rate(DR) and false positive rate(FPR), which may be defined as follows: DR is definded as the number of correctly detected attacks divided by total number of true attacks. And FPR is defined at the number of false alarms divided by total number of normal samples.

4.3 Experiment and Analysis on SSAD Algorithm

Firstly, we evaluate whether the extended labeled dataset in SSAD will improve the accuracy of intrusion detection. In the experiment, we use only the extended labeled dataset, but not data stream model and history information attenuation, to evaluate the detection accuracy. SVM algorithm is used as classifier. Here, we run SVM with extended labeled dataset, abbreviated as SVMw, and SVM without extended labeled dataset, abbreviated as SVMw/o, ten times on the same training and test dataset, respectively. Then the average results are calculated. Fig.1 reports the comparison of F-value between SVMw and SVMw/o.

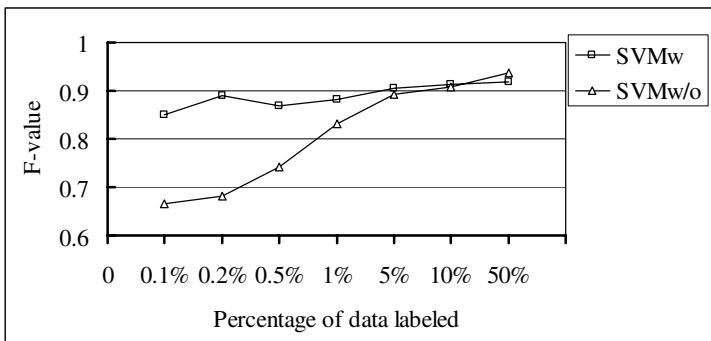


Fig. 1. Comparison between SVMw and SVMw/o with different proportions of labeled data

As represented in Fig. 1, detection accuracy of SVMw is better than SVMw/o apparently when the labeled data are scarce. With the increase of the number of labeled data, the curve of SVMw/o ascends sharply. When the proportion of labeled data reaches 50%, SVMw/o becomes better than SVMw. The reason is that the extended labeled dataset maybe consist of wrongly labeled data, which can degrade the accuracy of SVM. Therefore, it is considered that extended labeled dataset helps to improve the detection accuracy when the labeled data are scarce.

Secondly, we evaluate whether our SSAD algorithm can improve the detection accuracy in evolving data stream. In the experiment, we divide the dataset into 12 equal size of chunks according to their arriving time. Here, we assume that the data arrive with uniform speed. We apply SSAD with different half-lift ρ to each chunk where ρ equals 0.5, 1.0 and 5.0. None denotes that attenuation rule is not used. The DR and FPR of each chunk is showed in Fig. 2 and Fig.3, respectively.

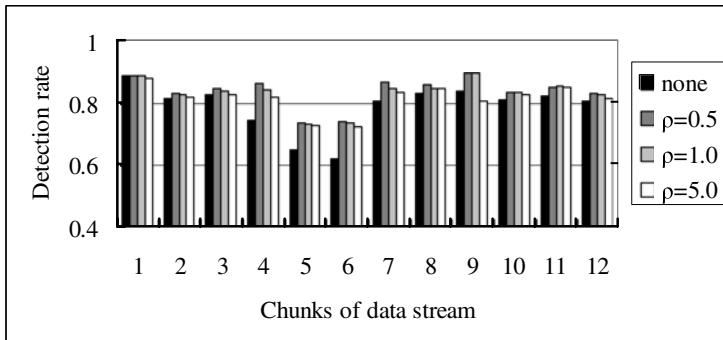


Fig. 2. Detection rate of SSAD algorithm for evolving data stream

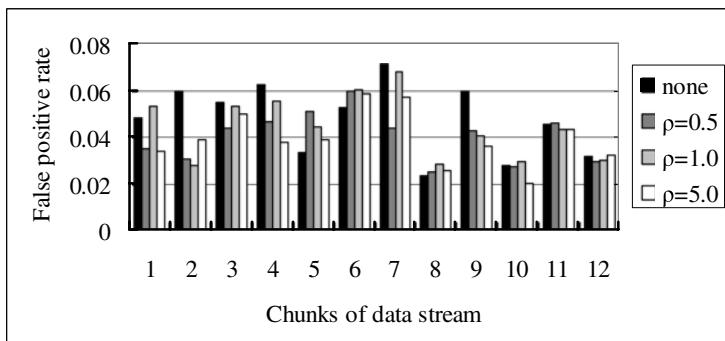


Fig. 3. False positive rate of SSAD algorithm for evolving data stream

From Fig. 2 and Fig. 3, we can notice that the DRs without attenuation are the lowest among of all chunks under the condition of low FPR. That implies that the detection algorithm learned equivalently of all historical dataset can not adapt to the change of traffic pattern well. Therefore, it can be considered that forgetting historical data gracefully will help detection algorithm to adapt to the change of traffic pattern. Accordingly, detection accuracy will be better than that based on all historical data.

5 Conclusions

This paper proposes an anomaly detection algorithm for evolving data stream based on semi-supervised learning, SSAD. Aiming at the problem of deficiency of labeled data in network traffic, SSAD uses semi-supervised learning to extend labeled dataset. At the same time, data stream model is used to characterize the time-varying traffic pattern, which help SSAD to adapt to the change of traffic pattern. The experimental results manifest that SSAD can achieve higher accuracy than that learned equivalently of all historical data, as while as be suitable for scarce labeled data.

References

1. Paxson, V.: Bro: A System for Detecting Network Intruders in Real-Time. Computer Networks 31(23–24), 2435–2463 (1999)
2. Porras, P.A., Neumann, P.G.: EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In: 9th National Computer Security Conference, pp. 353–365 (1997)
3. Burbeck, K., Nadjm-Tehrani, S.: ADWICE – Anomaly Detection with Real-Time Incremental Clustering. In: Park, C.-s., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 407–424. Springer, Heidelberg (2005)
4. Kasabov, N.: Evolving Connectionist Systems: The Knowledge Engineering Approach. Springer, London (2007)
5. Basu, S., Bilenko, M., Mooney, R.J.: A Probabilistic Framework for Semi-Supervised Clustering. In: 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 59–68. ACM Press, New York (2004)
6. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained K-Means Clustering with Background Knowledge. In: 18th International Conference on Machine Learning, pp. 577–584. Morgan Kaufmann, San Francisco (2001)
7. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and Issues in Data Streams. In: 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 1–16. ACM Press, Madison (2002)
8. Aggarwal, C.C.: On Change Diagnosis in Evolving Data Streams. IEEE Trans. Knowl. Data Eng. 17(5), 587–600 (2005)
9. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In: 1996 ACM SIGMOD International Conference on Management of Data, pp. 103–114. ACM Press, Montreal (1996)
10. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
11. Mukkamala, S., Sung, A.H., Abraham, A.: Intrusion Detection using An Ensemble of Intelligent Paradigms. J. Netw. Comput. Appl. 28(2), 167–182 (2005)
12. The UCI KDD Archive,
<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
13. Joshi, M., Agarwal, R., Kumar, V.: Predicting Rare Classes: Can Boosting Make Any Weak Learner Strong? In: 8th ACM Conference ACM SiGKDD International Conference on Knowledge Discovery and Data Mining, pp. 297–306. ACM Press, Edmonton (2002)

An Incident Analysis System NICTER and Its Analysis Engines Based on Data Mining Techniques

Daisuke Inoue¹, Katsunari Yoshioka², Masashi Eto¹, Masaya Yamagata³,
Eisuke Nishino⁴, Jun'ichi Takeuchi⁴, Kazuya Ohkouchi⁵, and Koji Nakao¹

¹ National Institute of Information and Communications Technology (NICT)

² Yokohama National University

³ NEC Corp.

⁴ Kyushu University

⁵ Hitachi, Ltd.

dai@nict.go.jp, yoshioka@ynu.ac.jp, eto@nict.go.jp,
yamagata@da.jp.nec.com, nishino@kairo.csce.kyushu-u.ac.jp,
tak@csce.kyushu-u.ac.jp, kazuya.okochi.ak@hitachi.com,
ko-nakao@nict.go.jp

Abstract. Malwares are spread all over cyberspace and often lead to serious security incidents. To grasp the present trends of malware activities, there are a number of ongoing network monitoring projects that collect large amount of data such as network traffic and IDS logs. These data need to be analyzed in depth since they potentially contain critical symptoms, such as an outbreak of new malware, a stealthy activity of botnet and a new type of attack on unknown vulnerability, etc. We have been developing the Network Incident analysis Center for Tactical Emergency Response (NICTER), which monitors a wide range of networks in real-time. The NICTER deploys several analysis engines taking advantage of data mining techniques in order to analyze the monitored traffics. This paper describes a brief overview of the NICTER, and its data mining based analysis engines, such as Change Point Detector (CPD), Self-Organizing Map analyzer (SOM analyzer) and Incident Forecast engine (IF).

1 Introduction

Malwares, such as worms, virus, and bots are spread all over cyberspace and often lead to serious security incidents that can cause significant damages to both infrastructures and end users. To grasp the present trends of malware activities over networks, a number of ongoing network monitoring projects are already in their operational phase [1]-[11]. In these projects, a great amount of time series data, such as network traffic and IDS log, are collected. These data need to be analyzed in depth since they potentially contain critical symptoms, such as an outbreak of new malware, a stealthy activity of botnet and a new type of attack on unknown vulnerability, etc. As powerful tools for the in-depth analysis on time series data, data mining technologies have been paid a great attention in the field of cybersecurity.

To fight against cybersecurity issues caused by the malwares, we have been developing the Network Incident analysis Center for Tactical Emergency Response

(NICTER) [12][13], which integrates a large-scale network monitoring and a fully automated malware analysis. This integrated system is capable of correlating the phenomena observed in the network monitoring and their root causes, i.e., malwares. The NICTER has several analysis engines taking advantage of data mining techniques such as Change Point Detector (CPD), Self-Organizing Map analyzer (SOM analyzer) and Incident Forecast engine (IF). The purposes of the engines are: 1) to detect a rapid change in monitored traffic far earlier than human operators, 2) to classify observed attacking hosts in terms of their network behaviors, and 3) to predict a drastic change in monitored traffic several hours ahead.

In this paper, we describe a brief overview of the NICTER, and focus on the analysis engines that use data mining technologies in order to analyze time series data resulting from the large-scale network monitoring. In Sect. 2, we explain the overview of NICTER. In Sect. 3, we describe the data mining based analysis engines in the NICTER. Finally, in Sect. 4, we give a conclusion.

2 Overview of NICTER

There are two main approaches to fight against malwares: macroscopic and microscopic approaches. The macroscopic approach is based on network monitoring and focuses on grasping the present trends of malicious activities over networks. The microscopic approach is focused on analyzing malware executables to obtain a deep understanding of their characteristics and behaviors. The NICTER integrates both the macroscopic and microscopic approaches.

The NICTER is composed of four systems as depicted in Fig. 1, namely, the Macro analysis System (MacS), the Micro analysis System (MicS), the Network and malware enchain System (NemeSys), and the Incident Handling System (IHS). The MacS uses distributed sensors to monitor *darknet* in real-time. A darknet is a set of globally announced unused IP addresses and using it is a good way to monitor network attacks such as malware's scans. This is because there is no legitimate host using these addresses, and we can thus consider all incoming traffic as a consequence of some kind of malicious activities or that of a misconfiguration. All incoming traffic is input to analysis engines to detect incident candidates such as detection of new scan patterns or sudden increase of scans. We call the monitoring method that quietly monitors incoming packets of a darknet *black hole monitoring*. Meanwhile, the MicS captures malwares in the wild by utilizing several low/high interaction sensors (i.e., honeypots). Captured malware executables are then input to a malware behavior analyzer and a malware code analyzer to automatically extract their characteristics and behaviors. The NemeSys enchains the phenomena, i.e., incident candidates, and their root causes, i.e., malwares. Once it has been given an attacking host observed in the MacS, the correlation analyzer in the NemeSys outputs a list of malwares that have similar network behavior (i.e., scans) as the host. By finding the root causes of the observed network attacks, we have a much clearer view of what is happening in the Internet. Finally, the IHS helps the operator to diagnose the results from the above analyses and make an incident report.

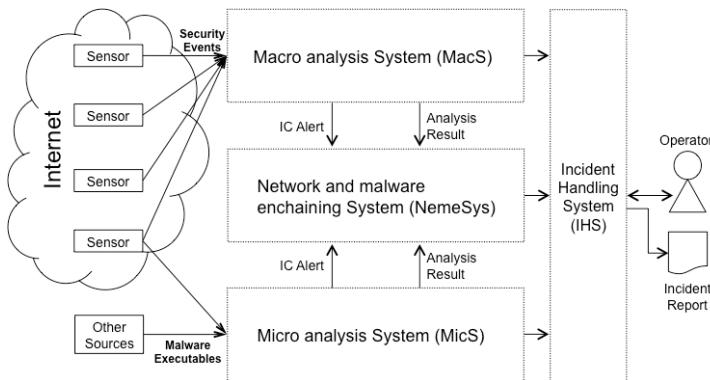


Fig. 1. Overview of NICTER

3 Analysis Engines Based on Data Mining Techniques

The Macro analysis System in the NICTER consists of widely distributed black hole sensors and various visualization and analysis engines. These sensors are monitoring large-scale darknet and sending observed traffic to the visualization and analysis engines. The NICTER presently has three analysis engines using data mining techniques: CPD, SOM analyzer and IF. The CPD looks for the change points in time series data that may indicate a shift of trends in monitored network attacks. As it is deployed in real time manner, the CPD is able to detect a rapid change in monitored traffic far earlier than human operators. The SOM analyzer classifies the observed attackers in terms of their behaviors and therefore it is useful to grasp the attackers' populations and potentially to detect new cluster of attackers including a rise of new botnet. The IF aims at forecasting network incidents in order to earn us time to respond against them. Although the IF is in the experimental phase, it is capable of two hours ahead prediction. These three analysis engines utilize unsupervised learning algorithms, because, in most cases in the cybersecurity field, the definite supervised data does not exist. This section describes the three analysis engines in the NICTER.

3.1 Change Point Detector (CPD)

The CPD is a time series analysis engine that uses two-stage on-line discounting learning based on the Auto-Regression (AR) model [14][15]. The purpose of the CPD is detecting a rapid change in monitored traffic promptly; therefore it is designed to achieve a low complexity for real-time detection. The CPD takes various time series data as an input such as the number of packets to a certain destination port per unit time (e.g., 10 minutes, 1 hour or 1day), the number of IDS alerts per unit time, the number of attacking hosts per unit time, etc. By analyzing these time series data, the CPD is capable of detecting the very beginning of network incidents such as an outbreak of new worms and a large scale DDoS attacks.

In a change point detection process, which is created for a time series data, the CPD firstly learns a statistical model from the time series data and calculates anomaly scores for each time point. After smoothing the anomaly scores, for each time point,

the CPD calculates a change point score that indicates the likelihood for the time point to be a change point. If the change point score exceeds a predefined threshold, it sends an alert to the Incident Handling System. The flow of the CPD is as follows.

1. **First stage of learning and scoring:** The CPD learns an AR model from a time series data of the darknet traffic using the on-line discounting learning algorithm. Then it calculates anomaly scores for each time point as the Shannon information according to the model learned so far.
2. **Smoothing:** The CPD prepares a fixed size window and constructs a time series of moving averages of the anomaly scores by sliding the window.
3. **Second stage of learning and scoring:** The CPD again learns an AR model from the time series of the moving-averaged anomaly scores. Then it calculates a change point score for each time point as the Shannon information of a moving-averaged score according to the model learned so far.

The CPD carries out the change point detection in real-time. The order of computation is $O(nk^2)$ where n is the sample size and k is the dimension of data.

Fig. 2 shows an adaptation of the CPD to detect a propagation of pandemic worm called MSBlast in August 2003. The figure illustrates a time series of the number of packets toward port 135/tcp (i.e., target port of the MSBlast), and its change point score. We could observe two distinct peaks in the change point score around Aug. 12th and 19th. It was actually reported that the MSBlast emerged in two steps. The CPD could promptly detect the first slight increase of scan packets, which was the very beginning of the outbreak of MSBlast.

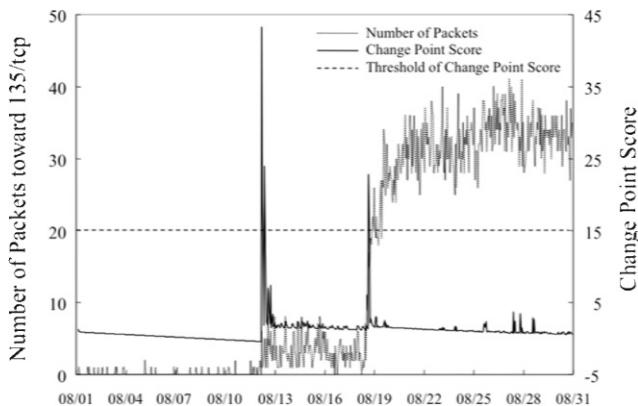


Fig. 2. Example of Worm Detection using CPD

3.2 Self-Organizing Map Analyzer (SOM Analyzer)

The SOM analyzer is a clustering (and visualization) engine of attacking hosts by means of the self-organizing map. The purpose of the analyzer is classifying and then detecting unknown malwares and their variants or a malicious activity of botnets, in terms of their network behaviors.

The SOM is an unsupervised learning neural network algorithm. Since the supervised data rarely exists in the cybersecurity field, we use the SOM instead of any supervised algorithm such as the nearest neighbor classifier. Another important feature of the SOM is its self-learning property, namely, high-dimensional input vectors that have higher similarity are located closer to each other in a lower-dimensional map. The SOM analyzer takes advantage of this property for making and visualizing clusters of attacking hosts [16].

The input vector of the SOM analyzer can include a variety of attributes such as a set of destination ports, the scan type (e.g., sequential or random), and the number of packets per unit time, etc. The flow of the SOM analyzer is as follows.

1. **Preprocessing:** The preprocessing module of the SOM analyzer slices the darknet traffic for each attacking host, and generates the input vectors of each host in every unit time (e.g., every five minutes).
2. **SOM calculation:** The SOM analyzer performs the SOM calculation a finite number of times (e.g., 100 times) with the input vectors.
3. **Postprocessing:** If the postprocessing module of the SOM analyzer detects a newly emerged or drastically changed cluster, it sends an alert to the Incident Handling System.

The SOM analyzer repeatedly executes the above flows with the batch learning in every unit time. Note that the reference vectors resulting from the SOM calculation will be used as the initial reference vectors for next calculation so as not to make a gap in visualization when the analyzer adapts new input vectors.

Fig. 3 shows the visualized results of the SOM analyzer, which illustrates the outbreak of the Dasher worm and its variants observed in December 2005. In the figure, a circle indicates a cluster of attacking hosts whose diameter corresponds to the number of hosts in the cluster.

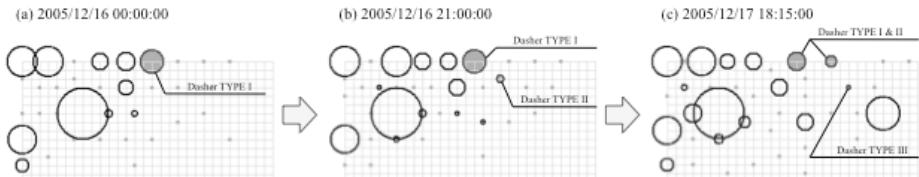


Fig. 3. Example of Worm Variants Detection using SOM Analyzer

On the first day, the SOM analyzer found a new cluster that only scanned 1025/tcp, which was the cluster of Dasher type I (in Fig. 3 (a)). On the same day, the analyzer found a new cluster again that scanned 1025/tcp, 42/tcp and 1433/tcp, which was the cluster of the Dasher type II (in Fig. 3 (b)). On the next day, the analyzer finally found one more new cluster that scanned 1025/tcp, 42/tcp, 1433/tcp plus 445/tcp, which was the Dasher type III (in Fig. 3 (c)). These Dasher variants had a similar scanning behavior however gradually added the scanning ports.

3.3 Incident Forecast Engine (IF)

The IF is a forecasting engine for future incidents [17]. The purpose of the IF is predicting the amount of traffic several hours ahead since the incidents are often accompanied by a mass traffic. The IF takes a time series data as an input such as the number of packets to a certain destination port per unit time (e.g., an hour). We expect that the IF enables us to gain the time to respond against the coming incident.

The IF is based on a wavelet decomposition and a model selection with the Akaike's Information Criterion (AIC). We use the Maximal Overlap Discrete Wavelet Transform (MODWT) [18], which is suited for prediction of the long memory time series such as network traffic. The model selection is used to detect a change of time series model from the stationary one to the one with keen positive trend. The flow of the IF is as follows:

1. **Wavelet decomposition:** The IF decomposes an input time series to wavelet coefficients (high frequency components) and scaling coefficients (DC component).
2. **Eight-hour ahead prediction:** The IF makes an eight-hour ahead prediction for each component, by means of the AR models for the wavelet coefficients (method in [19]), and of the polynomial regression for the scaling coefficients with determining the degree of polynomial based on the AIC.
3. **Inverse wavelet transform:** The IF makes a prediction for original time series using the inverse wavelet transform on the predicted components.
4. **Sending alert:** If the predicted value is significantly larger than the average value of the input time series in a certain past period, the IF issues an alert.

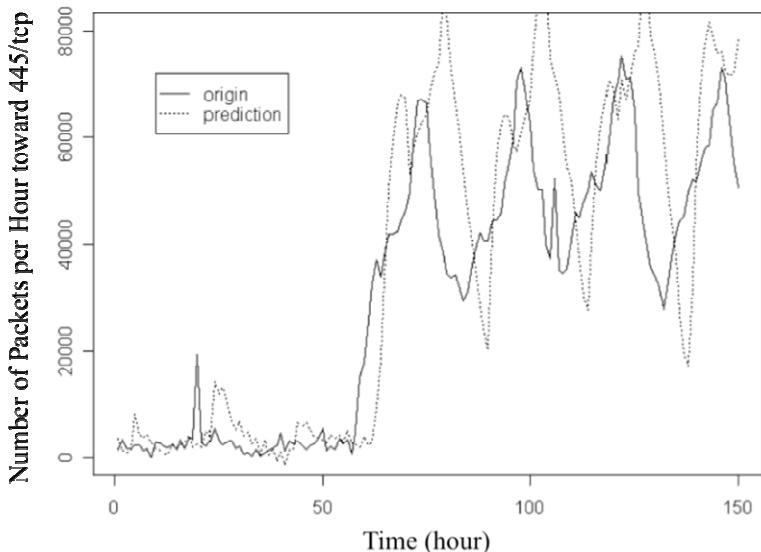


Fig. 4. Example of Incident Prediction using IF

In the eight-hour ahead prediction, we restrict the degree of polynomial within two, where degree two corresponds to the keen trend.

Fig. 4 shows an experimental result of prediction by the IF, where the input data is a time series of the number of packets per hour toward 445/tcp. The process of the eight-hour ahead prediction took some delay, however it was less than six hours. Consequently, the IF achieved two-hour ahead prediction of possible incidents.

4 Conclusion

This paper describes a brief overview of the NICTER, and its analysis engines that use the data mining techniques. The NICTER presently adopts three data mining based analysis engines, namely the CPD, the SOM analyzer, and the IF. The CPD looks for the change point in time series data that indicates a shift of trend in network attacks. The SOM analyzer clusters the observed attackers in terms of their behaviors, and therefore is useful to grasp the attackers' populations and potentially to detect new cluster of attackers including a rise of new botnets. The IF aims at forecasting network incidents in order to gain the time to respond against them. Presently, we are in the operational phase to evaluate the effectiveness of our engines. It is an important issue to properly configure and tune these engines so as to deal with the rapid changes in network attacks and to decrease the false positive/negative. Moreover, we will explore the potential of data mining technologies in the cybersecurity field.

References

1. Song, D., Malan, R., Stone, R.: A Snapshot of Global Internet Worm Activity. In: 14th Annual FIRST Conference on Computer Security Incident Handling and Response (2002)
2. Moore, D.: Network Telescopes: Tracking Denial-of-Service Attacks and Internet Worms around the Globe. In: 17th Large Installation Systems Administration Conference (LISA 2003), USENIX (2003)
3. Bailey, M., Cooke, E., Jahanian, F., Nazario, J., Watson, D.: The Internet Motion Sensor: A Distributed Blackhole Monitoring System. In: 12th Annual Network and Distributed System Security Symposium (NDSS 2005) (2005)
4. Pouget, F., Dacier, M., Pham, V.H.: Leurre.com: On the Advantages of Deploying a Large Scale Distributed Honeypot Platform. In: E-Crime and Computer Conference (ECCE 2005) (2005)
5. Leita, C., Pham, V.H., Thonnard, O., Ramirez-Silva, E., Pouget, F., Kirda, E., Dacier, M.: The Leurre.com Project: Collecting Threats Information using a Worldwide Distributed Honeynet. In: WOMBAT Workshop on Information Security Threats Data Collection and Sharing (WISTDCS 2008), pp. 40–57 (2008)
6. REN-ISAC: Research and Education Networking Information Sharing and Analysis Center, <http://www.ren-isac.net/>
7. Horenbeeck, M.V.: The SANS Internet Storm Center. In: WOMBAT Workshop on Information Security Threats Data Collection and Sharing (WISTDCS 2008), pp. 17–23 (2008), <http://isc.sans.org/>
8. JPCERT/CC, Internet Scan Data Acquisition System (ISDAS), <http://www.jpcert.or.jp/isdas/>
9. @police, <http://www.cyberpolice.go.jp/detect/observation.html>

10. MUSTAN, http://mustan.ipa.go.jp/mustan_web/
11. WCLSCAN, <http://www.wclscan.org/>
12. Nakao, K., Yoshioka, K., Inoue, D., Eto, M.: A Novel Concept of Network Incident Analysis based on Multi-layer Observations of Malware Activities. In: The 2nd Joint Workshop on Information Security (JWIS 2007), pp. 267–279 (2007)
13. Inoue, D., Eto, M., Yoshioka, K., Baba, S., Suzuki, K., Nakazato, J., Ohtaka, K., Nakao, K.: Nicter: An Incident Analysis System toward Binding Network Monitoring with Malware Analysis. In: WOMBAT Workshop on Information Security Threats Data Collection and Sharing (WISTDCS 2008), pp. 58–66 (2008)
14. Yamanishi, K., Takeuchi, J.: A Unifying Approach to Detecting Outliers and Change-Points from Nonstationary Data. In: The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002). ACM Press, New York (2002)
15. Takeuchi, J., Yamanishi, K.: A Unifying Framework for Detecting Outliers and Change-points from Time Series. IEEE Trans. on Knowledge and Data Engineering 18(4), 482–492 (2006)
16. Ohkouchi, K., Rikitake, K., Nakao, K.: A Study on Network Incident Analysis Using Self-Organizing Maps. In: The 2006 Symposium on Cryptography and Information Security (SCIS 2006) (2006)
17. Nishino, E., Ishitobi, K., Takeuchi, J., Yoshioka, K., Eto, M., Inoue, D., Nakao, K.: Forecast and Detection of Security Incident Based on the Network Traffic. In: The 30th Symposium on Information Theory and its Application (SITA 2007) (2007)
18. Percival, D.B., Walden, A.T.: Wavelet Methods for Time Series Analysis. Cambridge University Press, Cambridge (2000)
19. Daoudi, K., Frakt, A.B., Willsky, A.S.: Multiscale Autoregressive Models and Wavelets. IEEE Trans. on Information Theory 45(3), 828–845 (1999)

Multi-layered Hand and Face Tracking for Real-Time Gesture Recognition

Farhad Dadgostar, Abdolhossein Sarrafzadeh, and Chris Messom

Massey University, Auckland, New Zealand

{f.dadgostar, h.a.sarrafzadeh, c.h.messom}@massey.ac.nz

Abstract. This paper presents research leading to the development of a vision-based gesture recognition system. The system comprises of three abstract layers each with their own specific type and requirements of data. The first layer is the skin detection layer. This component provides a set of disperse skin pixels for a tracker that forms the second layer. The second component is based on the Mean-shift algorithm which has been improved for robustness against noise using our novel fuzzy-based edge estimation method making the tracker suitable for real world applications. The third component is the gesture recognition layer which is based on a gesture modeling technique and artificial neural-networks for classification of the gesture.

1 Introduction

Gesture recognition and gesture-based interaction is becoming an increasingly attractive research topic in human-computer interaction (HCI). There is a wide range of applications for this technology, including interactive games, performance analysis, surveillance monitoring, disability support, virtual reality and many others.

Although there are some commercially available solutions based on Digital Gloves and Body Markers, they are very expensive and also require physical attachment to the body. Consequently, these technologies are not widely available for general use and are sometimes seen as intrusive from the user's point of view.

Vision-based gesture recognition is an alternative approach which holds potential for being made available for a wider range of uses. Furthermore, it can also be employed in applications like surveillance monitoring, or video analysis in which the human subject is not under our control.

Vision-based gesture recognition has been a challenging area of research. It requires information on the movement of various body parts like the hands and head which is not easily extracted from a 2D image. For most of the applications, real-time performance is an important feature which makes the approaches involving time-consuming computations less favorable. Furthermore, adoption of a new technology in real world applications is affected by its cost and adaptability to the existing technology.

Based on the discussed features for a gesture recognition system, we took a component-based approach to system development. The system, contains three major components: i) skin detection, ii) hand and face tracking, and iii) gesture recognition. In the next section, we describe the research background of each component. In

Section 3, we discuss the components we have developed, the interface between them and the performance results of the complete system. Discussion and conclusions are presented in Section 4.

2 Research Background

The gesture recognition problem consists of pattern representation and recognition. Several methods have been used for gesture recognition: template matching, dictionary lookup, statistical matching, linguistic matching, neural-networks, Hidden-Markov models, and other ad hoc methods [1].

Hidden Markov Model (HMM) is used widely in speech recognition, and recently many researchers have applied HMM to temporal gesture recognition. However, because of the difficulty of data collection for training an HMM for temporal gesture recognition, the vocabularies are very limited. Moreover, to reach an acceptable level of accuracy, a great amount of data is required and a lot of time is spent to estimate the parameters of the HMM. Some researchers have suggested to use a better approach for more complex systems [2] which has remained an open question. Due to their known application, natural gestures have been the focus of some research including Wilson and Bobick's work [3].

Yang and Xu [1] proposed a method for developing a gesture-based interaction system using a multi-dimensional HMM. They used the Fast Fourier Transform (FFT) to convert the input gesture to a sequence of symbols to train the HMM. Zhu, Ren, Xu and Lin [4] used visual spatio-temporal features of the gesture for developing a real-time gesture controller which included visual modeling, analysis, and recognition of continuous dynamic hand gestures.

Watnabe and Yachida [5] proposed a method of gesture recognition from image sequences. The input image is segmented using maskable templates and then the gesture space is constituted by Karhunen-Loeve (KL) expansion. Oka, Satio and Kioke [6] proposed a gesture recognition method based on measured finger trajectories for an augmented desk interface system. They used a Kalman-Filter for predicting the location of multiple fingertips and HMM for gesture detection. New, Hasanbelliu and Aguilar [7] proposed a gesture recognition system based on hand-shape template matching for hand tracking and detecting the number of fingers being held up to control an external device. Lementec and Bajcsy [8] proposed an arm gesture recognition algorithm from Euler angles acquired from multiple orientation sensors for controlling unmanned aerial vehicles with manned aircrew. The gesture trajectory itself can be translated into a temporal matching problem as discussed in Black and Jepson [9].

3 Vision-Based Gesture Recognition

In this section, we introduce the components of our vision-based gesture recognition system. Each component provides the output for the next layer of the application but can also be used individually. This system was designed for in-door HCI¹ applications. The assumption in this setting was that the camera is in a fixed position, and the background is not moving.

¹ Human Computer Interaction.

3.1 Skin Detection

The purpose of the first component of the system is to provide reliable skin detection. The core of this component is our adaptive skin detection algorithm. In ideal conditions (e.g. using a blue background) non-adaptive skin detection might have been sufficient. Our adaptive skin detection algorithm was originally introduced by the authors [10]. This is a histogram based segmentation algorithm which adapts its thresholds based on the local information extracted from the video sequence. This algorithm has four main steps as follows:

- 1) Training the Global Skin Detector:** By using a set of training data, the thresholds of the skin color in hue color space are recognized. The Global Skin Detector can detect skin properly but may falsely detect some non-skin pixels. To reduce the false detection, we introduced the next steps to make this method adaptive and more accurate.
- 2) Detection of in-motion skin pixels:** The next step is detecting the in-motion pixels of the image and filtering the detected pixels using the Global Skin Detector. For this purpose, any kind of motion detection algorithm can be used. However, frame subtraction which is one of the simplest methods, shows acceptable results with a fixed camera and static background. The output of this step is those pixels with a higher probability of belonging to the skin regions of the image.
- 3) Recalculating the thresholds:** In the third step, the pixels that were considered as moving pixels belonging to the user's skin are used for retraining the detector. In this research we have used a histogram of Hue factor as the base for calculating low (T_L) and high (T_U) thresholds for filtering the image. From the in-motion skin pixels, another histogram is extracted and the second histogram is merged with the original histogram using the following equation:

$$H_{n+1} = (1-A)*H_n + A*H_M$$

H_{n+1} is the new histogram for skin detection (for the next frame)

H_n is the histogram for skin detection in the current frame

H_M is the histogram of the in-motion pixels of the skin colour and A , is the weight for merging the two histograms.

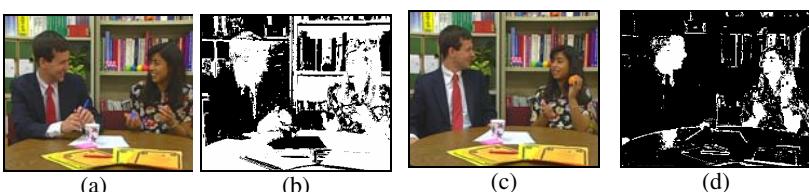


Fig. 1. The output of the adaptive skin detection algorithm on the input image sequence: a) first frame, b) first filtered image, c) the last frame (frame 90), d) the last filtered frame

- 4) Filtering using adaptive skin detector:** For each frame, the thresholds of the Hue factor are recalculated such that they cover 90% of the area of the new histogram. Finally, the filter for each frame is described as follows.

$$f(I) = \begin{cases} \text{true} & \text{if } T_L(H_n) \leq I \leq T_U(H_n) \\ \text{false} & \text{else} \end{cases}$$

I , is the Hue factor for each pixel

H_n is the Hue histogram for the skin colour

T_L is the calculated lower threshold for histogram H_n

T_U is the calculated upper threshold for the histogram H_n .

3.2 Object Tracking and Boundary Estimation

The next step following the detection of the skin in the image is tracking the blobs of pixels. One of the drawbacks of the fast skin segmentation algorithms are producing disperse multiple detections. This feature makes the output more favorable for mass tracking algorithms and specifically the Mean-shift algorithm. One of the limitations of the Mean-shift tracking algorithm is the lack of estimation of the size of the tracked object. The original implementation of the Mean-shift algorithm for face tracking known as the CAM-Shift² algorithm [11], uses a measurement function based on the density of the kernel which is not robust in some conditions mainly because of its underlying assumptions on the quality of the output of the skin detection algorithm.

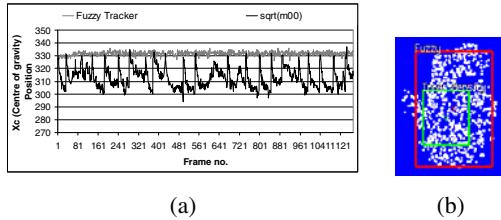


Fig. 2. a) Behavior of the algorithms with a Salt and Pepper noise level of 20%, b) Correct detection determined by edge density-fuzzy, smaller rectangle is the result of the CAM-Shift algorithm in comparison with our Fuzzy-based Mean-shift algorithm (the bigger rectangle)

Our contribution here is the introduction of the fuzzy-based boundary detection for estimating the boundaries of the kernel. The result of one of the experiments demonstrating the stability of this method against noise is presented in Fig 2. More details and the results of experiments with this algorithm can be found in Dadgostar, Sarrafzadeh and Overmyer [12].

3.3 Gesture Modeling and Interpretation

The result of the tracker is a sequence of coordinates showing the position of the object of interest over time. The task of the gesture interpretation system is to recognize a gesture pattern in the motion history which is to be performed by the third component.

This component was built based on our novel feature selection technique and a single classifier. Feature selection in this context is a challenging task. The number of the selected features should be as small as possible to be efficiently used by the classifier. There are also other issues such as sensitivity to input noise which may be caused

² Continuously Adaptive Mean-Shift.

by vibrations of the hand, small rotation and scale which may vary from person to person or with different input devices.

We have used the angle space for modeling 2D gestures measured as follows.

$$G_m = \left\{ \alpha_t \mid \alpha_t = \tan^{-1} \left(\frac{y_t - y_{t-1}}{x_t - x_{t-1}} \right) \right\}_{t=1..m}$$

- where G_m is a gesture signal recorded between time steps 0 to m .
- α_t is the angle at time step t
- (x_t, y_t) represents the coordinate of the gesture movement trajectory at time-step t .

To reduce the effect of vibration and also a smaller number of feature-values, the calculated angle is quantized to values of 10° . Thus, after quantization each sample will have a value between 0 to 350° . Hence, the input gestures can be described as a finite set of integer values which implicitly include the time and the direction of the gesture movements. With this approach, a *gesture* is translated to a *gesture signal* (Fig 3) reducing the gesture recognition problem to a signal matching problem.

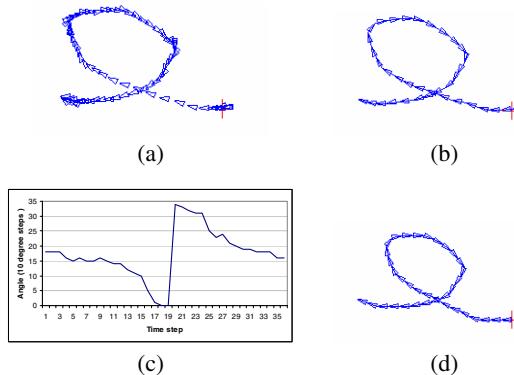


Fig. 3. a) Original gesture trajectory, b) sampling from the gesture trajectory, c) gesture signal of the collected data over time, d) reconstructed gesture using sampled data from c)

3.4 Gesture Signal Classification

In the next step, a comprehensive set of gesture signals with some freedom in terms of the size and position of gesture signals were selected. For this purpose, using the gesture recording software and a digital tablet, a total of 7392 gesture signals of 13 gesture classes were recorded by three individuals (Two males and one female.).

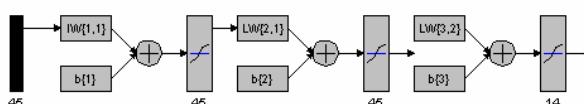


Fig. 4. The structure of ANN for detecting multiple gesture signals

For gesture classification, an ANN with 45 inputs (the maximum signal size) and 14 outputs was designed. This is a single feed forward ANN with 45 inputs and 14 outputs for classification (Fig 4).

For training the ANN, each element of the training dataset was labeled by the relevant gesture numbers from 1 to 13 and random gesture signals labeled as 14. The method of size adjustment in this prototype was based on padding the input vectors with random values. Hence, the test and the training data both carry two types of noise. The first one is what was recorded together with the user's input (e.g. vibrating hand). The second one is the random values which were added to the gesture signal. It is expected that by using this strategy, the ANN adjusts its weights such that some of the inputs are less sensitive in classification and at the same time able to classify the real input data robustly. The results indicating the accuracy of the ANN for each gesture signal are presented in Table 1. Table 2 presents the accuracy of the proposed gesture recognition technique in comparison to other techniques introduced in the literature.

Table 1. The evaluation of a single ANN in classifying 14 classes using 2 hidden layers

Gesture No.	Correctly detected	Total correct	Correct positive detection/%	Falsely detected	Total false	False positive to correct positive detection (%)	False positive detection ($\times 10^4$)
						Correct	Incorrect
1.	164	165	99.39	6	2424	0.25	2.4
2.	166	169	98.22	1	2420	0.04	0.4
3.	151	153	98.69	3	2436	0.12	1.2
4.	137	139	98.56	7	2450	0.29	2.9
5.	160	166	96.38	3	2423	0.12	1.2
6.	160	164	97.56	2	2425	0.08	0.8
7.	146	159	91.82	1	2430	0.04	0.4
8.	159	166	95.78	2	2423	0.08	0.8
9.	156	156	100	0	2433	0	0
10.	130	140	92.85	3	2449	0.12	1.3
11.	156	163	95.71	0	2426	0	0
12.	175	179	97.76	4	2410	0.17	1.6
13.	176	185	95.14	3	2404	0.12	1.3
14.	419	485	86.39	14	2104	0.58	6.6
Average		96.02			0.14	1.5	

Application / Description	technique applied	Accuracy		Gesture alphabet
		Correct positive	False positive	
Movement trajectory detection [6]	HMM	97.5% - 99.2%	NA	Drawing three shapes (circle, triangle, square)
Mobile sign language recognition using [13]	GT ² K – HMM	52.38% (Vision-based) 65.87% (Accelerometer)	NA	40 words vocabulary in a controlled environment
Workshop activity recognition [13]	GT ² K – HMM	93.33%	NA	10 gestures
Movement analysis based on intensity image sequence and a hand mask image sequence [4]	Spatio-temporal appearance in 2D space	89%	NA	12 gestures
Dynamic hand writing recognition [1]	HMM and STFT for feature selection	99.7%	NA	9 gestures
Gesture controller [4]	Gesture signal matching and DTW	89.6%	NA	12 gestures
Gesture movement trajectory recognition	Single ANN and angle space	96.02%	0.14%	13 gestures

3.5 Implementation

The implementation was carried out using the ANN described in the previous section. Then, the parameters of the ANN including weights and bias values were exported to a data file and used by a feed-forward ANN implementation in C++. The processing power which the ANN required to process its inputs was also measured. It was on average equal to 68 microseconds on a PC with a Pentium 4 - 2.8GHz processor. This also indicates that the ANN itself requires just 0.2% of the CPU processing time indicating that it can be used in real-time.

The inputs of the ANN were provided through an input buffer of size 45 containing the gesture's movement trajectory signal. In each time-step the detected trajectory angle was pushed into the input buffer. Then, the input buffer was delivered to the ANN for processing and the outputs of the ANN were recorded for demonstration. Figure 5 presents the output of the described implementation. The diagram on the top-right corner of the image represents the gesture signal. The other diagram on the bottom-right corner of the image represents the history of detected gesture signals. The horizontal axis of this diagram is frame number and the vertical axis is gesture number. A marker against each gesture number indicates one positive detection per frame for that gesture. As it is observable in this figure, gestures number 5 and 14 were detected. It is worth mentioning that detection of gesture number 14 has no particular meaning here. However, its presence indicates the existence of a noisy input signal³.

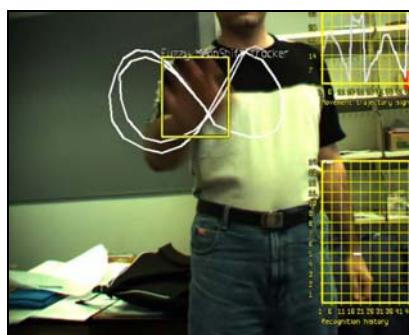


Fig. 5. Implementation of the vision-based gesture recognition using one ANN and the Fuzzy-based Mean-shift tracker. The diagram on the right bottom corner of the image represents the detected gestures over time.

4 Discussion and Conclusion

In this paper, we described the components we have designed and developed for our vision-based gesture recognition system. Based on this design there are three abstract layers: i) adaptive skin detection and segmentation, ii) hand and face tracking which is performed by a fuzzy-based Mean-shift algorithm, and iii) gesture recognition.

The gesture recognition system which is the main focus in this article, is based on our modeling technique for 2D gestures. This technique can be used with a variety of front-end input systems such as vision-based input, hand and eye tracking, digital tablet, mouse, and digital glove. We have used an ANN for gesture classification. However, the gesture signal can be applied with other classifiers such as support vector machines or Eigen vector-based classifiers.

These components together make our prototype gesture recognition system. The gesture recognition system has several applications. Firstly it provides a command-source for the user to communicate with the system through simple gestures. This

³ The original videos are available at:
<http://www.massey.ac.nz/~fdadgost/xview.php?page=videos>

facility can particularly be helpful for disabled users. It also has the potential and is in the least a step toward a replacement for the keyboard and mouse paradigm.

Vision-based gesture recognition technology is still in the early stages of development but it is growing steadily. According to Gartner, in the next few years, gesture recognition will probably be used primarily in niche applications [14]. It is said that gesture recognition is one of the five technologies which will change the way business is done in the future⁴.

References

1. Yang, J., Xu, Y.: Gesture Interface: Modeling and Learning. In: IEEE International Conference on Robotics and Automation (1994)
2. Perrin, S., Cassinelli, A., Ishikawa, M.: Gesture recognition using laser-based tracking system. In: Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition (2004)
3. Wilson, A.D., Bobick, A.E., Cassell, J.T.: Classification of natural gesture and application to video coding. In: Computer Vision and Pattern Recognition (1997)
4. Zhu, Y., et al.: Toward real-time human-computer interaction with continuous dynamic hand gestures. In: Proceedings. Fourth IEEE International Conference on Automatic Face and Gesture Recognition (2000)
5. Watanabe, T., Yachida, M.: Real time gesture recognition using eigenspace from multi-input image sequences. In: Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan (1998)
6. Oka, K., Sato, Y., Koike, H.: Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems. In: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (2002)
7. New, J.R., Hasanbelliu, E., Aguilar, M.: Facilitating User Interaction with Complex Systems via Hand Gesture Recognition. In: Proceedings of the 2003 Southeastern ACM Conference, Savannah, GA (2003)
8. Lementec, J.-C., Bajcsy, P.: Recognition of arm gestures using multiple orientation sensors: gesture classification. In: Proceedings of The 7th International IEEE Conference on Intelligent Transportation Systems (2004)
9. Black, M.J., Jepson, A.D.: A Probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, p. 909. Springer, Heidelberg (1998)
10. Dadgostar, F., Sarrafzadeh, A.: A Fast Real-Time Skin Detector for Video Sequences. In: Kamel, M.S., Campilho, A.C. (eds.) ICIAR 2005. LNCS, vol. 3656, pp. 804–811. Springer, Heidelberg (2005)
11. Bradski, G.R.: Computer Vision Face Tracking For Use in a Perceptual User Interface. Intel Technology Journal 2(2), 1–15 (1998)
12. Dadgostar, F., Sarrafzadeh, A., Overmyer, S.P.: Face Tracking Using Mean-Shift Algorithm: A Fuzzy Approach for Boundary Detection. In: Tao, J., Tan, T., Picard, R.W. (eds.) ACII 2005. LNCS, vol. 3784, pp. 56–63. Springer, Heidelberg (2005)
13. Westeyn, T., et al.: Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition. In: Proceedings of the International Conference on Perceptive and Multimodal User Interfaces, Vancouver, B.C., Canada (2003)
14. Geer, D.: Will gesture-recognition technology point the way? Computer, 20–23 (2004)

⁴ <http://www.inc.com/magazine/20060701/column-freedman.html>

Towards a Reliable Evaluation Framework for Message Authentication in Web-Based Transactions Based on an Improved Computational Intelligence and Dynamical Systems Methodology

Dimitrios Alexios Karras¹ and Vasilios C. Zorkadis²

¹ Chalkis Institute of Technology, Greece, Automation Dept., Psachna, Evoia, Hellas, Greece, P.C. 34400

dakarras@ieee.org, dakarras@teihal.gr

² Data Protection Authority, 1-3 Kifisia Av. 11523 Athens, Greece
zorkadis@dpa.gr

Abstract. The strength of message authentication, digital signature and pseudonym generation mechanisms relies on the quality of the one-way hash functions used. In this paper, we propose two tests based on computational intelligence and evolutionary algorithms theory to assess the hash function quality, which may be used along with other known methods and thus comprise a testing methodology. Based on the known nonlinearity test, which might confirm uniformity of digests, we formulate two tests using Support Vector Machines (SVM)/ MLP neural networks as well as Genetic Algorithms (GA). Both tests attempt to confirm that the produced digests cannot be modeled and, moreover, that it is impossible to find two or more messages that lead to a given digest apart from involving brute force computations. Both tests are applied to confirm the quality of the well-known MD5 and SHA message digest algorithms.

1 Introduction

Hash functions are used in a number of applications, such as digital signatures, message authentication, data integrity, password protection and pseudo-random number generation. Hash functions take as input strings of variable length and produce, according to their mapping algorithm, output of fixed length, generally smaller than the input string, called digest or hash value [1, 2, 5, 7, 9, 10, 13]. Hash-functions which take as an additional input a secret key are called keyed hash functions. In this paper, we deal with unkeyed hash functions and use only for them the key-word (one-way) hash functions [4].

Hash functions used in applications such as databases, compilers and operating systems are required to be fast and pre-image resistant, i.e. it is infeasible to find a message that results to a given digest. However, these properties are not sufficient for hash functions used in cryptographic protocols. For instance, applications such as digital signatures and digital time stamping require that hash functions are collision resistant, i.e. that it is infeasible to find any pair of messages which result to the same digest [5].

Several one-way hash functions have been proposed and are, in the meantime, extensively used in security mechanisms, such as MD4, MD5, RIPEMD, HAVAL SHA-1 and SHA-2 [4, 8, 11, 12].

The evaluation of one way hash functions bases on security, computational efficiency, including memory requirements, hardware and software suitability, flexibility and simplicity characteristics. It should be computationally infeasible to find the message it resulted to a given digest (1st pre-image resistance), while it should be easy to compute the digest for a message. Given a message x and its corresponding digest, it should also be computationally hard to find another message y with the same digest (2nd pre-image resistance). In addition, cryptographic hash functions should be collision resistant, i.e. they should be 1rst and 2nd pre-image resistant and it should be infeasible to find two different messages with the same digest. Moreover, hash functions have to be near-collision resistant, which means that is hard to find two messages resulting in digests with small difference. Furthermore, the hash function output should be indistinguishable from a random oracle. A hash function that satisfies the first three properties mentioned above is sometime called a collision resistant hash function (CRHF).

As stated in [4], ‘computational easiness’ might mean polynomial time and space; or more practically, within a certain number of machine operations or time units. On the other hand, “hardness” might mean requiring effort far exceeding affordable resources or a lower bound on the number of operations or memory required in terms of a specified security parameter or the probability that a property is violated be exponentially small ([5]).

Computational efficiency refers to software and hardware implementations [6]. Memory requirements include gate counts for hardware implementations, and code size and RAM requirements for software implementations. Also, desired are design simplicity and flexibility in terms of parameterizable number of compression function rounds and parallel execution. The requirement for rapid execution of the algorithm leads to the selection of simple sets of operations, as in the case of MD5, SHA-1 and SHA-2, which, designed for 32-bit or 64-bit architectures, base on addition mod 2^{32} or 2^{64} . Furthermore, rapid execution implies the selection of simple data structures and operations to program and the avoidance of large substitution tables [4]. Regarding MD5 and SHA-1, the two one-way hash functions under consideration, if we suppose good non-linearity and uniform behavior, then by using brute force attack the difficulty of finding a message having a given digest is of the order of 2^{128} for the MD5 and of 2^{160} for the SHA-1. Correspondingly, as mentioned above, finding two messages with the same digest is of the order of 2^{64} for the MD5 and of 2^{80} for the SHA [4]. So, collisions in hash functions are easier to find than pre-images. In [5, 14, 15], cryptoanalytic attempts against MD5, SHA-1 and other hash functions are referenced or presented, which revealed weaknesses in the compression function of the algorithms.

1st and 2nd pre-image resistance are of comparable difficulty to achieve, since choosing appropriate nonlinear function operations makes the corresponding problems hard to solve. Therefore, both, the avoidance of structural flaws that are vulnerable to cryptoanalytic attacks such as differential analysis and the use of non-linear operations with uniform behavior, prevent the attacker of employing other methods than brute-force [4]. In the case of brute-force or birthday attacks, the length of the digests is critical, since it indicates the number of trials to find a message that leads to

a given digest or two messages with the same digest, provided each possible digest has equal probability to occur. In [10], a bound on the probability of a collision with a particular hash value is determined, as well as the probability that there are no collisions at all.

However, the above estimations base on the assumptions that the one-way hash functions behave non-linearly and each possible digest has the same probability to be the output of the one-way function [4].

2 Message Digests Quality Assessment Tests

In [4] we proposed an evaluation methodology consisted of three tests, the non-linearity, the entropy and the neural-based modeling. The first test addresses the question about the non-linear behavior of one-way hash functions. The entropy test assesses the 1st pre-image and the collision resistance of hash functions by computing the actual probability to find a message with a given digest or two messages with the same digest in relation to the digest length. The modeling test aims at revealing potential weaknesses in the algorithms, which may be exploited by attackers to break them. That latter test, however, did not provide a mechanism for properly selecting training and test sets in order to suitable involve the MLP (Multilayer Perceptrons) presented there. It is the goal of this paper to revise and improve the computational intelligence test proposed in [4] as well as to propose a complementary test based on the same improved design.

2.1 A Revised Direct Neural Network Based Test through Involving Genetic Operators to Properly Select Training and Test Sets

In [4] an MLP has been involved to model message digest derivation. That MLP has been trained and tested with pairs (M_i, d_i), $i=1\dots N$, where M_i is a message of n bits length and d_i its associated message digest. Messages in that test were selected only randomly without involving any other policy. In that case the test results seem good even if there are weaknesses in the hash functions involved. Since messages are uncorrelated in that test it is almost impossible to properly sample a space of 2^n possible messages when n is large and model hash function derivation. Therefore, that test proved to be rather weak. The idea is now to construct training and test sets from correlated only messages. If it is shown that correlated messages lead to uncorrelated digests or better to non modeled digests through involving neural networks modeling then, this evidence is much stronger than the previous one of [4] about the quality of both the digest and the relevant test.

To this end, it is important to devise a proper strategy for selecting correlated messages. This paper presents such a strategy emanating from the non-linearity test. The non-linearity test consists of measuring the impact on the digest bits of changing input message bits [4]. Formally, for each digest bit, the non-linearity test is defined as follows [4]. Let us assume a boolean function $f : F_2^n \rightarrow F_2$ where, F_2^n is the set of all boolean vectors with length n , and F_2 is the set {0,1}. The boolean function f is considered as passing the non-linearity test if it satisfies the *propagation criterion of degree k*, that is if

$$(\forall a : 1 \leq W_a(a) \leq k) P(f(X) = f(X \oplus a)) = \frac{1}{2}, \quad (1)$$

where X is the input message of length n bits, α is a boolean vector of length n and Hamming weight $W_H(\alpha)$, P defines probability and the symbol \oplus defines the xor-operation between boolean vectors. As one understands, application of the above formula (1) to X for $\alpha = 1,2$ leads to only $n + (n-1)*n/2$ new Boolean vectors. However, if $\alpha > 1$ then, depending on formula (1) above, it leads to combinatorial explosion, since it involves the whole 2^n message space. Actually, iteratively invoking formula (1) it bears some similarities with finding the correlates of input message X .

The problem here is to find as many correlates of input message X as possible without leading to combinatorial explosion and, on the other hand to properly sample such a space so that the space becomes as dense as possible and not sparse. Therefore, there is need to examine as many α values as possible in a computationally feasible way. To this end we propose involvement of genetic algorithms, which can search the whole 2^n message space in a computationally effective way. More specifically, consider a random input message X and all of its possible small changes implied by the above given formula (1) for $\alpha=1,2$. All these messages derived from X through utilizing (1) with $\alpha = 1,2$ form a population of messages associated with X . The proposed derivation of X correlates attempts to produce correlates based on both small input message X changes as well as on changes implied due to the genetic evolution (use of genetic operators) of the input message X . Therefore, we attempt to produce all input message X correlates by applying small and larger changes to that message. The genetic operators applied to the input message X associated population above are mutation and crossover. The final message population corresponding to X consists of the members of all populations produced through the following steps, and can be considered the correlates of message X .

- 1) Assume a random input message X
- 2) Application of formula (1) above with $\alpha=1,2$ produces the first population associated with X .
- 3) Application of genetic operators on the above message population -providing the parents - produces their children as members of the derived second population. However, in order to keep a child in the final population we examine its Hamming distance H with the original message X . If this distance is greater than $H = 2$ we keep the child otherwise it is discarded. After this step finishes we have an extended population, consisting of step 3 admissible children and population of step 2 above.
- 4) Apply step 3 above but now the Hamming distance should be greater than $H+1$, where H is given in step 3. Form the new extended population by adding the new admissible children and so on, while H less than n .

If we assume that involving the previously proposed approach we have derived a certain class of input message correlates, then, the above produced final population forms the training/test sets for the neural networks involved in the revised quality assessment tests we herein propose.

In the sequel we define the revised neural modeling test for evaluating message digest algorithms. The neural modeling test is based on either SVM (Support Vector Machines) or MLP (Multilayer Perceptron) neural network's ability to map any

input-output function onto network's structure and therefore, to model any input-output mapping implied by a set of training samples [3]. Therefore, to perform this test, a set of training samples is constructed by applying the one way hash function under consideration to a subset of the set of input messages derived above by the GAs. More specifically, the final population of the correlates of a randomly selected input message X , found by applying GAs as above defined in steps 1-4, comprises both training and test sets of the involved neural networks. Each training/test sample consists of the pair of the input message and the associated digest bits (pattern). This latter pattern is considered as the desired output one. Then, the generalization ability of the SVM/MLP is calculated by assessing the average error bits for all the test samples. If this average error is equal to digest's length then, it takes its maximum value, and of course it is an indication that the corresponding SVM/MLP cannot model such a one way hash function.

More formally, let us assume a training set of input messages M_i with length n and their associated digests d_i with length m . The one way hash function under evaluation could be considered as a set of m boolean functions f_k , $f_k : F_2^n \rightarrow F_2$ where, F_2^n is the set of all boolean vectors with length n and F_2 is the set {0,1}. We, therefore, consider a set of m neural networks of SVM/MLP type with which we attempt to model each of the boolean functions f_k . Each such function f_k estimates the k -th bit of the one way hash function digest which is associated with an input message M_i . The k -th SVM/MLP is trained using its own training set of pairs (M_i, k_i) , where k_i is the k bit of the one way hash function digest d_i corresponding to the input message M_i . R such pairs comprise the training sets of the m SVMs/MLPs. We then, consider the rest T similar pairs of the final population of $R+T$ pairs produced by the GAs through steps 1-4 above, with $R+1 \leq i \leq R+T$, and we estimate the generalization capability of each one of the m SVMs/MLPs in the set of T test samples (M_i, k_i) ($R+1 \leq i \leq R+T$ and $1 \leq k \leq m$). That is, the average generalization error is measured as

$$GE = (1/Tm) \sum_{i,k} (k_i - [k_i])^2, R+1 \leq i \leq R+T, 1 \leq k \leq m, \quad (2)$$

and $[k_i]$ is the predicted by the k -th SVM/MLP bit k_i , of the digest d_i . GE is maximum when $GE \cong 1$.

2.2 A New Neural Network Based Quality Assessment Test, the Digest Non-invertibility Test, for Message Digest Evaluation

In the sequel we define the new proposed neural modeling test for evaluating message digest algorithms, the so called digest non-invertibility test. This neural modeling test is based again on either SVM (Support Vector Machines) or MLP (Multilayer Perceptron) neural network's ability to model any input-output mapping implied by a set of training samples [3]. While, the previous test aimed at modeling message digests, that is, it aimed at calculating from any given message its digest, the presented new test aims at the opposite goal. More specifically, as it was explained in the introduction, a strong message digest algorithm produces digests from which its associated message cannot be predicted. That is, the corresponding hash function is not invertible. The goal of the herein presented test is precisely to show this property. This was impossible with the previous test which could only show whether hash function calculations

were complex or not to be modeled by nonlinear models as neural networks. With the present test, however, we explicitly attempt to assess the non-invertibility property of the hash functions involved that is, we attempt to assess whether it is hard to find the message given its digest. Similarly to the above specified test, a set of training samples is constructed by applying the one way hash function under consideration to a subset of the set of input messages derived above by the GAs. More specifically, the final population of the correlates of a randomly selected input message X , found by applying GAs as above defined in steps 1-4, comprises both training and test sets of the involved neural networks in this new test. Each training/test sample consists of the pair of the digest bits and its associated input message. This latter pattern is considered as the desired output one. Then, the generalization ability of the SVM/MLP is calculated by assessing the average error bits for all the test samples. If this average error is equal to message's length then, it takes its maximum value, and of course it is an indication that the corresponding SVM/MLP cannot predict the message given its digest.

More formally, let us assume a training set of input messages M_i with length n and their associated digests d_i with length m . The training/test pairs are now (d_i, M_i) . The one way hash function under evaluation could be considered as a set of n boolean functions $f_k, f_k : F_m \rightarrow F_2$ where, F_m is the set of all boolean vectors with length m and F_2 is the set {0,1}. We, therefore, consider a set of n neural networks of SVM/MLP type with which we attempt to model each of the boolean functions f_k . Each such function f_k predicts the k -th bit of the input message M_i , given its digest d_i . The k -th SVM/MLP is trained using its own training set of pairs (d_i, k_i) , where k_i is the k bit of the input message M_i corresponding to the message digest d_i . R such pairs comprise the training sets of the n SVMs/MLPs. We then, consider the rest T similar pairs of the final population of $R+T$ pairs produced by the GAs through steps 1-4 above, with $R+1 \leq i \leq R+T$, and we estimate the generalization capability of each one of the n SVMs/MLPs in the set of T test samples (d_i, k_i) ($R+1 \leq i \leq R+T$ and $1 \leq k \leq n$). That is, the average generalization error is measured as

$$GE = (1/Tn) \sum_{i,k} (k_i - [k_i])^2, R+1 \leq i \leq R+T, 1 \leq k \leq n, \quad (3)$$

and $[k_i]$ is the predicted by the k -th SVM/MLP bit k_i , of message M_i . GE is maximum when $GE \cong 1$.

3 Application of the Proposed Methodology in the Evaluation of SHA and MD5 Message Digests

We have conducted a series of experiments to illustrate the performance of the proposed methodology based on the above specified tests, for evaluation of one way hash functions. Namely, we have considered the MD5 algorithm and SHA with input messages of 512 bits. For comparison purposes we have included also, the nonlinearity test. The non-linearity test has been performed for 1 and 2 bit changes ($\forall \alpha: 1 \leq W_H(\alpha) \leq 2$), as described in [4], of 5000 different input messages $X_1, X_2, \dots, X_{5000}$. With these assumptions the result of this non-linearity test for MD5/ SHA is shown in Table 1.

Table 1. Results of the non-linearity test

Message Digest Algorithm	Average Probability $P(f_k(X) = f_k(X \oplus \alpha))$, for all digest bits k , calculated over 5000 messages	Standard deviation of the Probability $P(f_k(X) = f_k(X \oplus \alpha))$, for all digest bits k
MD5	0,5001	0,0015
SHA	0,50002	0,0012

It is obvious that the nonlinearity test does not reveal any drawback in the MD5, SHA algorithms. Concerning the neural modeling tests defined above we have considered a random message and by applying the above specified methodology of GAs (steps 1-4, section 2) we have produced a set of 5000 input messages for constructing the training set of the 128 and 160 SVM/MLPs, respectively, as well as 2000 messages for the test set respectively. Each SVM used 233 support vectors [3] while each MLP (with an architecture of 512-120-55-1 architecture) was trained with the on-line back-propagation learning rule [3] with the same learning rate and momentum coefficients of 0.4, 0.5 respectively, following the procedure outlined in the previous sections. The average generalization errors GE for the two proposed tests, estimated from the test set of 2000 pairs, as determined in the previous sections by equations (2), (3) above, has been estimated in Table 2. This table shows that especially with the new proposed non-invertibility test some small flaws have started to reveal for the message digests algorithms under investigation.

Table 2. Results of the proposed computational intelligence based quality assessment tests

Proposed Neural Modeling Tests	MD5 - Average GE	SHA - Average GE
Initial Direct Neural Modeling test-SVM (random training sets)	100%	100%
Initial Direct Neural Modeling test-MLP (random training sets)	100%	100%
Revised Direct Neural Modeling test-SVM (training/test sets through message correlates)	99,4%	99,6%
Revised Direct Neural Modeling test-MLP (training/test sets through message correlates)	99,8%	100%
Digest Non-Invertibility Neural Test-SVM (training/test sets through message correlates)	97,5%	98,1%
Digest Non-Invertibility Neural Test-MLP (training/test sets through message correlates)	98%	98,8%

4 Conclusions and Future Trends

We proposed and described an evaluation methodology for one-way hash functions, based on computational intelligence and Genetic Algorithms. The methodology consists

of two main tests, the direct neural modeling test as well as the digest non-invertibility test, both based on SVM/MLP neural models. It is shown that while the known nonlinearity test as well as an initial direct neural modeling test introduced by the authors, do not reveal any weak point in the message digests herein involved, on the other hand, the proposed two tests present some capabilities to reveal flaws in the algorithms. We intend to extent this methodology by introducing further and more improved tests based on computational intelligence techniques and to apply them to keyed hash functions as well.

References

1. Damgard, I.B.: Collision Free Hash Functions and Public Key Signature Schemes. In: Price, W.L., Chaum, D. (eds.) *EUROCRYPT* 1987. LNCS, vol. 304, pp. 203–216. Springer, Heidelberg (1987)
2. Damgård, I.B.: A Design Principle for Hash Functions. In: Brassard, G. (ed.) *CRYPTO* 1989. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (1990)
3. Haykin, S.: *Artificial Neural Networks. A comprehensive foundaiton*, 2nd edn. Prentice Hall, Englewood Cliffs (1999)
4. Karras, D.A., Zorkadis, V.: A Novel Suite of Tests for Evaluating One-Way Hash Functions for E-Commerce Applications. In: Proc. of IEEE Euro-micro Conference, pp. 464–468 (2000)
5. Menezes, A.J., Bvan Oorschot, P.C., Vanstone, S.A.: *Handbook of Applied Cryptography*. CRC Press, Boca Raton (1996)
6. NIST (National Institute of Standards and Technology), Announcing the Development of New Hash Algorithm(s) for the Revision of Federal Information Processing Standard (FIPS) 180-2, Secure Hash Standard (2007)
7. Peyravian, M., Roginsky, A., Kshemkalyani, A.: On Probabilities of Hash Value Matches. *J. Computers & Security* 17(2), 171–176 (1998)
8. Pfleeger, C.P.: *Security in Computing*. Prentice-Hall, Englewood Cliffs (1997)
9. Preneel, B.: Cryptographic Hash Functions. *Transactions on Telecommunications* 5, 431–448 (1994)
10. Schneier, B.: *Applied Cryptography*. John Wiley and Sons, Chichester (1996)
11. Simmons, G.J. (ed.): *Contemporary Cryptology, The Science of Information Integrity*. IEEE Press, Los Alamitos (1992)
12. Stinson, D.: Combinatorial Techniques for Universal Hashing. *J. of Computer and System Sciences* 48, 337–346 (1994)
13. Wegman, M.N., Carter, J.L.: New Hash Functions and Their Use in Authentication and Set Quality. *J. of Computer and System Sciences* 22, 265–279 (1981)
14. Wang, X., Yin, Y.L., Yu, H.: Finding collisions in the full SHA-1. In: Shoup, V. (ed.) *CRYPTO* 2005. LNCS, vol. 3621, pp. 17–36. Springer, Heidelberg (2005a)
15. Wang, X., Yu, H.: How to break MD5 and other hash functions. In: Cramer, R. (ed.) *EUROCRYPT* 2005. LNCS, vol. 3494, pp. 19–35. Springer, Heidelberg (2005)

Part VI

Special Session: Computational Models and Their Applications in Machine Learning and Pattern Recognition

A Neuro-GA Approach for the Maximum Fuzzy Clique Problem

Sanghamitra Bandyopadhyay and Malay Bhattacharyya

Machine Intelligence Unit, Indian Statistical Institute,
203 B. T. Road, Kolkata - 700108, India
{sanghami,malay_r}@isical.ac.in

Abstract. The maximum clique problem, into which many problems have been mapped effectively, is of great importance in graph theory. A natural extension to this problem, emerging very recently in many real-life networks, is its fuzzification. The problem of finding the maximum clique in a fuzzy graph has been addressed in this paper. It has been shown here, that this problem reduces to an unconstrained quadratic 0-1 programming problem. Using a maximum neural network, along with, chaotic mutation capability of genetic algorithms, the reduced problem has been solved. Empirical studies have been done by applying the method on a gene co-expression network and on some benchmark graphs.

1 Introduction

The Maximum Clique Problem (MCP), which is known to be NP-complete, accomplishes unearthing the largest complete subgraph of a graph [1]. In networks, depicting relations between the objects, this problem maps to finding the largest group of similar objects. Eventually, in many real-life networks, e.g., the WWW, social networks, ecological networks, protein interaction networks, the relations between the objects are not always crisp; rather they may be fuzzy or probabilistic. For such cases, MCP does not prove to be suitable as it suffers from the constraint of binary relation between the vertices (objects). To present an effective approach to this motivation, the fuzzy version of the Maximum Clique Problem has been addressed in this paper. This problem, which is described as the Maximum Fuzzy Clique Problem (MFCP), concerns the discovery of the largest association of vertices in a fuzzy graph [2]. The largest association of vertices referred to here is bounded by a threshold of association density.

Various neural networks have been developed to date for solving MCP of which the Maximum Neural Networks (MNN) have shown the most promising results [3][4][5][6]. Being inspired by the Ising model (principle of maximum entropy) and the Artificial neural networks (inherent parallelism), MNN is capable of producing near-optimum solutions of computationally intractable problems in bounded time [3]. Thanks to the work of Lee et. al. [4], the MNN model is known to converge to the optimal or near-optimum solution. This model has been improved suitably (using nonlinear self-feedback [5], incorporating rich chaotic dynamics [6]) for solving MCP severally. The problem MFCP, being a fuzzy

extension of the MCP, can therefore be best handled using this type of parallel neuron model. An MNN model, which incorporates chaotic mutation capability of genetic algorithms, is proposed in this paper for the solution of MFCP.

To evaluate the proposed method, a gene co-expression network (fuzzy graph) has been constructed from a benchmark dataset. On this graph and on some DIMACS maximum clique instances [7], the methodology has been tested. The results are very promising in finding fuzzy cliques from gene co-expression networks. It also identifies the maximum cliques from the DIMACS graphs successfully, without incurring extra computational costs in parameter tuning.

2 Maximum Fuzzy Clique Problem

The main problem is now introduced on a fuzzy complete graph (generalization of fuzzy graphs) with basic definitions and preliminaries. The term *graph*, will hereafter signify undirected labeled simple graphs and $|S|$ will denote the cardinality of a set S . The other notations are customary, unless specified otherwise.

2.1 Foundation of the Problem

Definition 1 (Fuzzy Complete Graph). A Fuzzy Complete Graph (FCG), $\tilde{G} = (V, \tilde{E}, \Omega)$, is defined as a graph in which V denotes the set of vertices, \tilde{E} denotes the set of fuzzy relations (v_i, v_j) ($v_i \neq v_j, \forall v_i, v_j \in V$) and Ω is a fuzzy membership function defined over the set \tilde{E} such that $\Omega : \tilde{E} \rightarrow (0, 1]$.

Definition 2 (Association Density of a vertex). Given an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, the Association Density, $\mu_{v_i/\mathbb{V}}$, of a vertex v_i of \tilde{G} is defined, with respect to a set of vertices \mathbb{V} ($v_i \notin \mathbb{V}$), as the ratio of the sum of the fuzzy edge weights between v_i and each of the vertices belonging to \mathbb{V} , and the maximum possible sum of edge weights between them. Evidently, the denominator of the aforesaid ratio represents the cardinality of the set \mathbb{V} . Thus, the Association Density of a vertex v_i with respect to \mathbb{V} is computed as,

$$\mu_{v_i/\mathbb{V}} = \frac{\sum_{v_j \in \mathbb{V}} \Omega_{v_i v_j}}{|\mathbb{V}|}. \quad (1)$$

In Eqn. (1), $\Omega_{v_i v_j}$ denotes the fuzzy membership value of the vertex pair (v_i, v_j) . This density definition computes the degree of participation of a single vertex. Putting the constraint of a lower bound to this density factor for every vertex within an association of vertices, a fuzzy clique is now defined as follows.

Definition 3 (Fuzzy Clique). Given an FCG $\tilde{G} = (V, \tilde{E}, \Omega)$, a fuzzy clique, $\mathcal{F}_\delta(\mathbb{V})$, is defined, with respect to an Association Density threshold δ , as the subgraph of \tilde{G} ($\mathcal{F}_\delta(\mathbb{V}) \subseteq \tilde{G}$) induced by the set of vertices \mathbb{V} such that,

$$\min_{\forall v_i \in \mathbb{V}} \left(\mu_{v_i/\mathbb{V}-\{v_i\}} \right) \geq \delta. \quad (2)$$

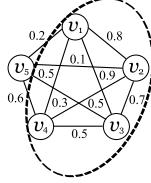


Fig. 1. A fuzzy clique associating the vertices $\{v_1, v_2, v_3, v_4\}$ w.r.t. $\delta = 0.5$

Eqn. (2) ensures that not only does a fuzzy clique attain a minimum overall density, but also all the vertices within it should have a minimum participation density. A fuzzy clique that occurs within an FCG with respect to Association Density threshold $\delta = 0.5$ is shown with a dotted circle in Fig. 1.

2.2 Formalism

Having introduced the required preliminaries, the formal statement of the problem, that will be tackled in this paper, is now given follows.

Problem Statement(MFCP). Given an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, and an Association Density threshold, δ , of a fuzzy clique, locate a fuzzy clique $\mathcal{F}_\delta(\mathbb{V}_{max})$ of \tilde{G} that has the maximum cardinality, i.e., $|\mathbb{V}_{max}| \geq |\mathbb{V}_i| : \forall \mathcal{F}_\delta(\mathbb{V}_i) \subseteq G$.

Follow that, for $\delta = 1$, MFCP simply reduces to MCP, which implies that the addressed generalized version (MFCP) is at least as hard as MCP. A heuristic methodology for solving this problem has been described in the following section.

3 Methods

The problem MFCP can be viewed as an optimization problem wherein the target is twofold. First, the associated vertices that will form the fuzzy clique must support the minimum Association Density threshold. And next, the size of the fuzzy clique (cardinality of the set of vertices it includes) is to be maximized. Combining both of these targets into a single objective function, a minimization problem can be formulated from MFCP that will be useful for the solution purpose. This is presented through the following significant derivations.

Theorem 1. *In MFCP, searching for the fuzzy clique $\mathcal{F}_\delta(\mathbb{V}_{max})$ in an FCG $\tilde{G} = (V, \tilde{E}, \Omega)$, for a given Association Density threshold δ , is equivalent to minimizing the following objective function,*

$$\mathbb{Z} = - \sum_{i=1}^{|V|} v_i + \sum_{i=1}^{|V|} \left[\delta \left(\sum_{j=1}^{|V|} v_j - 1 \right) - \sum_{j=1}^{|V|} \Omega_{ij} v_i v_j \right]; \forall v_i, v_j \in \{0, 1\} . \quad (3)$$

where, $v_i = 1$, if the vertex is in the fuzzy clique, otherwise $v_i = 0 : \forall v_i \in V$.

Proof. Consider a vertex v_i in the FCG $\tilde{G} = (V, \tilde{E}, \Omega)$. To be within the fuzzy clique $\mathcal{F}_\delta(\mathbb{V}_{max})$, v_i should have at least a cumulative edge weight of $\delta|\mathbb{V}_{max} - 1|$ from the remaining vertices. More formally, a vertex v_i is required to satisfy the following condition for being in $\mathcal{F}_\delta(\mathbb{V}_{max})$,

$$\left[\delta(|\mathbb{V}_{max}| - 1) - \sum_{j=1}^{|\mathbb{V}_{max}|} \Omega_{ij} \right] \leq 0 . \quad (4)$$

Considering, $v_i = 1$, if $v_i \in \mathbb{V}_{max}$, otherwise $v_i = 0$ for all the vertices $v_i \in V$, the condition in Eqn. (4) can be rewritten in a more generalized form as,

$$\left[\delta \left(\sum_{j=1}^{|V|} v_j - 1 \right) - \sum_{j=1}^{|V|} \Omega_{ij} v_i v_j \right] \leq 0 . \quad (5)$$

Using the above condition (minimization from an upper bound) for every vertex of the graph, that could be considered in $\mathcal{F}_\delta(\mathbb{V}_{max})$, the following objective function which is required to be minimized can be formulated,

$$\sum_{i=1}^{|V|} \left[\delta \left(\sum_{j=1}^{|V|} v_j - 1 \right) - \sum_{j=1}^{|V|} \Omega_{ij} v_i v_j \right] . \quad (6)$$

Maximization of the size of the fuzzy clique being another objective of MFCP, the complete minimization problem can be given as,

$$\min \mathbb{Z} = - \sum_{i=1}^{|V|} v_i + \sum_{i=1}^{|V|} \left[\delta \left(\sum_{j=1}^{|V|} v_j - 1 \right) - \sum_{j=1}^{|V|} \Omega_{ij} v_i v_j \right] . \quad (7)$$

Thus, the required formulation is achieved. \square

Theorem II provides a simple reduction from the MFCP to an Unconstrained Quadratic 0-1 Programming Problem (UQ0-1PP) where the variables (v_i, v_j) can only have the values 0 and 1 [8]. A few problems in UQ0-1PP have been mapped earlier to the neural network models [3] [4]. Inspired by the modified energy function developed in [4] for ensuring the generation of valid solution and avoiding the coefficient tuning problem, the following energy function can be derived that will fit to a neural network capable of solving the MFCP,

$$\mathbb{E} = \frac{C}{2} \sum_{i=0}^{|V|} \left[\delta \sum_{j=0, j \neq i}^{|V|} \sum_{k=1}^2 v_j^k - \sum_{j=0, j \neq i}^{|V|} \sum_{k=1}^2 \Omega_{ij} v_i^k v_j^k \right] . \quad (8)$$

In Eqn. (8), C is a constant. The MNN used here for solving MFCP contains $2 \times |V+1|$ neurons ($|V+1|$ clusters of 2 neurons). While using the energy function given in Eqn. (8), in the proposed MNN, the motion equation of the k^{th} neuron

in the i^{th} cluster becomes, $\frac{du_i^k}{dt} = -\frac{u_i^k}{\tau} - \frac{\partial \mathbb{E}}{\partial v^k}$. Here, the decay term $-\frac{u_i^k}{\tau}$ is known to perturb the convergence of the system [4]. Hence, by removing this term, the motion equation of the k^{th} neuron of the i^{th} cluster is derived as,

$$\frac{du_i^k}{dt} = -\frac{\partial \mathbb{E}}{\partial v^k} = -\sum_{j=0, j \neq i}^{|V|} (\delta - \Omega_{ij}) v_j^k . \quad (9)$$

Expanding Eqn. (9), following the Taylor series, and then, by approximating it using the first-order Euler method, the updating factors become,

$$\Delta u_i^k = \sum_{j=0, j \neq i}^{|V|} (\Omega_{ij} - \delta) v_j^k . \quad (10)$$

Notably, while mapping the MFCP, from a $|V|$ -vertex FCG, into an MNN, $|V+1|$ clusters of neurons are required that correspond to the vertices. To resolve this disparity, an extra vertex v_0 is attached with the original FCG. The edge weights between v_0 and the remaining vertices are given in the form of Eqn. (II) such that v_0 is always selected in $\mathcal{F}_\delta(\mathbb{V}_{max})$. The stable presence of v_0 in the solution set expedites the convergence of the system. The edge weights from v_0 are given by,

$$\Omega_{0i} = \Omega_{i0} = -\delta(|V| - 1) + \sum_{j=1, j \neq i}^{|V|} \Omega_{ij} . \quad (11)$$

Using the previous derivations, an MNN model (MNNGA), that incorporates chaotic mutation capability of genetic algorithms, is now proposed. The functionality of the model is described in the following algorithm.

Algorithm (MNNGA): An MNN model with chaotic genetic algorithms

- 1: Set $t \leftarrow 1$
- 2: $u_i^k(t) \leftarrow 0 : \forall i \in \{1, 2, \dots, |V|\}, \forall k \in \{1, 2\}$ // Initialize the input variables
- 3: **while** The output variables are not in equilibrium state **do**
- 4: **for** $i = 1$ to $|V|$ **do**
- 5: **for** $k = 1$ to 2 **do**
- 6: $v_i^k(t) \leftarrow 1$, if $u_i^k(t) = \max\{u_i^1(t), u_i^2(t)\}$ // Fire the outputs
 $v_i^k(t) \leftarrow 0$, otherwise
- 7: **end for**
- 8: **end for**
- 9: Invert the output variables $v_{pos}^1(t)$ and $v_{pos}^2(t)$ where pos is a randomly selected position over the output ports // Chaotic mutation operator
- 10: **for** $i = 1$ to $|V|$ **do**
- 11: **for** $k = 1$ to 2 **do**
- 12: $\Delta u_i^k(t) \leftarrow \sum_{j=0, j \neq i}^{|V|} (\Omega_{ij} - \delta) v_j^k$ // Compute input updating factors
- 13: **end for**
- 14: **end for**

```

15:  $u_i^k(t+1) \leftarrow (1 - z(t))u_i^k(t) + \alpha \Delta u_i^k(t) : \forall i \in \{1, 2, \dots, |V|\}, \forall k \in \{1, 2\}$   

// Update the inputs  

16:  $z(t+1) \leftarrow \beta^{mt} z(t)$   

17: Set  $t \leftarrow t + 1$  and go to Step 3 // Next iteration  

18: end while

```

This model initializes (Steps 1-2) the input variables with zero values at time step 1. Then it undergoes an iterative process (Steps 3-18), where the inputs are updated and the outputs are fired depending on the updated input value, until all the outputs do not change over some iterations. In the repetitive steps, the most important modification adopted (Step 9) is the mutation process (influenced by chaotic dynamics) that disturbs the outputs (Steps 4-8) to overcome the local minima problem. This is done by allowing an adaptive mutation process over every iteration. After that, the inputs are updated (Steps 10-16) with a much simplified model without impairing the performance of the algorithm. Finally, the solution (maximum fuzzy clique) is derived by considering the vertices that attain the value 1 at the output ports in the 1st layer of the network.

4 Experimental Results

To show the efficiency of MNNGA, it has been applied to locate the maximum fuzzy cliques (functionally associated genes) from an FCG (gene co-expression network). Data has been collected from a reverse transcriptioncoupled PCR experiment producing mRNA expressions of 112 genes during Rat Central Nervous System (Rat CNS) development [9]. The expression values ($\log_2(R/G)$) of 111 genes (removing one gene containing missing value) over 9 time points have been taken to compute the absolute leave-one-out Pearson correlation between each gene pair. Taking this value as the proximity measure, an FCG has been prepared. The size of the maximum fuzzy cliques, denoted by $\tilde{\omega}(\tilde{G})$, found in this FCG using MNNGA for various δ values is shown in column 2 of Table II.

The value of the parameters considered to be $\alpha = 0.01$, $\beta \lesssim 1$, $m = 0.7$, $z(t) = 0.15$. Silhouette Index ([−1,1]) is a quantitative measure of clustering results [10].

Table 1. Size of the maximum fuzzy cliques found in the FCG constructed from the Rat CNS dataset for different δ values

δ	$\tilde{\omega}(G)$ found	Silhouette Index
0.1	111	Not applicable
0.2	111	Not applicable
0.3	85	0.2239
0.4	72	0.2764
0.5	64	0.2869
0.6	58	0.3336
0.7	43.38 (Min = 43, Max = 44)	0.3626
0.8	25.46 (Min = 25, Max = 26)	0.5166
0.9	0	Not applicable

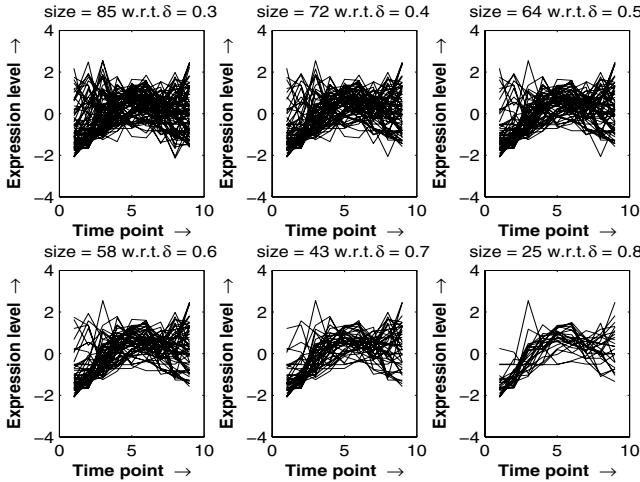


Fig. 2. Expression profile plots of gene set corresponding to the fuzzy cliques found in Rat CNS dataset for various δ

This value has been computed for every solution (except when none of the genes or all of the genes is the result). These values are given in the 3rd column of Table 1. The values indicate that the gene sets become more compact for higher δ values. The expression profile plots of the identified fuzzy cliques are shown in Fig. 2. Observably, the size of the maximum fuzzy cliques get reduced with the increment in the parameter δ . The figure also highlights that for higher δ values the pattern of the expression profiles become more correlated and compact, hence, they reveal functionally more associated gene set.

It has been discussed earlier that MFCP reduces to MCP for $\delta = 1$. So, a study has been made to compare the existing models like MNN with Nonlinear Self-feedback (MNNNS) [5], Chaotic MNN (CMNN) [6] and the proposed MNNGA to evaluate their efficacy in solving MCP. For each test graph taken from DIMACS instances [7], 50 simulations is done independently. The mean values of the results received are given under the 7th column in Table 2. The other values are from the literature. The heading, $\omega(G)$, over column 4 of the table denotes actual clique size. On comparing the results, MNNGA is found to compete with the existing ones involving very limited parameter tuning. The results clearly indicates the superiority of the chaotic mutation process adopted here.

Table 2. Comparative results on finding cliques from some DIMACS graphs

Test graph	Order value	Density [0, 1]	$\omega(G)$	MNNNS (3 parameters)	CMNN (8 parameters)	MNNGA($\delta = 1$) (4 parameters)
MANN_a9	45	0.9273	16	15.4	15.8	15.6
p_hat300-3	300	0.744	36	33.6	35.8	35.2
p_hat700-1	700	0.249	11	9	10.2	9
p_hat700-3	700	0.748	≥ 62	59.8	61.2	60.4

5 Concluding Remarks

This paper describes a useful problem of finding the largest association of vertices, forming a fuzzy clique, in a *fuzzy graph*. Many problems can be mapped into this general version problem for an effective solution. The MNNGA proposed for the solution of MFCP can also be used for solving the other problems in UQ0-1PP following the current study. Literature surveys show that the performance of neural networks in some models (also in MNN) are parameter specific. The tuning of parameters, thus becomes, a hectic task for specific problems. Therefore, the number of parameters have been minimized in the proposed simplified model opposing the earlier trends. In spite of this, MNNGA demonstrates good performance within limited iterations without having stuck into local minima.

References

1. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: The Maximum Clique Problem. In: Du, D.Z., Pardalos, P.M. (eds.) *Handbook of Combinatorial Optimization: Supplementary*, vol. A, pp. 1–74. Kluwer Academic, Dordrecht (1999)
2. Rosenfeld, A.: Fuzzy graphs. In: Tanaka, K., Zadeh, L.A., Fu, K.S., Shimura, M. (eds.) *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, pp. 77–95. Academic Press, New York (1975)
3. Lee, K.C., Funabiki, N., Cho, Y.B., Takefuji, Y.: A parallel neural network computing for the maximum clique problem. In: *International Joint Conference on Neural Networks*, Singapore, pp. 905–910 (1991)
4. Lee, K.C., Funabiki, N., Takefuji, Y.: A Parallel Improvement Algorithm for the Bipartite Subgraph Problem. *IEEE Transactions on Neural Networks* 3(1), 139–145 (1992)
5. Wang, J., Tang, Z., Wang, R.: Maximum neural network with nonlinear self-feedback for maximum clique problem. *Neurocomputing* 57, 485–492 (2004)
6. Yang, G., Tang, Z., Yi, J.: An Improved Chaotic Maximum Neural Network for Maximum Clique Problem. *International Journal of Computer Science and Network Security* 7(2), 1–7 (2007)
7. Johnson, D.S., Trick, M.A., Johnson, D.J.: Cliques, coloring, and satisfiability: second dimacs implementation challenge. *Dimacs Series on Discrete Mathematics and Theoretical Computer Science* 26, 11–13 (1993)
8. Barahona, F., Junger, M., Reinelt, G.: Experiments in quadratic 0-1 programming. *Mathematical Programming* 44(2), 127–137 (1989)
9. Wen, X., Fuhrman, S., Michaels, G.S., Carr, D.B., Smith, S., Barker, J.L., Somogyi, R.: Large-scale temporal gene expression mapping of central nervous system development. *Proceedings of the National Academy of Sciences* 95, 334–339 (1998)
10. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20, 53–65 (1987)

Hybrid Feature Selection: Combining Fisher Criterion and Mutual Information for Efficient Feature Selection

Chandra Shekhar Dhir^{1,3,*} and Soo Young Lee^{1,2,3}

¹ Department of Bio and Brain Engineering

² Department of Electrical Engineering and Computer Science

³ Brain Science Research Center

KAIST, Daejeon 305-701, Korea

Tel.: +82-42-869-5431; Fax: +82-42-869-8490

shekhardhir@neuron.kaist.ac.kr

Abstract. Low dimensional representation of multivariate data using unsupervised feature extraction is combined with a hybrid feature selection method to improve classification performance of recognition tasks. The proposed hybrid feature selector is applied to the union of feature subspaces selected by Fisher criterion and feature-class mutual information (MI). It scores each feature as a linear weighted sum of its interclass MI and Fisher criterion score. Proposed method efficiently selects features with higher class discrimination in comparison to feature-class MI, Fisher criterion or unsupervised selection using variance; thus, resulting in much improved recognition performance. In addition, the paper also highlights the use of MI between a feature and class as a computationally economical and optimal feature selector for statistically independent features.

1 Introduction

Raw sensory inputs from facial scenes impose higher computational demands, hence, restricting its application to real-time recognition tasks. In contrary, humans with normal vision perception can easily categorize individuals with dissimilar facial images under different lightening conditions. It is argued that human learning is mostly unsupervised which attempts to discover underlying low-dimensional representation of raw sensory data [1]. In this paper, we present a two-step approach to face recognition utilizing unsupervised feature extraction at first stage and supervised selection of extracted features at second stage.

Motivated by human unsupervised learning, several researchers have exploited linear low-dimensional representation of images which decomposes raw data to provide orthonormal basis using principal component analysis (PCA). Also, statistically independent representations from face images can be extracted using independent component analysis (ICA). ICA is a multivariate approach of data representation using statistically independent features and has found wide applications in the field on blind source separation and feature extraction [2]. In

* Corresponding author.

this paper, we also use non-negative matrix factorization (NMF) which linearly decomposes raw facial image data to learn its spatially localized parts [3].

Efficient data representation can be achieved by using the entire set of features extracted using unsupervised learning, however, all the features may not be relevant for improvements in classification tasks. Humans can classify face images in a supervised manner by utilizing *apriori* information about the classes. Working on this philosophy, Fisher criterion has been widely used as a feature selection criteria for classification tasks [7].

Fisher criterion is a sub-optimal feature selection method which assumes that the conditional distributions of feature given the class are Gaussian. This *apriori* assumption may not results in efficient selection of features with non-Gaussian distribution. On the other hand, Mutual information (MI) between feature and class has been widely used as a feature selection criteria [4][5]. The motivation to apply MI in the feature subspace is to maximize the information between the class and the given feature [6]. Statistical independence among ICA features makes MI criterion computationally scalable and an optimal feature selector. However, accurate estimation of conditional distribution of feature given the class is restricted by the number of available class samples which in turn affects the calculation of MI.

Considering these drawbacks of Fisher criteria and MI as a feature selection methods, a hybrid approach is proposed (refer to sec. 3.3 for more details). Experimental results shows that hybrid feature selection method show superior performance over MI and Fisher criteria. In addition, it also shows improved performance of MI as feature selection method over Fisher criteria.

The paper is organized as follows: In section 2, unsupervised feature extraction methods are discussed. Section 3 concentrates on feature selection using Fisher criterion, MI and presents the proposed hybrid feature selection method for applications to recognition tasks. Experimental results are presented in section 4 followed by conclusions.

2 Unsupervised Feature Extraction Methods

Classification of high resolution face images is computationally demanding if no pre-processing is done on the raw images. In an attempt to reduce the computational burden several efficient feature extraction methods have been studied [7][5]. PCA is a well-known method in the field of image processing and has been widely used to extract meaningful features. Working on same philosophy, ICA and NMF are also used to represent high dimensional image data into low dimension feature space.

2.1 Principal Component Analysis

From a given data matrix of N images with P pixels, $X_{P \times N} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, PCA finds the eigenfaces, $E_{P \times P} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_P]$, which are the orthonormal axis or the uncorrelated basis for representation of face data. Here \mathbf{x}_i is the

vector containing the pixel intensity of i -th image and \mathbf{e}_i is the i -th eigenface. Using singular value decomposition, the data can be represented as

$$\bar{X} = EDV^T \approx \sum_{r=1}^R \lambda_r e_r v_r^T, \quad (1)$$

where \bar{X} is the zero-mean image data, D is a diagonal matrix with elements $[\lambda_n]_{N \times N}$, V is a orthonormal matrix with elements $[v_{ij}]_{N \times P}$, and $[]^T$ is the transpose operator.

To achieve dimension reduction with minimal loss in data representation as shown in (3), we select a subset of eigenfaces,

$$E^* = [\mathbf{e}_1^*, \mathbf{e}_2^*, \dots, \mathbf{e}_R^*] \subseteq E, \quad (2)$$

where E^* represents the orthonormal subspace containing R eigenfaces from E which corresponds to the top R values of λ_n (square of λ_n is the same as the eigenvalues of covariance matrix of X). Thus, the low-dimensional representation of the image from P pixels to R coefficients ($R \ll P$) can be given as

$$[U]_{R \times N} = E^* \bar{X}. \quad (3)$$

2.2 Independent Component Analysis

ICA is an unsupervised learning algorithm that tries to remove higher order dependencies among the basis of natural scenes. The task is to find basis, $A = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_R]$, such that the coefficients, $S_{R \times N}$ are independent of each other and $X \approx AS$, where R is the number of basis and N is the number of images. This is also referred as factorial code representation [7].

Considering high dimensionality of X , it is often beneficial to represent data by the coefficients pertaining to the important principal components. Therefore, the input to the ICA network is U which is given in (3). ICA is performed using Infomax learning algorithm to find the independent features [8].

In this study, both ICA architectures proposed in [7] are used for recognition tasks. The input data matrix to architecture I is X^T , thereby, considering each pixel location as observation to give statistically independent basis images. In contrast, architecture II finds a factorial code for the face images giving independent features.

2.3 Non-negative Matrix Factorization

NMF is a constrained optimization problem which attempts to factorize non-negative matrix, X (pixels value of images is $\{x_{ij}\} \geq 0$), into basis matrix, $W_{P \times R} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_R]$, and features, $H_{R \times N} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_R]^T$, such that $X \approx WH$. The NMF factorization is a solution to the constrained minimization problem

$$\min_{W, H} D(X \| WH) : \quad W, H \geq 0, R \leq \frac{PN}{P+N}, \quad (4)$$

where $D(A\|B)$ is the generalized Kullback-Leibler (KL) divergence of A from B . To obtain the factors W and H multiplicative update rule of [9] can be used to optimize the problem mention in (4).

3 Supervised Feature Selection Methods

Given a set of R features, $\Phi = \{\phi_1, \phi_2, \dots, \phi_R\}$, define $\Lambda = \{\lambda(\phi_1), \dots, \lambda(\phi_R)\}$ as a set of scores given to features Φ by an operator $\lambda(\cdot)$. Feature selection methods utilize Λ to find a subset, $\Psi = \{\psi_1, \psi_2, \dots, \psi_K\} \subseteq \Phi$, of $K \leq R$ most significant features for recognition. The k -th feature in Ψ is

$$\begin{aligned}\psi_k &= \phi_j : k \leq K, k \in \mathbf{I}, \\ j &= \arg \max_r \{\Lambda - \cup_{r=1}^{r-1} \lambda(\phi_r)\} : j \leq R, j \in \mathbf{I}.\end{aligned}\quad (5)$$

where \mathbf{I} refers to integer space.

This paper uses Fisher score and MI as the feature scoring operator $\lambda(\cdot)$. These feature selection criterion are supervised as they exploit *a priori* information of the dependence of features with their classes.

3.1 Fisher Criterion

The Fisher score for a r -th feature, ϕ_r , is given as

$$\lambda_{Fisher}(\phi_r) = \frac{\sum_c (\bar{\phi}_{rc} - \bar{\phi}_r)^2}{\sum_{i,c} (\phi_{ric} - \bar{\phi}_{rc})}, \quad (6)$$

where $\bar{\phi}_{rc}$ is the mean of feature ϕ_r corresponding to class c and $\bar{\phi}_r$ is the mean of the features. Top R features with highest value of λ_{Fisher} are selected using (5) to give the feature subset, $\Psi^{Fisher} \subseteq \Phi$.

3.2 Mutual Information Criterion

Efficient selection of K features from Φ can be accomplished by maximizing the MI,

$$\max I(\Psi^{MI}; c) = I(\psi_1^{MI}, \psi_2^{MI}, \dots, \psi_K^{MI}; c) : \Psi^{MI} \subseteq \Phi. \quad (7)$$

Search of feature subset which maximizes $I(\Psi^{MI}; C)$ is computationally demanding for large value of K , hence, the problem is not-scalable. If Φ consists of statistically independent features as in case of ICA Architecture II features, the solution of (7) is given by (5) where scoring operator $\lambda(\cdot)$ operator is $I(\phi_r; c)$; thus, involving only K calculations in comparison with $R C_K$ calculation in case of dependent features.

MI between r -th feature and the class c is used as the feature selection criteria

$$\lambda_{MI}(\phi_r) = I(\phi_r; C) = \sum_c \sum_{\phi_r} p(\phi_r, c) \log \frac{p(\phi_r, c)}{p(\phi_r)p(c)}, \quad (8)$$

where $p(\cdot)$ is the probability distribution of the input random variable. Top R features with highest value of λ_{MI} are selected using (5) to give the feature subset, $\Psi^{MI} \subseteq \Phi$.

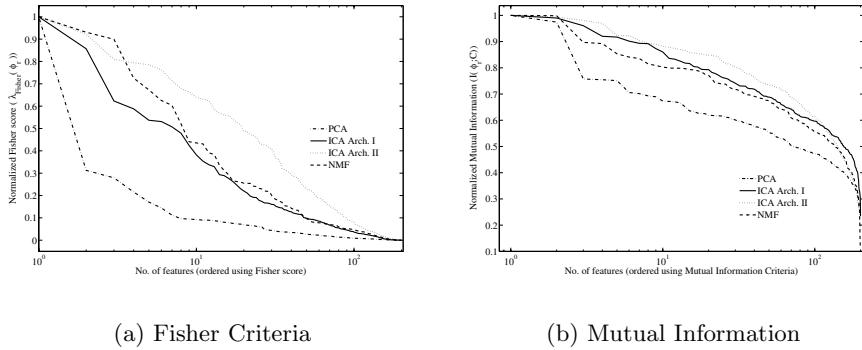


Fig. 1. Class discrimination by feature coefficient sorted by their magnitude

3.3 Hybrid Feature Selection

Feature selection methods mentioned in section (3.1–3.2) are combined to give a feature subset which further improves the discrimination between data from different classes. We define a new feature set,

$$\Psi = \{\psi_1, \psi_2, \dots, \psi_{|\Psi|}\} = \Psi^{Fisher} \cup \Psi^{MI} : K \leq |\Psi| \leq 2K, \quad (9)$$

where $|\cdot|$ is the cardinality of the input set.

The proposed feature selection criteria is a weighted sum of $\lambda_{Fisher}(\cdot)$ and $\lambda_{MI}(\cdot)$ which is defined as

$$\lambda^*(\psi_k) = \alpha \lambda_{Fisher}(\psi_k) + (1 - \alpha) \lambda_{MI}(\psi_k). \quad (10)$$

By using the value of $\lambda^*(\psi_k)$, one can select top R features using (5) to define a subset $\Psi^* \subseteq \Psi \subseteq \Phi$, where $|\Psi^*| = K$. In this paper, α and β are given values from 0.1 to unity in steps of 0.1.

4 Experimental Results

4.1 Experimental Setup

To evaluate the performance of feature selection methods mentioned in sec. 3, face recognition experiments were performed on subset of FERET database [10][11] similar to the test sets used in [7]. The training data set consists of 423 frontal view face images with neutral and other facial expressions taken on the same day. All 423 images are of different individuals. Face recognition experiments were performed on the following test sets: a) Test Set I: 423 frontal view face images of individuals in training set with different expressions taken on same day, b) Test Set II: 59 frontal view face images of individuals in training set with same expressions but taken on different day (upto two years after taking images of training data set), and c) Test Set III: 59 frontal view face images of individuals in training set with different expressions and taken on different

Table 1. Comparison of average improvement in recognition performance (ref. to equation 11) when feature are selected using methods discussed in sec. 3 over no feature selection. The maximum recognition performance using subset of selected features is also given in (). All the values in the table are in %.

Test Set	Feature Selection Method	PCA	ICA Arch. I	ICA Arch. II	NMF
Test Set I	Hybrid	-3.89 (92.67)	4.42 (93.14)	13.76 (92.19)	10.01 (93.38)
	MI	-5.20 (92.67)	3.59 (93.14)	8.07 (91.72)	9.62 (93.14)
	Fisher Criteria	-7.03 (92.43)	-1.04 (92.43)	1.81 (91.96)	-0.64 (91.25)
	Variance	0.00 (92.43)	-3.80 (91.72)	1.48 (91.72)	-9.64 (91.72)
Test Set II	Hybrid	-1.96 (71.18)	13.38 (69.49)	30.31 (69.49)	19.26 (66.10)
	MI	-6.59 (71.18)	8.50 (67.79)	19.33 (67.79)	9.22 (64.40)
	Fisher Criteria	-16.55 (69.49)	-3.62 (66.11)	2.86 (67.79)	-9.22 (61.01)
	Variance	0.00 (69.49)	-11.93 (67.79)	5.73 (67.79)	7.79 (62.71)
Test Set III	Hybrid	-2.80 (62.71)	3.83 (59.32)	39.63 (59.32)	27.93 (49.15)
	MI	-5.44 (62.71)	-4.23 (59.32)	23.53 (57.62)	12.50 (49.15)
	Fisher Criteria	-18.62 (61.02)	-13.51 (59.32)	23.22 (57.62)	4.79 (47.45)
	Variance	0.00 (61.02)	-25.20 (59.32)	23.22 (57.62)	-5.85 (47.45)

day. The location of eyes and mouth provided by FERET database were used to align the face images. The aligned images were histogram equalized and have a resolution of 60 x 50 pixels at gray-scale level of 256.

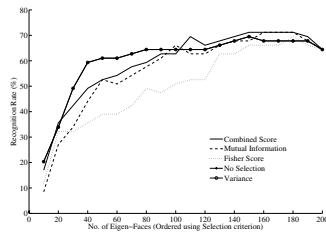
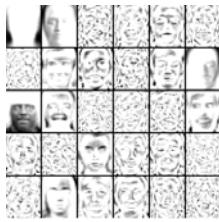
In this study, 1-nearest neighbor is used as the classifier with cosine similarity as the distance metric between training and test features.

4.2 Experimental Results

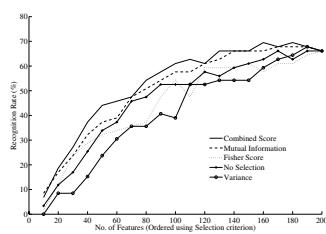
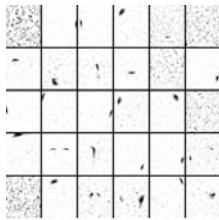
Figure 1 shows the class discriminability by the coefficients of the low dimensional unsupervised features mentioned in sec. 2. Table 11 shows the average improvement in recognition performance when feature are selected using methods discussed in sec. 3 over no feature selection. The average improvement in recognition performance, ϵ , is given as

$$\epsilon = \frac{\sum_{k=1}^{|\Phi|} (\delta'(\Psi_k^*) - \delta(\Phi_k^*))}{\sum_{k=1}^{|\Phi|} \delta(\Phi_k^*)}, \quad \Psi_k^* = \psi_k \cup \Psi_{k-1}^*, \quad \Phi_k^* = \phi_k \cup \Phi_{k-1}^*, \quad 1 \leq k \leq |\Phi|, \quad (11)$$

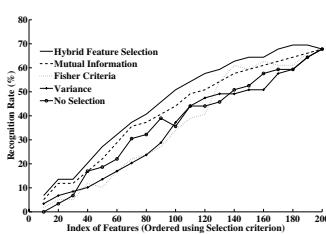
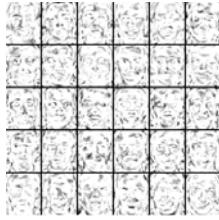
where δ' and δ are the recognition performance with and without feature selection, respectively. Table 11 also shows the maximum recognition obtained using subset of selected features from the low-dimensional feature space. The left panel of fig. 2 shows top 30 basis images of unsupervised feature extraction methods arranged in the decreasing order of their $I(\phi_r; c)$. Right panel of fig. 2 compares the performance of feature selection methods discussed in sec. 3. Figure 2 also shows the recognition performance when no feature selection is done and when features are selected in unsupervised manner using their variance.



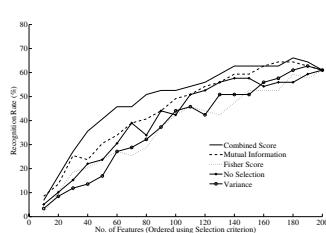
(a) PCA



(b) ICA Architecture I



(c) ICA Architecture II



(d) NMF

Fig. 2. Left panel shows the basis images and right panel compares the performance of feature selection criteria for a linear low-dimensional feature space

5 Conclusions

Recognition tasks with features selected using the proposed hybrid feature selection method results in superior performance over other feature selection techniques discussed in sec. 3. The hybrid feature selector and feature-class MI criterion are computationally optimal when applied to statistically independent ICA architecture II features. In addition, highest average improvement in recognition performance is seen when mutually independent ICA Architecture II features are selected using the proposed hybrid feature selector. Unsupervised selection of orthogonal PCA features using variance gives better average improvement in recognition performance, ϵ , compared to other feature selection methods, however, it fails to give maximal recognition which is achieved by using hybrid feature selector as seen from table 1.

Acknowledgement

This research was supported as a Brain Neuroinformatics Research Program by Korean Ministry of Commerce, Industry and Energy. Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office. Authors would like to thank Dr. Marian S. Bartlett for providing data labels of test sets used in [7].

References

1. Hinton, G., Sejnowski, T.: *Unsupervised learning: Foundations of Nuerla Computation*. MIT Press, Cambridge (1999)
2. Hyvarinen, A., et al.: *Independent Component Analysis*. John Wiley & sons, Inc., Chichester (2001)
3. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788–791 (1999)
4. Wang, G., et al.: Feature selection with conditional mutual information MaxMin in text categorization. In: Proc. Int. Conf. on Information and Knowledge Management, pp. 342–349 (2004)
5. Su, H., et al.: Face Recognition Method Using Mutual Information and Hybrid Feature. In: Proc. Int. Conf. on Computational Intelligence and Multimedia Applications, pp. 436–440 (2003)
6. Dhir, C.S., et al.: Efficient feature selection based on information gain criterion for face recognition. In: IEEE Int. Conf. on Information Acquisition, pp. 523–527 (2007)
7. Bartlett, M.S., et al.: Face recognition by independent component analysis. *IEEE Transactions on Neural Networks* 13(6), 1450–1462 (2002)
8. Bell, A.J., et al.: An information-maximization approach to blind separation and blind deconvolution. *Neural Computation* 7, 1129–1159 (1995)
9. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. In: *Advances in Neural Information Processing Systems*, pp. 556–562. MIT Press, Cambridge (2000)
10. Philips, P.J., et al.: The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing* 16(5), 295–306 (1998)
11. Philips, P.J., et al.: The FERET Evaluation Methodology for Face Recognition Algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22, 1090–1104 (2000)

Sensibility-Aware Image Retrieval Using Computationally Learned Bases: RIM, JPG, J2K, and Their Mixtures

Takatoshi Kato¹, Shun'ichi Honma^{1,2}, Yasuo Matsuyama¹,
Tetsuma Yoshino¹, and Yuuki Hoshino¹

¹Department of Computer Science and Engineering,
Waseda University, Tokyo 169-8555, Japan

²SONY, Tokyo 108-0075, Japan
{tkato,shunichi1029,yasuo,tetsuma-y,yu-ki_hs}
@wiz.cs.waseda.ac.jp
<http://www.wiz.cs.waseda.ac.jp/index-e.html>

Abstract. Sensibility-aware image retrieval methods are presented and their performances are compared. Three systems are discussed in this paper: PCA/ICA-based method called RIM (Retrieval-aware IMage format), JPEG, and JPEG2000. In each case, a query is an image per se. Similar images are retrieved to this query. The RIM method is judged to be the best settlement in view of the retrieval performance and the response speed according a carefully designed set of opinion tests. An integrated retrieval system for image collections from the network and databases which contain RIM, JPEG and JPEG2000 is realized and evaluated lastly. Source codes of the RIM method is opened.

1 Introduction

The sensibility, or *kansei*, is an essential cry from human clients as a lubricant in the contemporary dry network society. Many studies have been reported towards the realization of human-friendly tools. But, plain strategies usually lead to demanding turnaround time because of their ad hoc natures. By reflecting this, we address the human-aware image retrieval based upon computational intelligence methods, especially by “learning from data.” All images are handled in their compressed domains. Addressed problems are as follows:

- (1) To present image compression methods based on PCA (Principal Component Analysis) and ICA (Independent Component Analysis).
- (2) To show the coding system using such learned bases. This format is called RIM (Retrieval aware IMage format).
- (3) To compare the RIM method with JPEG and JPEG2000. Obtained results can be previewed as follows.
 - (a) On the data compression, the performance is
 $\text{JPEG2000} \gtrsim \text{RIM} \gtrsim \text{JPEG}$.

- (b) On the similar image retrieval, the performance is RIM \gtrsim JPEG \gtrsim JPEG2000.
 - (c) The retrieval speed constrained by the performance is RIM \gg JPEG $\gg\gg$ JPEG2000.
 - (d) By considering the items (a) \sim (c), one finds that the RIM which is based on the learning from data is a viable method on the joint data compression and retrieval of images reflecting the human sensibility.
- (4) A similar-image retrieval system for mixtures of {RIM, JPEG, JPEG2000} is realized and tested.

In order to guide readers to grasp the addressed problem, we give Figure 1 in advance which conveys the purport of this paper. As can be understood from this figure, a query to an image collection or a database is an image per se. There are three types of image expressions, RIM, JPEG, and JPEG2000. Installed systems find similar images as is shown in this illustration. Images marked by a circle are correct ones according to tested persons' opinions. The third image may or may not be judged similar depending on the tested person. The sensibility or the similarity thus differ personally. The RIM method will be found the best from the viewpoint of the joint retrieval performance and the speed.

2 Image Compression Using Learned Bases

This paper's joint image compression and retrieval by RIM uses bases learned from source data. The steps are as follows: (1) sampling, (2) average separation, (3) average quantization, (4) entropy coding on the average, (5) PCA bases learning, (5') ICA bases learning, (6-1) bases normalization, (6-2) entropy coding on bases (7-1) coefficients calculation, (7-2) coefficients quantization, (7-3)

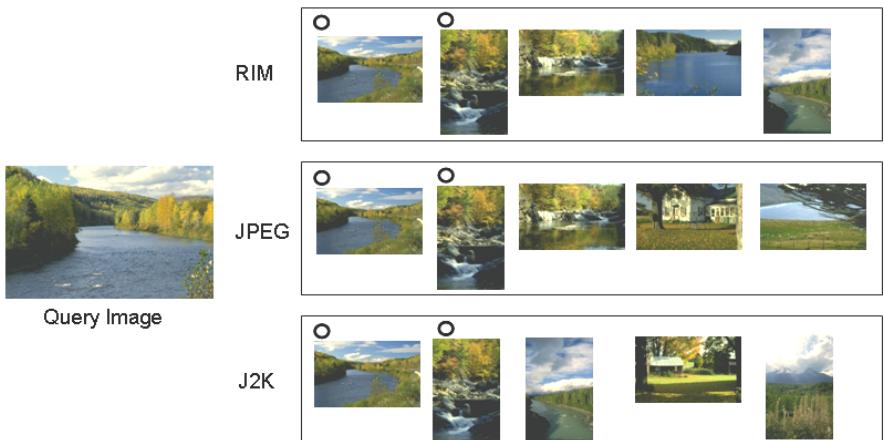
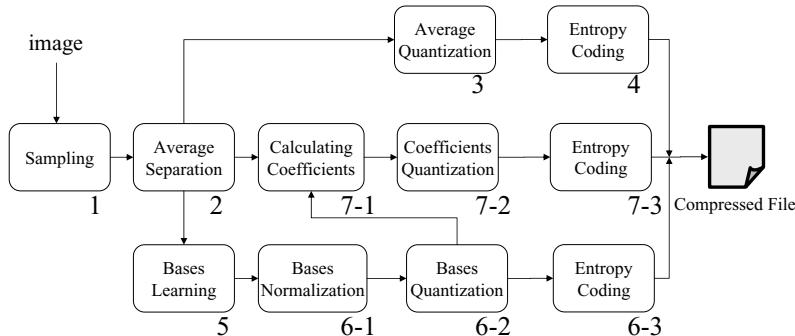
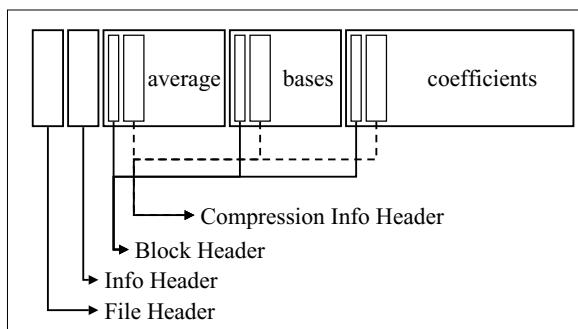


Fig. 1. Sensibility-aware image retrieval in three types of coded domains

**Fig. 2.** Image encoding using learned bases**Fig. 3.** RIM

entropy coding on coefficients. Figure 2 illustrates the flow of the encoding which is based on the learned bases. The above numbering matches to block numbers appearing in this figure.

It is necessary to pack encoded quantities efficiently. That is, an effective format needs to be defined. Figure 3 illustrates such a format called RIM [1]. This format contains headers. But, this overhead is absorbed within the margin of the compression performance advantage over JPEG. The item (3)(a) of Section 1 was obtained by a preliminary experiment for this paper.

3 Similarity Measures for RIM, JPEG and JPEG2000

The goal of this paper is rephrased by the following items.

- (a) Retrieval performance improvement of RIM
- (b) Comparison of three methods, RIM, JPEG and JPEG2000, on the joint performance on the similar-image retrieval and its speed.
- (c) Checking to see if the sensibility-aware image retrieval is possible in format-mixed environments. This is in the compressed domains

It is important to understand that the selection of similarity measures depends on each image format. This is because image format conversions are avoided in

the process of the image retrieval. Defining characteristics of three compression methods can be reviewed as follows.

RIM: There are two types, PCA-based and ICA-based. The format contains average colors, bases and coefficients which can be decoded separately.

JPEG: Fixed DCT bases are used. There are DC and AC components.

JPEG2000: Fixed wavelet transformation is used. Multiple subband images are encoded.

Table 1. Features extracted from each format

features	RIM	JPEG	JPEG2000
colors	average colors	DC components	lower frequencies
edges	average colors	DC components	lower frequencies
textures	bases	AC components	higher frequencies

Table 1 lists up extractable information from coded components. Quantities appeared in this table are utilized in various similarity definitions. Color and edge features can be utilized commonly to RIM, JPEG and JPEG2000. On the other hand, the texture information contributes to the similarity measure strongly depending on individual methods. The following summarizes sub-similarity measures S_{color} , S_{edge} and $S_{texture}$ which form the total similarity measure between two images:

$$S_{total} = b\{aS_{color} + (1 - a)S_{edge}\} + (1 - b)S_{texture}, \quad \{a, b\} \in [0, 1]^2. \quad (1)$$

Details of the sub-similarity measure are as follows.

- (1) Color similarity S_{color} : We test two types of color similarity measures.
 - (a) Color structure descriptor(CSD) [2]
 - (b) Auto-color-correlogram (ACG) [3]
- (2) Edge similarity S_{edge} : We use the edge histogram descriptor (EHD) of MPEG7 [2]. This feature was newly added in this paper.
- (3) Texture similarity $S_{texture}$: This similarity measure depends on the image compression methods RIM, JPEG and JPEG2000.
 - (3-1) For RIM: Either PCA or ICA basis set is used [1]. The similarity computation uses a weighted summation of inner products of bases [4].
 - (3-2) For JPEG: Variance of AC coefficients (AC) obtained by a patch-wise zigzag scanning of images [5] is used.
 - (3-3) For JPEG2000:
 - (a) Variance of subband coefficients (VSC): This method uses variance of subband information stored in bins of *color_component × directions × resolution_levels* [6].
 - (b) Wavelet correlogram (WCG): This method uses the correlogram on higher subband frequencies [7].
 - (c) Generalized Gaussian density distance (GGD): This method uses estimated higher frequency distribution by generalized Gaussian densities. Distances are measured by the K-L divergence [8].

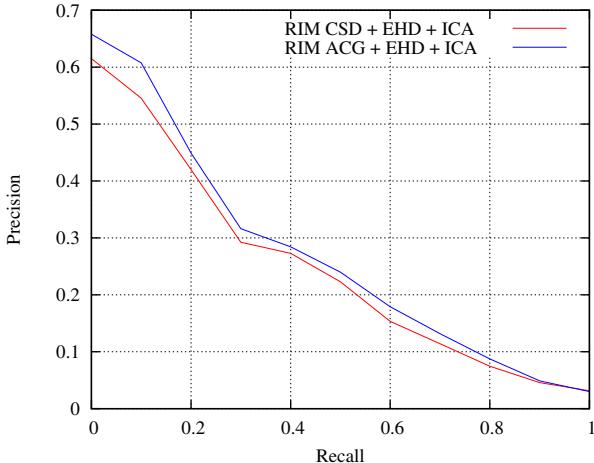


Fig. 4. Precision-recall curves for RIM via ICA

4 Similar-Image Retrieval Performance

4.1 Opinion Test Design

The sensibility is different individually. So is the human judgment on the image similarity as can be understood from Figure 1. Therefore, a set of well-designed opinion tests is essential for the choice of a viable similar-image retrieval system. The following is the description of the database generation for the opinion test.

The groundtruth of 5,200 images was generated from a 20,000 image set of 52 categories in 5 themes [9]. Then, 100 query images (20 images form each theme) were chosen randomly. Each of 20 subjects (opinion test persons) checks to see 5,200 images for a given query. For each query, 5 ~ 20 similar images were manually selected as correct ones in advance. Thus, the judgment on the similarity is strongly *human sensibility dependent*. The average number of correct images is only 0.2% of the total groundtruth.

4.2 Compatibility with Human Sensibility

Retrieval performances were measured as precision-recall curves. We tested all combinations of sub-similarity measures on {RIM, JPEG, JPEG2000} appeared in Section 3. This was a set of very demanding tasks both to human subjects and machines. Due to the space limitation, we give precision-recall curves only for the RIM via ICA+EHD+{CSD, ACG} as Figure 4. Here, Recall=1 means that all similar images are hit regardless of the subjects' sensibility. This can not occur since the sensibility or the similarity judgment strongly depends on the personality. Therefore, $0 \leqslant \text{Recall} \leqslant 0.2$ is the range of interest. It is also necessary to emphasize that $\text{Precision} \geqslant 0.5$ is a high performance region because of the variety of the subjects' judgments.

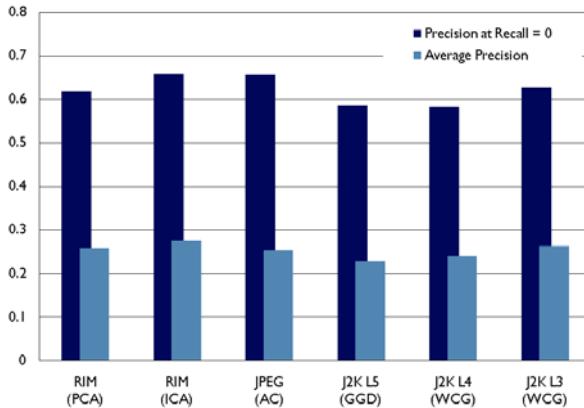


Fig. 5. Image retrieval performances for RIM, JPEG and JPEG2000

Instead of giving all combinations of sub-similarity measures for RIM, JPEG, JPEG2000, we show representing and average performances for all formats as Figure 5. One might think that average values are low. But, the averages are computed from the whole recall range (see Figure 4). From Figure 5, one finds that the RIM by ICA is the best and the JPEG by AC follows well. The JPEG2000 by WGG of the level 3 and the RIM by PCA follows as the third. Therefore, we can claim the RIM method is the most creditable from the compatibility with the human sensibility. But, there will be a further merit on the RIM. It is the speed which is a crucial factor for the retrieval. This will be shown in the next subsection.

4.3 Joint Performance of the Retrieval and Speed

The speed is an important factor ranked with the retrieval correctness. By considering this, we computed the joint performance of the average precision and speed. Figure 6 illustrates this result. In this figure, “XXX (YYY)” means that the method is “XXX” (J2K stands for JPEG2000), and the applied sub-similarity is “YYY.” The marks {diamond, square, triangle} stand for the similarity measures {color, edge, texture}. Positions of dots are obtained by adjusting parameters to show the best retrieval result. From this set of experiments, one finds the following.

- (a) RIM is speedy. Its color similarity is especially good from the joint viewpoint of the retrieval and speed.
- (b) JPEG shows creditable retrieval performance in each sub-similarity measure. But, its retrieval time requires 3~4 times more than RIM.
- (c) JPEG2000 shows creditable retrieval performance in color and texture sub-similarity measures. But, its color similarity computation requires 3~4 times more than RIM. Moreover, its texture similarity computation demands 10~20 times more than RIM.

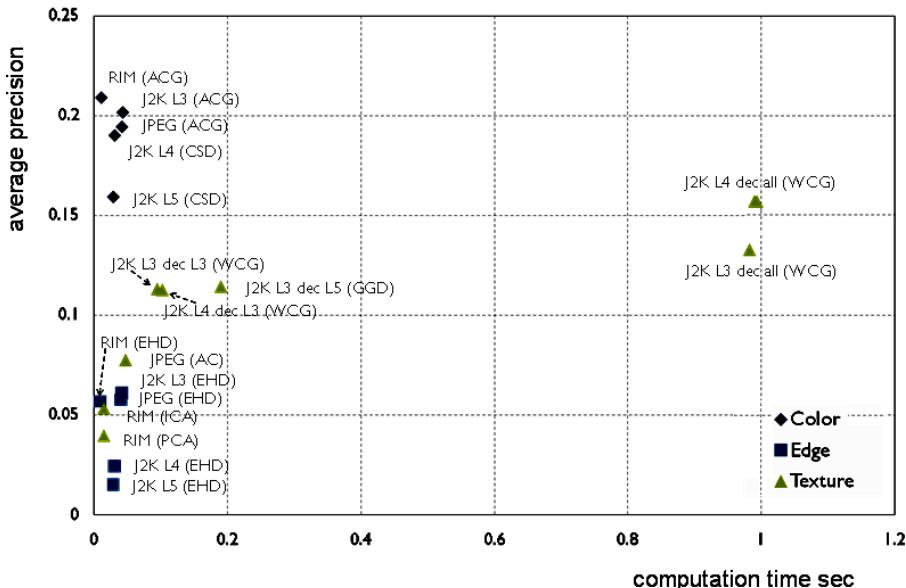


Fig. 6. Joint retrieval and speed performance

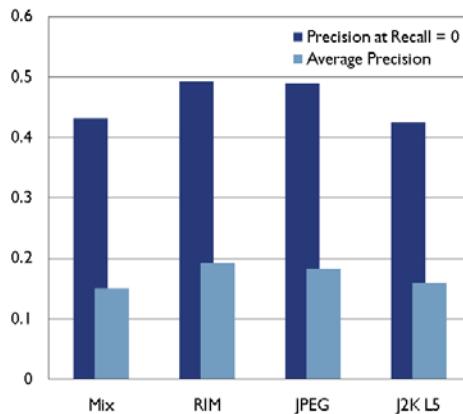


Fig. 7. Conventional all-in-one system

- (d) Considering the joint performance of the human sensibility awareness with respect to the similar image retrieval and its speed, RIM is judged to outperform JPEG and JPEG2000.

5 Concluding Remarks

In this paper, similar-image retrieval systems are generated and tested on the compatibility with the human sensibility. There were three formats tested: RIM

using learned PCA/ICA bases, JPEG using DCT bases, and JPEG2000 using wavelets. It was shown that RIM is the best from the viewpoints of the retrieval performance and the speed.

Since we generated similar-image retrieval systems on {RIM, JPEG, JPEG2000}, it has become possible to realize an all-in-one system which can work on mixed images of {RIM, JPEG, JPEG2000}. By reviewing Figure 6, however, the full integration of {RIM, JPEG, JPEG2000} suffers from the retrieval speed; even for putting the opinion tests together. Therefore, we generated a conventional all-in-one system for {RIM(ACG), JPEG(DC), JPEG2000(L5)}. Figure 7 illustrates the result on {RIM, JPEG, JPEG2000} with the same percentage mixture. This is another viable system besides the RIM of Figure 3 since any format conversions among {RIM, JPEG, JPEG2000} are unnecessary.

This paper's version-up of the full RIM method with its viewer called Wisvi (Waseda image search viewer) is open and can be downloaded from the authors' web site given in the first page of this paper.

References

1. Katsumata, N., Matsuyama, Y., Chikagawa, T., Ohashi, F., Horiike, F., Honma, S., Nakamura, T.: Retrieval-aware image compression, its format and viewer based upon learned bases. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4233, pp. 420–429. Springer, Heidelberg (2006)
2. Martines, J.M.(ed.): ISO/IEC JTC1/SC29/WG11 Coding of moving pictures and audio: MPEG-7 overview, N6828 (2004)
3. Huang, J., Kumar, S.R., Mitra, M., Shu, W.-J., Zabih, R.: Image indexing using color correlograms. In: Proc. IEEE Comp. Soc. Conference on Visual and Pattern Recognition, pp. 762–768 (1997)
4. Katsumata, N., Matsuyama, Y.: Database retrieval for similar images using ICA and PCA bases. Engineering Applications of Artificial Intelligence 18, 705–717 (2005)
5. Mandal, M.K., Idris, F., Panchanathan, S.: A critical evaluation of image and video indexing techniques in the compressed domain. Image and Vision Computing 17, 513–529 (1999)
6. Mallat, S.G.: A theory of multiresolution signal decomposition: The wavelet representation. IEEE Trans. Pattern Anal. and Machine Intelligence 11, 674–693 (1989)
7. Abrishami Moghaddam, H., Taghizadeh Khajoie, T., Rouhi, A.H.: Wavelet correlogram: A new approach for image indexing and retrieval. Pattern Recognition 38, 2506–2518 (2005)
8. Do, M.N., Vetterli, M.: Wavelet-based texture retrieval using generalized Gaussian demsity and Kullback-Leibler divergence. IEEE Trans. Image Processing 11, 146–158 (2002)
9. Corel Stock Photography 1 (1993)

An Analysis of Generalization Error in Relevant Subtask Learning

Keisuke Yamazaki¹ and Samuel Kaski²

¹ Precision and Intelligence Laboratory, Tokyo Institute of Technology
R2-5, 4259 Nagatsuta, Midori-ku, Yokohama, 226-8503 Japan
k-yam@pi.titech.ac.jp

² Department of Information and Computer Science,
Helsinki University of Technology
P.O. Box 5400, 02015 TKK, Finland
samuel.kaski@tkk.fi

Abstract. A recent variant of multi-task learning uses the other tasks to help in learning a task-of-interest, for which there is too little training data. The task can be classification, prediction, or density estimation. The problem is that only some of the data of the other tasks are relevant or representative for the task-of-interest. It has been experimentally demonstrated that a generative model works well in this *relevant subtask learning* task. In this paper we analyze the generalization error of the model, to show that it is smaller than in standard alternatives, and to point out connections to semi-supervised learning, multi-task learning, and active learning or covariate shift.

1 Introduction

Lack of a sufficient amount of training data is a recurring problem in practical data analysis and machine learning studies. In bioinformatics, for instance in microarray data analysis, this has been referred to as the large p , small n problem: It is hard to learn classifier or regression models when the dimensionality of the data samples p is large and the number of samples n is small. Recommender systems and user modeling share an analogous problem: it would be useful to give good predictions or recommendations already when only a few observations of the user's behavior have been made.

The problem can be alleviated by regularizing the predictors, and by including more prior knowledge to the model structure or in the prior distribution of the parameters. Collecting new data helps too. Several machine learning scenarios have been developed to help in cases where these straightforward alternatives are not applicable or available. For instance, in semisupervised learning a classification task where labeled data is the scarce resource, can be aided by including non-labeled data from the same distribution. In multi-task learning several tasks are learned together, in the hope that the tasks share properties which help in learning each task. Common to these scenarios is that they try to incorporate more data into the training data set.

A central practical problem in adding more data is that most models assume all training data to be “relevant”; typically the implicit assumption is that all training data come from the same distribution, or at least that adding the data to the learning set improves the performance. Requiring all data to come from the same distribution is a strong assumption, and if it could be relaxed, it would be possible to include data sets or tasks containing only partly relevant data. Useful sets abound in genomic databanks and measurement databases in bioinformatics, for instance, or data about other users or products in recommender systems.

Relevant subtask learning is a recent variant of multi-task learning, where the assumption of representative data is relaxed by assuming that the learning data is a mixture of relevant and irrelevant samples. The setup is that there is one task-of-interest, which is special in that the test data is known to come from its distribution. In the other tasks, some samples come from the same distribution and some not, and it is naturally not known which. In other words, the other tasks are contaminated by data from irrelevant subtasks, and we would only like to use the data from the relevant subtasks. The relevant subtask model (RSM) [1] builds a classifier for the task-of-interest under these assumptions; we will generalize this setup to unsupervised learning, which can naturally model classes as well. RSM has been empirically demonstrated to outperform standard multi-task learning and the straightforward alternatives of learning the task-of-interest separately and pooling all data together.

From the statistical point of view good performance of RSM is not trivial since the increased complexity of the learning model increases the generalization error. In this paper we derive the asymptotic generalization error for maximum likelihood estimates. Comparing the generalization error of alternative models we prove that RSM is still better than the others.

2 Relevant Subtask Learning

We start by formulating the relevant subtask learning problem, showing its connections to multi-task learning, active learning, and semi-supervised learning.

Relevant subtask learning is a variant of multi-task learning. In multi-task learning [2][3][4] there are several classification tasks, and the question is whether solving the problems together improves performance compared to solving each separately. In practice, the tasks are different data sets. More generally, instead of classification problems the tasks could be other statistical modeling tasks such as regression, clustering or density estimation; in all cases the research problem is to learn a good estimator for each task, transferring information between the tasks as needed.

In relevant subtask learning the setup is asymmetric. One of the tasks is a target task, the “*task-of-interest*”, and the research problem is whether the other tasks can be used to help in better solving the target task. In practice the tasks are data sets, and the goal is to find more data to complement the scarce learning data in the task-of-interest. The obvious problem is that not all data in the other tasks are relevant in the sense of coming from the same distribution or

at least helping in learning the task-of-interest. Relevant subtask learning makes the assumption that each task is a mixture of relevant and irrelevant data, that is, each task is a combination of a relevant subtask and an irrelevant subtask.

In this paper we will consider the unsupervised setting of density estimation, and two tasks without loss of generality: the task of interest (task number 1) and a supplementary task (task number 2). Let us denote the distribution of data in the task-of-interest by $q_1(x) = q_{01}(x)$, where the datum $x \in R^M$. The data of the irrelevant subtask within task 2 follow the distribution $q_{02}(x)$.

For learning we are given two training data sets D_1 and D_2 , one for each task, where $\#D_1 = \alpha n$ and $\#D_2 = (1 - \alpha)n$ for $0 < \alpha < 1$. Here αn of course needs to be a natural number. D_1 is known to be sampled from $q_{01}(x)$, but generally $\#D_1$ is too small for learning an adequate model for q_{01} .

The supplementary data set D_2 contains samples from both the same distribution as D_1 , and from the irrelevant subtask. Hence the density is a mixture,

$$q_2(x) = c^* q_{01}(x) + (1 - c^*) q_{02}(x),$$

where $0 < c^* < 1$ is a constant. All quantities above, the q and c^* , are unknown to us. Then, we formally rewrite $D_2 = D_{21} + D_{22}$, where D_{21} and D_{22} have been sampled from q_{01} and q_{02} , respectively. Obviously, the data in D_2 do not have a label showing which distribution they come from, q_{01} or q_{02} . The data in D_1 have the label since they are all from q_{01} . This situation is analogous to semi-supervised learning [5], in which a small labeled training data set of a classifier is complemented with additional unlabeled data. The difference is that in RSL the goal is not to classify samples according to the labels but instead to build a good model for $q_1(x)$ based on D_1 and D_{21} . In this sense relevant subtask learning even resembles “one-class classification” [6, 7].

The test data set D_t is known to come from $q_{01}(x)$. This fundamentally separates RSL from standard multitask learning; we are only interested in the target task. As a result, since we use both data sets D_1 and D_2 for training, the test distribution is different from the training distribution. Moreover, we can change the training distribution by selecting the data for training. This scenario is similar to active learning [8], which focuses on doing the active selection, and the covariate shift [9], where the research interest is in studying the effects of the difference in the training and testing distributions.

Let us assume that our learning model can attain the true distributions which generate the training data. In other words, we prepare models $p_{01}(x|a) = p_1(x|a)$ and $p_{02}(x|b)$, where $a \in R^d$ and $b \in R^d$ are the parameters, respectively. Then the assumption translates to the following: there exist true parameters a^* and b^* such that

$$q_{01}(x) = p_{01}(x|a^*), \quad q_{02}(x) = p_{02}(x|b^*).$$

Let us define a mixture model for modeling D_2 :

$$p_2(x|w) = cp_{01}(x|a) + (1 - c)p_{02}(x|b),$$

where $w = \{a, b, c\}$. More precisely, $a = (w_1, \dots, w_d)$, $b = (w_{d+1}, \dots, w_{2d})$, and $c = w_{2d+1}$.

With these definitions we can formulate the possible solutions to the learning problem more precisely. Table II summarizes the methods. We will use maximum likelihood learning in all cases.

(M1) Single-task learning. The first and the simplest approach is to use only data from the task of interest, discarding the supplementary data altogether.

- Training model $p_1(x|a)$, training data D_1
- Predictive model $p_1(x|\hat{a}_1)$, where

$$\hat{a}_1 = \arg \max_a \sum_{x_i \in D_1} \ln p_1(x_i|a)$$

- True parameters a^* .

Here the “training model” is the one applied to training data to learn the parameters, and the “prediction model” is applied to the test data. In the simple M1 they are the same, but later will differ. The predictive model is always a part of the training model.

(M2) Learning only from the supplementary task. A slightly artificial choice is to discard D_1 altogether and learn only from D_2 .

- Training model $p_2(x|w)$, training data D_2
- Predictive model $p_{01}(x|s(\hat{w}_2))$, where

$$\hat{w}_2 = \arg \max_w \sum_{x_i \in D_2} \ln p_2(x_i|w)$$

- True parameters $w_2^* = (a^*, b^*, c^*)$.

Here the $s(\cdot)$ is a function that chooses the parameters corresponding to the target task from w , namely $s(w) = a$. Estimation of s is discussed below. The model M2 is expected to be better than M1 when $\#D_1$ is too small and the estimation using D_1 is not reliable.

(M3) Relevant subtask model. M3 corresponds to the RSM of [II](#).

- Training model: $p_1(x|a)$ for D_1 and $p_2(x|w)$ for D_2 . The combined learning data is denoted by $D_1 \oplus D_2$.
- Predictive model $p_{01}(x|s(\hat{w}_3))$, where

$$\hat{w}_3 = \arg \max_w \left\{ \sum_{x_i \in D_1} \ln p_1(x_i|a) + \sum_{x_i \in D_2} \ln p_2(x_i|w) \right\}$$

- True parameters $w_3^* = (a^*, b^*, c^*)$.

Note that the complexity of the model is more than M1. You can see the dimension of the parameters increases from d to $2d + 1$, which could cause worse generalization.

Table 1. Data structures and learning models

method	data	training model	prediction model
M1	D_1	p_1	p_1
M2	D_2	p_2	$\min(p_{01}, p_{02})$
M3	$D_1 \oplus D_2$	$p_1 \oplus p_2$	p_{01}
M4	$D_1 + D_2$	p_2	$\min(p_{01}, p_{02})$

(M4) Pooled data model. The simplest approach, optimal when data of all tasks come from the same distribution, is to pool all data and estimate a single model. Since we know the data may be a mixture of relevant and irrelevant data, we will learn the mixture model.

- Training model: $p_2(x|w)$ for $D_1 + D_2$
- Predictive model $p_{01}(x|s(\hat{w}_4))$, where

$$\hat{w}_4 = \arg \max_w \sum_{x_i \in D_1 + D_2} \ln p_2(x_i|w)$$

- True parameters

$$w_4^* = (a^*, b^*, \alpha + (1 - \alpha)c^*).$$

Knowing which of the tasks is the task-of-interest is naturally central in relevant subtask learning. In general, the identity of the tasks in multitask learning can be represented as a *task label*; the model M4 is naive in not utilizing the knowledge of which task label the test data corresponds to. So M4 is the only option when the task label is not available, such as in typical applications of mixtures of experts [10, 11, 12].

Note that $D_1 \oplus D_2$ is the combined data set with the task label and $D_1 + D_2$ is the merged data set, where we cannot distinguish the difference between the original data sets any more.

The models M2 and M4 are unidentifiable in the sense that the model does not determine which mixture component corresponds to the task of interest and hence to the test data. In practice, we can select the proper one by comparing the likelihoods of the test data D_t ,

$$\sum_{x_j \in D_t} \ln p_{01}(x_j|s(\hat{w}_i)), \quad \sum_{x_j \in D_t} \ln p_{02}(x_j|s(\hat{w}_i))$$

for $i = 2, 4$. In this paper, the analysis assumes that the proper component p_{01} has been selected, which means that the derived expression shows the minimum error. In Table 1 the notation ‘ $\min(p_{01}, p_{02})$ ’ denotes this selection. The assumption is reasonable asymptotically, in the limit of a large test data set.

3 Analysis of the Generalization Error

Let us define the generalization error of the learning methods by

$$G_i(n) = E_{D_1 \oplus D_2} \left[\int q_1(x) \ln \frac{q_1(x)}{p_{01}(x|s(\hat{w}_i))} \right],$$

where $E_{D_1 \oplus D_2}[\cdot]$ denotes the expectation over the training data D_1 and D_2 , the suffix i stands for the error of the learning method M_i , and we define $s(\hat{w}_1) = \hat{a}_1$ to simplify the notation. The following theorem is the main contribution of this paper:

Theorem 1. *The generalization error has the asymptotic form*

$$\begin{aligned} G_1(n) &= \frac{d}{2\alpha n} + O\left(\frac{1}{n^2}\right), \\ G_2(n) &= \frac{1}{2(1-\alpha)n} \text{Tr}[I(a^*)J(w_2^*)^{-1}] + O\left(\frac{1}{n^2}\right), \\ G_3(n) &= \frac{1}{2n} \text{Tr}[I(a^*)K(w_3^*)^{-1}] + O\left(\frac{1}{n^2}\right), \\ G_4(n) &= \frac{1}{2n} \text{Tr}[I(a^*)J(w_4^*)^{-1}] + O\left(\frac{1}{n^2}\right), \end{aligned}$$

where the I, J, K are $(2d+1) \times (2d+1)$ -dimensional matrices where

$$\begin{aligned} I(a^*)_{ij} &= \int \frac{\partial \ln p_1(x|a^*)}{\partial w_i} \frac{\partial \ln p_1(x|a^*)}{\partial w_j} q_1(x) dx, \\ J(w^*)_{ij} &= \int \frac{\partial \ln p_2(x|w^*)}{\partial w_i} \frac{\partial \ln p_2(x|w^*)}{\partial w_j} q_2(x) dx, \\ K(w^*) &= \alpha I(a^*) + (1-\alpha) J(w^*) \end{aligned}$$

for $a^* = s(w^*) \subset w^*$.

As a reminder, d is the number of parameters in p_{01} , n is the total number of data samples, and α is the proportion of samples in the task-of-interest data $\#D_1$. Note that $I(a^*)$ has non-zero elements in the top-left $d \times d$ submatrix.

Sketch of proof: Consider the most general case M3. We can easily extend it to the other cases. Based on the Taylor expansion around w_3^* , the generalization error can be expressed as

$$\begin{aligned} G_3(n) &= \frac{1}{2} \sum_{i,j=2}^{2d+1} E_{D_1 \oplus D_2} \left[(\hat{w}_3 - w_3^*)_i (\hat{w}_3 - w_3^*)_j \right] \\ &\quad \times \int \frac{\partial \ln p_1(x|s(w_3^*))}{\partial w_i} \frac{\partial \ln p_1(x|s(w_3^*))}{\partial w_j} q_1(x) dx. \end{aligned}$$

Let us define the log-likelihood

$$L_3(w, D_1 \oplus D_2) = \sum_{k=1}^{\alpha n} \ln p_{01}(x_k|s(w)) + \sum_{k=\alpha n+1}^n \ln p_2(x_k|w)$$

Since $\partial L_3(\hat{w}_3, D_1 \oplus D_2) / \partial w = 0$,

$$\begin{aligned} & \frac{1}{n} \frac{\partial L_3(\hat{w}_3, D_1 \oplus D_2)}{\partial w} = 0 \\ &= \frac{1}{n} \sum_{k=1}^{\alpha n} \frac{\partial \ln p_{01}(x_k | s(w_3^*))}{\partial w} + \frac{1}{n} \sum_{k=1}^{\alpha n} \frac{\partial^2 \ln p_{01}(x_k | s(w_3^*))}{\partial w_i \partial w_j} (\hat{w}_3 - w_3^*) \\ &+ \frac{1}{n} \sum_{k=\alpha n+1}^n \frac{\partial \ln p_2(x_k | w_3^*)}{\partial w} + \frac{1}{n} \sum_{k=\alpha n+1}^n \frac{\partial^2 \ln p_2(x_k | w_3^*)}{\partial w_i \partial w_j} (\hat{w}_3 - w_3^*). \end{aligned}$$

Using the central limit theorem, we can prove that

$$\hat{w}_3 - w_3^* \sim N(w; 0, \frac{1}{n} K(w_3^*)^{-1}).$$

Therefore,

$$G_3(n) = \frac{1}{2n} \text{Tr}[I(a^*)K(w_3^*)^{-1}] + O(1/n^2),$$

which completes the proof. (*End of proof*)

Comparing to the coefficients of the generalization errors, we can find the following:

Corollary 1. *Generalization error of M3 is smaller than that of M1 or M2.*

Though we omit the proof for lack of space, it is easy to check that $\text{Tr}[I(a^*)K(w_3^*)^{-1}]$ is not more than d/α and $\text{Tr}[I(a^*)J(w_2^*)^{-1}]/(1-\alpha)$. This corollary implies that the advantage resulting from increasing the number of training data is stronger than the disadvantage caused by the cost for parameter tuning.

Examples and Experiments. Here, we compare the generalization errors of the $G_i(n)$ for $i = 1, \dots, 4$ in a simple example.

Suppose that the true distributions and the learning models are defined by

$$\begin{aligned} q_{01}(x) &= N(x; a^*, 1), \quad q_{02}(x) = N(x; b^*, 1), \\ p_{01}(x|a) &= N(x; a, 1), \quad p_{02}(x|b) = N(x; b, 1), \end{aligned}$$

where $N(x; \mu, \sigma^2)$ stands for the Gaussian distribution with respect to $x \in R^1$.

Figures 1 shows the generalization errors of the models as a function of α , the proportion of data in the task of interest, in a representative case. We studied a comprehensive set of parameter values, $b^* = 1, 2, \dots, 10$ and $c^* = 0.1, 0.2, \dots, 0.9$. The vertical axis is the coefficient of the leading term in each $G_i(n)$. When α is small, that is, the amount of data in D_2 is large enough and that of D_1 is not, the estimation of M1 which is based only on D_1 suffers badly. We can observe that M3 is always better than M1 and M2, as predicted by Corollary 1.

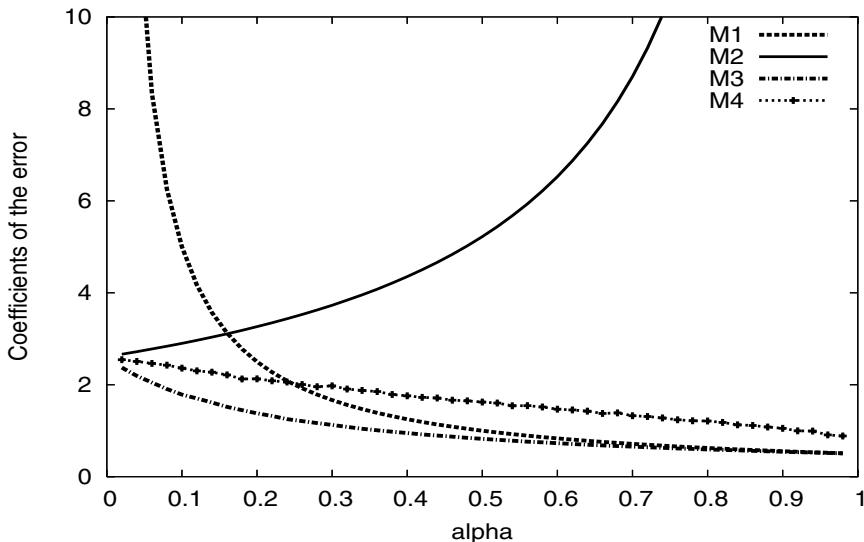


Fig. 1. Generalization errors of the four models: $a^* = 0, b^* = 2, c^* = 0.6$

4 Discussion

In this paper we have derived the asymptotic generalization error of relevant subtask learning models, estimated using maximum likelihood. The results gave an interesting insight that RSM is the optimal when the model can attain the true distribution. This means that the advantage to get more data is larger than the disadvantage to increase the complexity of the model. Since our model setting is general, applications to practical data such as bioinformatics are expected in future studies.

Acknowledgements

This research partly supported by the Ministry of Education, Science, Sports, and Culture in Japan, Grant-in-aid for scientific research 18079007. SK belongs to the Adaptive Informatics Research Centre of the Academy of Finland, Helsinki Institute for Information Technology, and PASCAL2 Network of Excellence of the European Community.

References

1. Kaski, S., Peltonen, J.: Learning from relevant tasks only. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) ECML 2007. LNCS, vol. 4701, pp. 608–615. Springer, Heidelberg (2007)
2. Caruana, R.: Multitask learning. Machine Learning 28(1), 41–75 (1997)
3. Marx, Z., Rosenstein, M.T., Kaelbling, L.P., Dietterich, T.G.: Transfer learning with an ensemble of background tasks. In: NIPS workshop on inductive transfer (2005)

4. Raina, R., Ng, A.Y., Koller, D.: Transfer learning by constructing informative priors. In: NIPS workshop on inductive transfer (2005)
5. Zhu, X.: Semi-supervised learning literature survey. Technical Report TR1530, Computer Science, University of Wisconsin Madison (December 2007)
6. Tax, D.M.J.: One-class classification. PhD thesis, Delft University of Technology (2001)
7. Tax, D.M.J., Duin, R.P.W.: Uniform object generation for optimizing one-class classifiers. *Journal of Machine Learning Research* 2, 155–173 (2001)
8. Fedorov, V.V.: Theory of Optimal Experiments. Academic Press, New York (1972)
9. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90, 227–244 (2000)
10. Wolpert, D.H.: Stacked generalization. Technical Report LA-UR-90-3460, Los Alamos, NM (1990)
11. Jacobs, R.A., Jordan, M.I., Nowlan, S., Hinton, G.E.: Adaptive mixtures of local experts. *Neural Computation* 3, 1–12 (1991)
12. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Technical Report AIM-1440 (1993)

Intelligent Automated Guided Vehicle with Reverse Strategy: A Comparison Study

Shigeru Kato^{1,2} and Kok Wai Wong¹

¹ Murdoch University, South St, Murdoch 6150 Western Australia

² Yuge National College of Maritime Technology, 1000 Shimoyuge Yuge Kamijima-cho
Ochigun 794-2593, Japan

skatou@mech.yuge.ac.jp, k.wong@murdoch.edu.au

Abstract. This paper describes the intelligent automated guided vehicle (AGV) control system. The AGV used in this paper is a virtual vehicle simulated using computer. The purpose of the control system is to control the simulated AGV for moving along the given path towards the goal. Some obstacles can be placed on or near the path to increase the difficulties of the control system. The intelligent AGV should trace the path by avoiding these obstacles. In some situations, it is inevitable to avoid the obstacles without reversing. In this paper, we look into the use of fuzzy automaton for controlling the AGV. In order to better avoid the obstacles, reverse strategy has been implemented to the fuzzy automaton controller. Another alternative to incorporate the human expertise and observations is to use a hybrid intelligent controller using fuzzy and case base reasoning to implement the reverse strategy. This paper presents the comparison results for the three intelligent AGV systems used to avoid obstacles.

1 Introduction

Fuzzy control system has become popular because of its ability to assign meaningful linguistic labels to the fuzzy sets in the rule base. Fuzzy control has been popular as an intelligent method to handle complex problems [1]. The controller sometimes faces with unexpected situation due to the complexity of the problem and may not be able to provide appropriate inference. Sometime when the structure of the data is complex and incomplete, it is also difficult to handle the inference [2].

This paper provides a comparison study for controlling an Automated Guided Vehicle (AGV) [3]. The AGV used in this paper is a computer simulated vehicle. The purpose of the control system is to control the simulated AGV by moving along the given guide path towards the goal called the docking station. Some obstacles can be placed on or near the path to increase the difficulties of the control system. The intelligent AGV should trace the guide path by avoiding these obstacles. In order to control the AGV, fuzzy reasoning and automata have been used [4]. Although the original fuzzy controller is sufficient to control AGV in most foreseeable situations, sometime the AGV will collide with the obstacles and get stuck in some unexpected situations. To handle such unexpected situations, it is essential to introduce the reverse strategy. For comparison purposes, we used three different intelligent AGV controllers. For the first

controller (i.e. Controller A), it is the original implementation of the AGV using fuzzy automaton [4]. However in the original AGV control system, they only use forward controlled strategy to avoid the obstacles. In the second controller (i.e. Controller B), we introduced the reverse strategy into the intelligent controller. In the third controller (i.e. Controller C), we use the hybrid intelligent controller which consists of fuzzy and case based reasoning (CBR) to implement the reverse strategy [5] [6].

2 Automated Guided Vehicle

In this section, we give a brief overview of the computerized AGV simulator. An AGV shown in Fig.1 (left) is an intelligent unmanned machine which is capable of determining its motion status according to the environment condition. It is equipped with an on-board controller which enables it to process information obtained from the sensors that are attached on the vehicle. In this paper AGV has two fixed directional wheels and four free directional wheels as shown in Fig.1 (left).

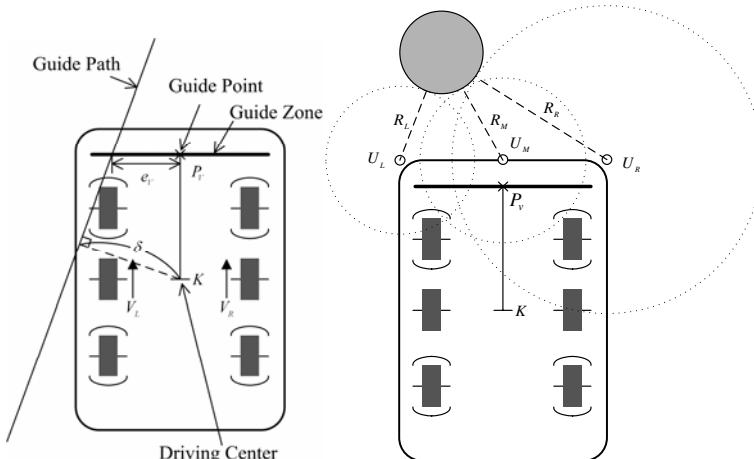


Fig. 1. Structure of AGV (left) and obstacle sensor (right)

The AGV can move and turn by controlling the contour speed V_L and V_R of two fixed directional wheels. In the cases $V_L = V_R$, $V_L > V_R$ and $V_L < V_R$, the AGV goes straight, turns right and turns left, respectively. The guide path is usually a painted marking or a passive or active wire (guide wire) glued onto or built into the floor. The main goal of the AGV is to follow the marking of the guide path. The guiding system senses the position of the guide path by special sensors (guide zone) tuned for the guide path. The guide zone is a section of the AGV as shown in Fig.1 (left). The distance e_v between the guide path and the guide point is calculated. The distance δ between the guide path and the driving centre (called path tracking error) is estimated value from e_v and other information [3].

The main goal of the steering control is path tracking (to follow a guide path). To make the simulated task more complex, we added a second goal as collision avoidance.

The base idea of the path tracking is very simple: keep the driving centre K of the AGV as close as it is possible to the guide path, and then if the driving centre is close enough to the guide path, simply turn the AGV into the docking station's direction (goal's direction). For defining the collision avoidance, we have to study the types of the possible collision situations. There are two different collision situations, the frontal and the side collision. It is sufficient to have three ultrasonic distance sensors on the front of the AGV as shown in Fig.1 (right). The three distances (R_L, R_M, R_R) are measured by the three obstacle sensors. However the observations to avoid the side collision are not so simple. For having the observations, we calculate the actual maximal left and right turning angle without side collision (α_{ML}, α_{MR}) [9].

3 Fuzzy Automaton Control System Using Reverse Strategy

Fig.2 (left) shows the control result by using the original fuzzy automaton AGV controller (Controller A). In this case, the fuzzy controller cannot handle this situation appropriately (accordingly the AGV collides with the obstacle). To avoid such collision, we know that the AGV should simply go backward (reverse) before collision. However the reverse behavior is not taken into account when designing the original fuzzy controller. In order to take into account of such behavior, in this paper we tried to incorporate reverse behavior in the original fuzzy controller. Fig.2 (right) shows the fuzzy automaton controller which the reverse strategy is incorporated (Controller B). The control system outputs V_a and V_d which are input of the AGV. V_a is required speed of the AGV and V_d is required steering degree. V_L and V_R are determined from required V_a and V_d [3].

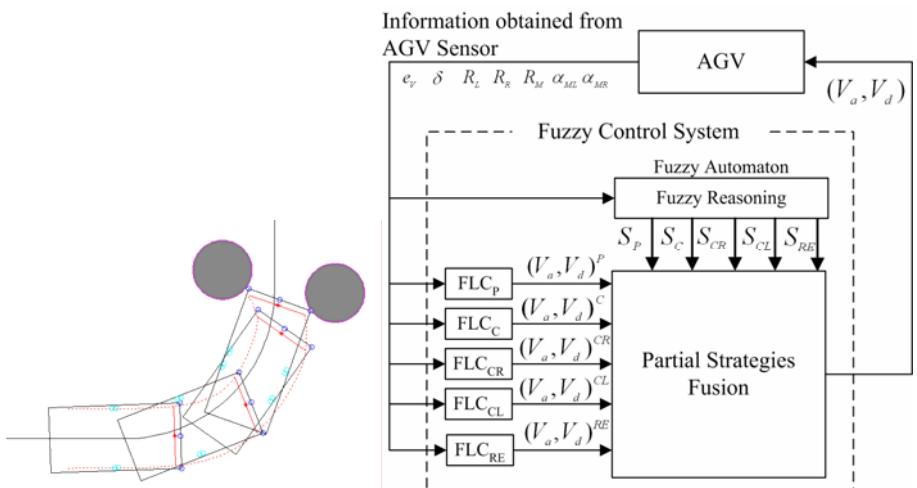


Fig. 2. The result of Controller A and fuzzy controller with reverse strategy (Controller B)

When V_a is positive value, AGV runs forward. When V_a is negative value, AGV runs backward. The bigger V_a is inputted to the AGV, the faster the AGV runs. When V_d is positive value, AGV turns right. When V_d is negative value, AGV turns left. The higher degree of V_d is given to the AGV, the bigger the angle of turn. The input to the fuzzy control system is sensed information by AGV sensor and calculated one after it moved, such as e_v , δ , R_L , R_M , R_R , α_{ML} and α_{MR} .

This system has five Fuzzy Logic Controllers (FLC) and each controller outputs each conclusion of (V_a, V_d) as shown in Fig.2 (right). The FLC has each partially valid strategy to control the AGV in particular situation. These five strategies are “*The path tracking and restricted collision avoidance strategy*”, “*The collision avoidance strategy*”, “*The collision avoidance with left tendency strategy*”, “*The collision avoidance with right tendency strategy*” and “*The reverse strategy*”.

The FLC_P has *the path tracking and restricted collision avoidance strategy*. The main goal of this strategy is following the guide path. And sub goal is avoiding obstacles without risking the chance of losing the guide path. The numbers of fuzzy rules of this strategy for V_d and V_a are 12 and 5, respectively. It may seem the number of rules is smaller than ordinal. However since fuzzy interpolation technique [7,10,11] is used, reasonable conclusions can be obtained from small number of significant rules. FLC_C has *the collision avoidance strategy*. The numbers of rules of this strategy for V_d and V_a are four and two, respectively. FLC_{CL} and FLC_{CR} have *the collision avoidance with left tendency strategy* and *the collision avoidance with right tendency strategy*, respectively. These strategies perform collision avoidance steering with left or right turning tendencies in case of no left or right turning difficulties. These strategies are necessary to aid finding guide path after leaving from the path to avoid obstacle [4]. The original fuzzy controller (Controller A) has only FLC_P, FLC_C, FLC_{CL} and FLC_{CR}. These strategies output positive value of V_a to run forward. FLC_{RE} has *the reverse strategy* which is implemented in this paper for comparison purpose. This strategy outputs negative value of V_a to reverse. The rules for this strategy for V_d and V_a are shown in Table.1. The first rule of V_d describes if the obstacle exists on the right hand, the AGV turns left. The third rule describes if the frontal obstacles do not exist, FLC_{RE} controls the AGV turns right. The rule of V_a means FLC_{RE} always outputs large negative value of V_a .

Table 1. The reverse strategy

	R_L	R_R	R_M	V_d
1	L	Z		NL
2	Z	L		PL
3	L	L		PL

	R_L	R_R	R_M	V_a
1				NL

N: negative, P: positive, L: large, Z: zero

Finally one conclusion of (V_a , V_d) is obtained by considering the output of fuzzy automaton. S_p , S_c , S_{CR} , S_{CL} and S_{RE} express the similarity between AGV state and the prerequisite of each strategy, respectively [4]. Five state-transition rule bases are defined. We have the R_{SP} state transition rule base for determining the next value of the S_p state variable, R_{SC} for S_c , R_{SCR} for S_{CR} , R_{SCL} for S_{CL} and R_{SRE} for S_{RE} . The observations of the state transition rule bases include e_v , R_L , R_M , R_R , α_{ML} , α_{MR} , the state variables themselves (S_p , S_c , S_{CR} , S_{CL} , S_{RE}) and additional observation PV expressing if the path sensing is available (valid) or not. The state variable means the influencing degree of the each FLC's output to the final conclusion (V_a , V_d) outputted to the AGV.

This part also shows the experimental result of the controller and discusses the future work. We conducted the simulation experiment of the proposed controller and the one in which S_{RE} is set to 0 (i.e. the original controller). In some situations where the original controller cannot go beyond the obstacles, the proposed controller can arrive in the docking station (goal) by avoiding the obstacles with reverse strategy.

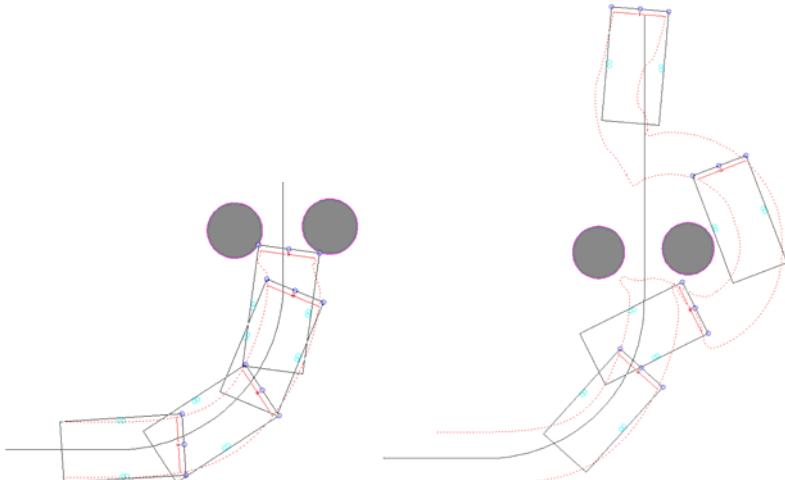


Fig. 3. Result by the Controller A (left) and one by the Controller B (right)

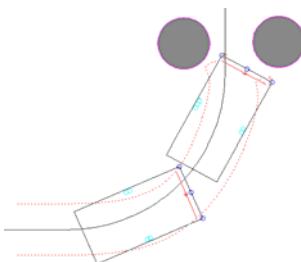


Fig. 4. Another result by the Controller B

Fig.3 shows the control results by the original controller and proposed one in the same situation. It is found that the AGV collides to the obstacles by the original controller, on the other hand the one by proposed controller arrives in the docking station by avoiding the obstacles using reverse strategy. We confirmed that the AGV reverses in the front of the obstacles. It is found that the required speed V_a becomes a big negative value when the AGV is reversing. That is why S_{RE} becomes big at the time. Therefore it is confirmed that by incorporating reverse strategy, it is effective to go beyond the obstacles.

Fig.4 shows another example by the Controller B where the AGV almost stops in front of the obstacles. It is found that the required speed V_a is changing around 0 point within narrow limits. The reverse strategy FLC_{RE} requires V_a to be a negative value, on the other hand the other strategies require V_a to be positive value. These strategies could conflict with each other. Therefore it is confirmed that the incorporation of the reverse strategy in the original fuzzy controller (Controller A) is complex. In the future work, we will investigate parameter optimization e.g. a hill-climbing method to improve the reversing strategy.

4 Hybrid Fuzzy and CBR Controller

With the complexity issue in mind, in Controller C for comparison purposes, another processing unit outside the original fuzzy controller is used to solve the crucial situations where reverse movement is essential. Fig.5 (left) shows the hybrid control system which consists of fuzzy control part (original fuzzy controller [4]) dealing with forward movement and CBR control part to manage reverse strategy [5] [6]. This controller is named as Controller C in this paper. Only one output to the AGV is selected from the outputs of controllers depending on the situation.

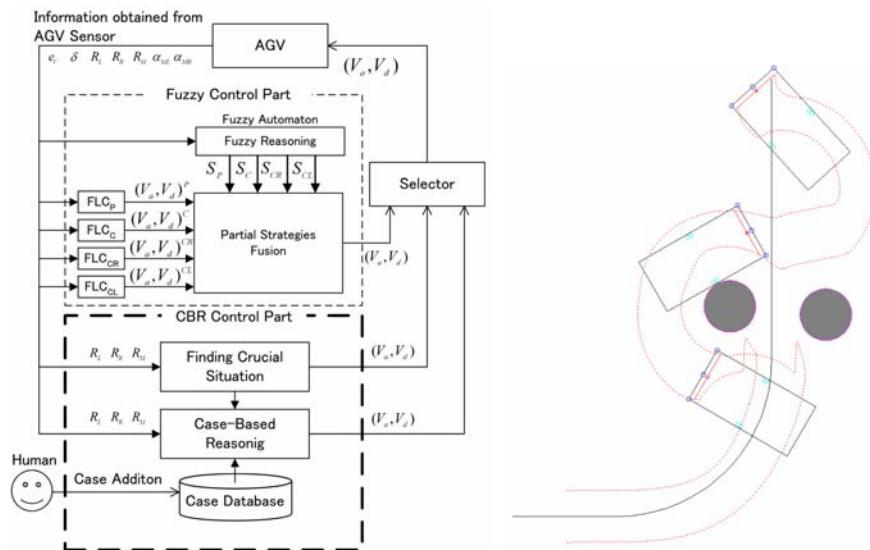
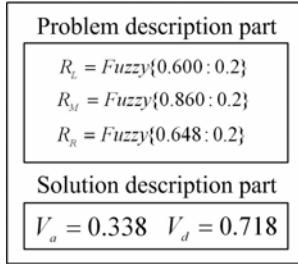


Fig. 5. Controller C (left) and the control result (right)

**Fig. 6.** Example of Case

The Case-Based Reasoning (CBR) [8] which can solve particular problem using accumulated case data directly without generalizing rules. In the hybrid system the case can be constructed in the way that the human only gives intuitively the appropriate control output values under certain condition by observing the control target situation, i.e. the system can derive the human intelligence interactively. And the system can solve the problem by using the case reflecting human intelligence.

The fuzzy control part mainly controls the AGV. However when the fuzzy control part encounter the crucial situation defined as $R_L \leq 0.1$ or $R_R \leq 0.1$ or $R_M \leq 0.1$, the CBR control part controls AGV to reverse straightly until R_L , R_R and R_M become higher than 0.6 which is safe situation without collision risk. After reverse, AGV should head towards to the docking station (goal). Therefore the CBR will find relevant case for control output to go beyond the obstacles.

The case used in CBR is a rule which has certain solution in certain situations. Therefore the case consists of the problem description part expressing the situation to be solved and solution description part expressing the method to solve the problem. The case is to be given by human expert. Fig. 6 shows the example of the case. The problem description part consists of three slots. The value of each slot is expressed by triangular fuzzy set. For example, $\text{Fuzzy}\{0.600 : 0.2\}$ means the triangular fuzzy set in which center value is 0.600 and width is 0.2. V_a and V_d in the solution description part are appropriate output for the situation in problem description part.

Some cases are preserved in the case database. The CBR control part retrieves similar case by comparing the situation (i.e. information obtained from AGV sensor) with the problem description part. And the similarity of i -th slot in the problem description part of j -th case is calculated by equation (1).

$$\text{sim}_i^j = \sup_{x \in X} (\mu_i^j(x) \wedge \mu_i^*(x)) \quad (1)$$

where X is the domain of the fuzzy set which is $[0,1]$, μ_i^j is the membership function of i -th slot of j -th case, μ_i^* is the membership function in which actual sensed value by AGV sensor is center value and width is 0.2. The case similarity of j -th case with the actual situation is calculated by equation (2).

$$\text{similarity}_j = \min_{i=1,2,3} (\text{sim}_i^j) \quad (2)$$

The cases with case similarity higher than the threshold (0.7) in the case database are selected as solution case candidate. The lower threshold such as 0.5 may cause the

control to fail. On the other hand, higher threshold such as 0.9 may lead to no applicability of the case. Therefore we define the threshold to be 0.7 as appropriate for our case. The highest similarity case among all solution case candidates is selected as the solution case. And then V_a and V_d in the solution description part of the solution case are outputted to the AGV for 1.5 [second] to control the AGV. While this value is being outputted, the output from fuzzy control part is disabled by the selector. When the solution case is not obtained, CBR control part asks the human to add the case interactively. In this case, CBR control part shows an interface to the human in order to request him/her to add the new case. The human sets the appropriate V_a and V_d using interface as he/she sees the AGV situation. And the case consisting of problem description part expressing R_L , R_R and R_M and solution description part expressing V_a and V_d is preserved in the case database. After that, AGV is controlled with set V_a and V_d for 1.5 [second]. Fig.5 (right) shows the control result in the situation where the original fuzzy controller collides to the obstacle and consequently cannot arrive in the goal. The dotted line means the trajectory of AGV. It is found that the AGV reverses just before colliding and the AGV turns left after reversing. To show the stability of the Controller C, the simulation experiment is performed in the condition where there are one guide path having one curve of 3.0 [m] radius and two obstacles deployed around the path randomly. The maximum speed, width and length of AGV are 1.5 [m/s], 1.1 [m] and 2.2 [m], respectively. Twelve cases are given in the case database. The AGV with Controller B tried 10 situations where AGV with Controller A collides to the obstacles. By Controller B the AGV went beyond the obstacles in 5 situations. On the other hand the AGV with Controller C tried other 10 situations where the Controller A failed, and the AGV cleared all situations.

5 Conclusions

This paper presented the comparison results of three intelligent automated guided vehicle (AGV) control system. For the first controller (i.e. Controller A), it is the original implementation of the AGV using fuzzy automaton [4]. However in the original AGV control system, they only use forward controlled strategy to avoid the obstacles. In the second controller (i.e. Controller B), we introduced the reverse strategy into the intelligent controller. In the third controller (i.e. Controller C), we use the hybrid intelligent controller which consists of fuzzy and case based reasoning (CBR) to implement the reverse strategy [5] [6]. Results show that Controller B and C could provide reasonable reverse strategy for AGV. In this comparison, no parameter optimization has been used. Future work will be focusing in this area.

References

1. Mamdami, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Human-Computer Studies 7, 1–13 (1975)
2. Wong, K.W., Gedeon, T.D., Koczy, L.: Fuzzy Signature and Cognitive Modelling for Complex Decision Model. In: Castillo, O., Melin, P., Montiel Ross, O., Sepúlveda Cruz, R., Pedrycz, W., Kacprzyk, J. (eds.) Theoretical Advances and Applications of Fuzzy Logic and Soft Computing. Advances in Soft Computing Series, vol. 42, pp. 380–389. Springer, Heidelberg (2007)

3. Cselényi, J., Kovács, S., Pap, L., Kóczy, L.T.: New Concepts in the Fuzzy Logic Controlled Path Tracking Strategy of the Differential Steered AGVs. In: 5th International Workshop on Robotics in Alpe-Adria-Danube Region, RAAD 1996, pp. 587–592 (1996)
4. Kovács, S.: Similarity based Control Strategy Reconfiguration by Fuzzy Reasoning and Fuzzy Automata. In: Proceedings of the IECON 2000, IEEE International Conference on Industrial Electronics, Control and Instrumentation, pp. 542–547 (2000)
5. Kato, S., Wong, K.W.: The Automated Guided Vehicle Using Fuzzy Control and CBR Techniques. In: Proceedings of SCIS & ISIS 2008, pp. 1788–1792 (2008)
6. Kato, S., Wong, K.W.: Fuzzy and Case Based Reasoning Obstacle Avoidance Techniques for Games and Robots. *IEEE Transactions on Computational Intelligence and AI in Games* (submitted)
7. Wong, K.W., Tikk, D., Gedeon, T.D., Kóczy, L.T.: Fuzzy rule interpolation for multidimensional input spaces with applications: A case study. *IEEE Trans of Fuzzy Systems* 13(6), 809–819 (2005)
8. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *Artificial Intelligence Communications* 7(1), 39–59 (1994)
9. Kovács, S., Kóczy, L.T.: Application of an approximate fuzzy logic controller in an AGV steering system, path tracking and collision avoidance strategy. In: *Fuzzy Set Theory and Applications*, Tatra Mountains Mathematical Publications. Mathematical Institute Slovak Academy of Sciences, Bratislava, Slovakia, vol. 16, pp. 456–467 (1999)
10. Kovács, S.: Fuzzy Rule Interpolation in Practice. In: Proceedings of the Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on advanced Intelligent Systems (SCIS & ISIS 2006), Tokyo Institute of Technology, Tokyo, Japan, p. 6 (2006) (invited talk)
11. Johanyak, Z.C., Tikk, D., Kovács, S., Wong, K.W.: Fuzzy Rule Interpolation Matlab Toolbox - FRI Toolbox. In: Proceedings of IEEE International Conference on Fuzzy Systems 2006, Vancouver, Canada, pp. 1427–1433 (2006)

Neural Networks for Optimal Form Design of Personal Digital Assistants

Chen-Cheng Wang¹, Yang-Cheng Lin², and Chung-Hsing Yeh³

¹ Department of Computer Simulation and Design, Shih Chien University,
Kaohsiung, 845, Taiwan
ccwang590@seed.net.tw

² Department of Arts and Design, National Dong Hwa University, Hualien, 970, Taiwan
lyc0914@cm1.hinet.net

³ Clayton School of Information Technology, Faculty of Information Technology,
Monash University, Clayton, Victoria 3800, Australia
ChungHsing.Yeh@infotech.monash.edu.au

Abstract. This paper presents a neural network (NN) approach to determining the optimal form design of personal digital assistants (PDAs) that best matches a given set of product images perceived by consumers. 32 representative PDAs and 9 design form elements of PDAs are identified as samples in an experimental study to illustrate how the approach works. Four NN models are built with different hidden neurons in order to examine how a particular combination of PDA form elements matches the desirable product images. The performance evaluation result shows that the number of hidden neurons has no significant effect on the predictive ability of the four NN models. The NN models can be used to construct a form design database for supporting form design decisions in a new PDA product development process.

1 Introduction

Products have been considered to be a symbol of occupation, personality, opinion, and other human attributes. Whether a product is successful largely depends on the final judgment of consumers [1]. Therefore, designers need to comprehend the consumers' needs in order to create successful products in an intensely competitive market. Moreover, a successful product should not only possess good functionalities, interface design, and operating performance, but also need to take the product image design into account to satisfy consumers' psychological requirements [6, 15]. The external appearance of a product can represent a product image that evokes consumers' internal resonance and consuming motivation. The product image engages a critical influential factor in consumers' preference structure [2]. When choosing a product, consumers tend to rely on their own particular perception of the product, which is regarded as something of a black box [3]. As an ergonomic consumer-oriented methodology, Kansei Engineering is developed as integrative design strategies for affective design to satisfy consumers' psychological requirements [7, 8, 12]. The word "Kansei" indicates

the consumers' psychological requirements or emotional feelings of a product. Kansei Engineering has been used to assist product designers in designing product forms that can best match specific product images [5, 9]. To help product designers work out the optimal combination of product forms for a particular design concept represented by product images, neural networks (NNs) are used in this paper because of their effective learning ability.

In subsequent sections, we first present an experimental study with personal digital assistants (PDAs). Then we construct and evaluate four NN models based on the experimental data. Finally we discuss how the NN models can be used as a design support tool for simulating the optimal form design process.

2 Experimental Procedures

We conduct an experimental study using the concept of Kansei Engineering in order to collect numerical data about the relationship between product images and design form elements of PDAs. The experimental study involves three main steps: (a) extracting representative experimental samples, (b) conducting morphological analysis of design form elements, and (c) assessing product images.

2.1 Extracting Representative Experimental Samples

In the experimental study, we investigated and categorized various PDAs on the market. We first collected 88 PDAs and then classified them based on their similarity degree. To collect opinions regarding the usage, function and form of PDAs, a focus group was formed by six subjects with at least two years' experience of using the PDA. The focus group eliminated some highly similar samples through discussions. Then the K-means cluster analysis was used to extract representative samples of PDAs. There are 32 representative PDA samples, including 27 samples as the training set and 5 samples as the test set for building four neural network models.

2.2 Conducting Morphological Analysis of Design Form Elements

The product form is defined as the collection of design features that the consumers will appreciate. The morphological analysis [17], concerning the arrangement of objects and how they conform to create a whole of Gestalt, is used to explore all possible solutions in a complex problem regarding a product form.

The morphological analysis was used to extract the form elements of the 32 representative samples. The six subjects of the focus group were asked to decompose the PDA samples into several dominant form elements and form types according to their knowledge and experience. The form type indicates the relationship between the outline elements. A number of design alternatives can be generated by various combinations of morphological elements [4, 10, 11]. Table 1 shows the result of the morphological analysis, with nine design elements and 32 associated form types being identified.

Table 1. Morphological analysis of PDA design forms

Form element	Form type			
	Type 1	Type 2	Type 3	Type 4
Open type (X ₁)				
Non-cap				
Dragging				
Folding				
Rotating				
Top shape (X ₂)				
Line				
Chamfer				
Fillet				
Shield				
Bottom shape (X ₃)				
Fillet				
Chamfer				
Shield				
Arc				
Function-keys arrangement (X ₄)				
Line				
Symmetry				
Irregular				
Grouping				
Arrow-key style (X ₅)				
Cycle				
Ellipse				
Diamond				
Straight				
Audio-hole style (X ₆)				
Dots				
Strips				
Radiation				
Audio-hole position (X ₇)				
Bottom				
Top				
In arrow-key				
Color treatment (X ₈)				
Single color				
Non-color Segment				
Color segment				
Outline partition style (X ₉)				
Normal partition				
Fitting outline				
Fitting surface				

2.3 Assessing Product Images

In Kansei Engineering, image assessment experiments are usually performed to elicit the consumers' psychological feeling or perception about a product using the semantic differential (SD) method. Pairs of image words are often used to describe the consumers' perception of the product in terms of ergonomic and psychological estimation. With the identification of the design elements of the product, the relationship between the image words and the design elements can be established.

Table 2. Product image assessment of 32 sample PDAs

PDA No.	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	C-D value	N-V value	S-C value
1	1	3	1	1	4	2	1	1	1	4.67	5.67	1.67
2	1	3	1	1	4	2	2	1	1	5.67	1.67	2.33
3	1	2	2	3	3	1	2	3	2	3.33	4.67	3.33
4	1	1	2	2	4	2	1	1	1	2.67	1.67	3.67
5	1	3	1	1	2	1	1	1	1	3.33	2.67	1.67
6	2	2	2	1	1	2	1	1	1	7.33	1.67	8.33
7	1	3	1	1	4	1	1	1	3	2.33	6.67	2.33
8	1	3	1	2	2	2	1	1	1	4.33	7.67	2.33
9	1	4	4	2	2	2	2	2	1	1.67	1.67	6.33
10	1	3	1	3	1	1	2	2	1	5.33	4.67	3.33
11	1	3	3	2	2	1	2	2	1	4.67	2.33	4.67
12	1	1	1	1	1	1	1	1	1	3.67	2.67	1.67
13	1	3	1	1	1	1	1	1	1	5.33	2.67	3.33
14	1	2	1	1	2	2	1	3	2	3.67	6.33	2.33
15	1	3	1	2	1	2	1	3	3	3.67	6.33	2.33
16	1	3	1	3	1	2	1	3	3	8.33	2.67	4.67
17	1	2	1	2	1	1	1	2	1	5.33	4.67	7.33
18	3	3	1	1	2	1	1	1	1	5.33	1.33	7.67
19	1	1	1	2	2	3	3	2	2	7.67	3.67	8.33
20	3	3	1	1	2	1	2	1	1	2.33	1.67	4.67
21	3	3	1	4	4	2	1	1	1	2.67	6.67	1.67
22	3	3	1	4	1	1	2	1	1	2.33	1.67	2.33
23	1	3	1	1	2	1	2	1	1	4.33	6.67	5.67
24	1	2	4	1	1	1	1	1	1	2.33	2.33	1.67
25	1	2	2	2	1	1	1	2	1	5.33	3.67	1.33
26	1	3	1	4	2	1	2	3	1	3.67	2.67	4.67
27	1	3	1	2	2	3	3	3	2	5.33	2.67	5.33
28*	1	2	2	1	2	1	2	1	1	2.67	2.33	2.33
29*	1	3	1	1	1	1	2	1	1	2.33	1.67	4.33
30*	4	3	1	2	1	1	2	1	1	3.67	1.67	5.67
31*	1	2	1	1	1	1	1	1	1	3.33	2.67	2.33
32*	1	3	1	1	4	2	1	2	2	3.67	5.33	4.33

In this paper, the image word pairs (product images) used are (a) Convenient–Difficult (C–D) about the operational aspect, (b) Noble–Vulgar (N–V) about the value aspect, and (c) Simple–Complex (S–C) about the visibility aspect, according to our previous study [16]. To obtain the assessment values for these three product images of 32 representative PDA samples, a 10-point scale (1–10) of the SD method is used. For example, 52 subjects (30 males and 22 females with ages ranging from 26 to 45, mean=35.4, SD=4.4) are asked to assess the form (look) of PDA samples on a simplicity-complexity scale of 1 to 10, where 10 is most simple and 1 is most complex. The last three columns of Table 2 show the assessment results of the 32 PDA samples, including 27 samples in the training set and 5 samples in the test set (asterisked). For each selected PDA in Table 2, the first column shows the PDA number and Columns 2–10 show the corresponding type number for each of its 9 form elements, as given in Table 1. Table 2 provides a numerical data source for building neural network models, which can be used to develop a design support system for simulating the optimal form design process for PDAs.

3 Neural Network Models

In this paper, we use the multilayered feedforward NN models trained with the back-propagation learning algorithm [13].

3.1 Building NN Models

To examine how a particular combination of PDA form element matches the C-D image, N-V image, and S-C image, we build four NN models, called NN1, NN2, NN3, and NN4 respectively. These four NN models use four most widely used rules [14] for determining the number of neurons in the single hidden layer respectively, given below:

- (a) (The number of input neurons + the number of output neurons) / 2
- (b) (The number of input neurons * the number of output neurons) $\wedge 0.5$
- (c) (The number of input neurons + the number of output neurons)
- (d) (The number of input neurons + the number of output neurons) * 2

Table 3 shows the neurons of the four NN models, including the input layer, the hidden layer, and the output layer. The learning rule used is Delta-Rule and the transfer function is Sigmoid [14] for all layers. All of input and output variables (neurons) are normalized before training. A commercially available NN software system is used with the default settings for the learning rate, momentum, bias terms, and weights.

Table 3. Neurons of four NN models

NN model	Input layer: 32 neurons, including 32 types of PDAs. Output layer: 3 neurons for the C-D image, N-V image, and S-C image values.
NN1	Hidden layer: 18 neurons, $(32+3)/2=17.5 \approx 18$.
NN2	Hidden layer: 10 neurons, $(32*3)^{0.5} \approx 9.80 \approx 10$.
NN3	Hidden layer: 35 neurons, $(32+3)=35$.
NN4	Hidden layer: 70 neurons, $(32+3)*2=70$.

The experimental samples were separated into two groups: 27 training samples and 5 test samples. Each model was trained 5,000 epochs at each run. When the cumulative training epochs were over 100,000, the training process was completed. Table 4 shows the training epochs of each model run and their corresponding root of mean square (RMS) errors. The lowest RMS error of each model is asterisked. As shown in Table 4, the RMS error of four NN models using the NN3 rule in (c) is the lowest (0.0323), as compared to the other three rules. However, the lowest RMS error of four NN models is almost the same. This result indicates that the number of hidden neurons has little influence on the training effect of NN models.

Table 4. RMS errors of the four NN models for the training set

Training set Epochs	NN model			
	NN1	NN2	NN3	NN4
5000	0.0968	0.0977	0.1008	0.0989
25000	0.0543	0.0651	0.0501	0.0601
50000	0.0465	0.0439	0.0373	0.0505
75000	0.0435	0.0369	0.0326	0.0476
100000	0.0432*	0.0364*	0.0323*	0.0474*

3.2 Performance Evaluation of Four NN Models

To evaluate the performance of the four NN models in terms of their predictive ability, the 5 samples in the test set were used. Rows 2-4 of Table 5 show the average assessed values of the C-D, N-V and S-C images on the 5 test samples given by the 52 subjects, and Rows 5-16 show the predicted values for the three images by using these four NN models trained in the previous section. The last column of Table 5 shows the RMS errors of four NN models for the test set.

Table 5. Predicted image values and RMS errors of the four NN models for the test set

	PDA No.	28	29	30	31	32	RMS errors
Consumer perception	C-D	2.67	2.33	3.67	3.33	3.67	
	N-V	2.33	1.67	1.67	2.67	5.33	
	S-C	2.33	4.33	5.67	2.33	4.33	
NN1 model	C-D	4.61	3.55	1.85	5.53	2.01	
	N-V	3.74	3.99	4.40	2.84	8.97	0.2082
	S-C	3.12	1.94	5.79	4.89	-0.51	
NN2 model	C-D	5.85	4.16	2.08	6.04	1.66	
	N-V	3.87	4.25	5.16	2.65	9.05	0.2392
	S-C	4.41	2.70	7.02	5.63	-0.59	
NN3 model	C-D	4.32	3.56	1.40	5.62	2.91	
	N-V	5.28	3.87	2.85	3.03	9.10	0.2035
	S-C	2.60	1.54	5.62	4.59	-0.04	
NN4 model	C-D	4.62	3.55	1.41	5.84	1.93	
	N-V	3.71	3.80	3.90	2.68	8.98	0.2099
	S-C	2.77	1.47	6.61	4.73	-0.46	

As indicated in Table 5, the RMS error (0.2035) of the NN3 model is the smallest among the four models, thus suggesting that the NN3 model has the highest predictive consistency (an accuracy rate of 79.65%, 100%-20.35%) for predicting the values of the C-D, N-V, and S-C images of PDAs. This suggests that the NN3 model is most promising for modeling consumers' perception on product images of PDAs. However, the other three models also have a quite similar performance, as the difference between

the RMS errors of the four models is almost negligible. This seems to suggest that the number of neurons in the hidden layer has no significant impact on the predictive ability of the NN models.

4 Neural Networks for Design Support

The NN models enables us to build a PDA design support database that can be used to help determine the optimal form design for best matching specific product images. The design support database can be generated by inputting each of all possible combinations of form design elements to the NN models individually for generating the associated image values. The resultant PDA design support database consists of 82,944 ($=4\times4\times4\times4\times3\times3\times3\times3$) different combinations of form elements, together with their associated C-D, N-V, and S-C image values. The product designer can specify desirable image values for a new PDA form design, and the database can then work out the optimal combination of form elements. In addition, the design support database can be incorporated into a computer aided design (CAD) system or virtual reality (VR) technology to facilitate the form design in the new PDA product development process. To illustrate, Fig. 1 shows the optimal combination of form elements for the three desirable images ("Most Convenient", "Most Novel" and "Simplest") of PDAs respectively.



Fig. 1. Optimal combination of form elements for three desirable images of PDAs

5 Conclusion

In this paper, we have demonstrated how NN models can be built to help determine the optimal product form design for matching a given set of product images, using an experimental study on PDAs. The result of the experimental study has shown that product forms greatly affect the consumers' perception about the convenient-difficult image, the noble-vulgar image, and the simple-complex image of PDAs. It has also

demonstrated the advantage of using NN models for supporting the product form design process. Although PDAs are used as the experimental product, the NN approach presented can be applied to other consumer products with a wide variety of design form elements.

References

1. Brown, S.L., Eisenhardt, K.M.: Product development: past research, present findings, and future directions. *Academy of Management Review* 20, 343–378 (1995)
2. Chuang, M.-C., Chang, C.-C., Hsu, S.-H.: Perceptual factors underlying user preferences toward product form of mobile phones. *International Journal of Industrial Ergonomics* 27, 247–258 (2001)
3. Chuang, M.-C., Ma, Y.-C.: Expressing the expected product images in product design of micro-electronic products. *International Journal of Industrial Ergonomics* 27, 233–245 (2001)
4. Cross, N.: *Engineering Design Methods: Strategies for Product Design*. John Wiley and Sons, Chichester (2000)
5. Hsiao, S.-W., Liu, M.-C.: A morphing method for shape generation and image prediction in product design. *Design Studies* 23, 533–556 (2002)
6. Jonathan, C., Craig, M.V.: *Creating Breakthrough Products- Innovation from Product Planning to Program Approval*, pp. 1–31. Prentice Hall, New Jersey (2002)
7. Lai, H.-H., Lin, Y.-C., Yeh, C.-H., Wei, C.-H.: User Oriented Design for the Optimal Combination on Product Design. *International Journal of Production Economics* 100, 253–267 (2006)
8. Lai, H.-H., Lin, Y.-C., Yeh, C.-H.: Form Design of Product Image Using Grey Relational Analysis and Neural Network Models. *Computers and Operations Research* 32, 2689–2711 (2005)
9. Lin, Y.-C., Lai, H.-H., Yeh, C.-H.: Consumer-oriented product form design based on fuzzy logic: A case study of mobile phones. *International Journal of Industrial Ergonomics* 37, 531–543 (2007)
10. Lin, Y.-C., Lai, H.-H., Yeh, C.-H.: Consumer Oriented Design of Product Forms. In: Yin, F.-L., Wang, J., Guo, C. (eds.) ISNN 2004. LNCS, vol. 3174, pp. 898–903. Springer, Heidelberg (2004)
11. Lin, Y.-C., Lai, H.-H., Yeh, C.-H.: Neural Network Models for Product Image Design. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) KES 2004. LNCS, vol. 3215, pp. 618–624. Springer, Heidelberg (2004)
12. Nagamachi, M.: Kansei engineering: A new ergonomics consumer-oriented technology for product development. *International Journal of Industrial Ergonomics* 15, 3–10 (1995)
13. Negnevitsky, M.: *Artificial Intelligence*. Addison-Wesley, New York (2002)
14. Nelson, M.: *Illingworth WT. A Practical Guide to Neural Nets*. Addison-Wesley, New York (1991)
15. Petiot, J.F., Yannou, B.: Measuring consumer perceptions for a better comprehension, specification and assessment of product semantics. *International Journal of Industrial Ergonomics* 33, 507–525 (2004)
16. Wang, C.-C.: Development of an Integrated Strategy for Customer Requirement Oriented Product Design. PhD Dissertation, Department of Industrial Design, National Cheng Kung University, Tainan, Taiwan (2008)
17. Zwicky, F.: The Morphological Approach to Discovery, Invention, Research and Construction, New Method of Thought and Procedure: Symposium on Methodologies, Pasadena, pp. 316–317 (1967)

Firing Rate Estimation Using an Approximate Bayesian Method

Kazuho Watanabe¹ and Masato Okada^{1,2}

¹ Department of Complexity Science and Engineering, The University of Tokyo,
Mail Box:409, 5-1-5, Kashiwanoha, Kashiwa, 277-8561, Japan
kazuho@mns.k.u-tokyo.ac.jp

² Brain Science Institute, RIKEN, 2-1, Hirosawa, Wako, Saitama, 351-0198, Japan
okada@k.u-tokyo.ac.jp

Abstract. Bayesian estimation methods are used for estimation of an event rate (firing rate) from a series of event (spike) times. Generally, however, the computation of the Bayesian posterior distribution involves an analytically intractable integration. An event rate is defined in a very high dimensional space, which makes it computationally demanding to obtain the Bayesian posterior distribution of the rate.

We consider the estimation of the firing rate underlying behind a sequence that represents the counts of spikes. We derive an approximate Bayesian inference algorithm for it, which enables the analytical calculation of the posterior distribution. We also provide a method to estimate the prior hyperparameter which determines the smoothness of the estimated firing rate.

1 Introduction

Several methods were introduced for the Bayesian estimation of an intensity (event rate) function of a non-homogeneous Poisson process [1][2][7]. Given a sequence of event times, these methods estimate the rate of event occurrence as a function of time. In neuroscience applications, spikes, that are rapid voltage changes in neuronal cells, are considered as the events. The event rate, also called the mean firing rate, is estimated from a sequence of spikes since it is considered to be conveying information between neurons [3][8].

An assumption of the smoothness of the underlying rate is reflected in the prior distribution. Then the posterior distribution of the event rate is obtained via the Bayes rule. However, it involves an analytically intractable integration and requires Monte Carlo sampling or numerical integration methods. The rate function is defined in an infinite dimensional space or in a finite but quite high dimensional space even when its domain is discretized into a set of intervals. Hence, it is computationally demanding to obtain the Bayesian posterior distribution.

The main purpose of this paper is to provide an approximation scheme for the Bayesian estimation of the event rate from a sequence of events. Given a discretized data, that is, a sequence of event (spike) counts, we propose to estimate

the event rate by using approximate posterior distribution which is efficiently computable. More specifically, we introduce additional parameters called variational parameters and approximate the log-likelihood function by a quadratic function. This approximation technique was first developed for the Bayesian logistic regression[6] and is called the local variational method[2]. The resulting approximate distribution makes a Gaussian distribution which can be obtained without any sampling or numerical integration technique. The variational parameters are optimized so as to improve the approximation to the true posterior distribution. We also provide a method to estimate the prior hyperparameter which determines the smoothness of the event rate. The hyperparameter estimation is combined with the above procedure to obtain the approximate posterior distribution.

The rest of this paper is organized as follows. Section 2 describes the model for event rate estimation. The approximate inference scheme is introduced in Section 3. Section 4 presents an example where we apply the proposed method to synthetic data. We examine the computational time and the approximation accuracy of the proposed method by comparing it to the true Bayesian approach implemented by a numerical integration. Discussion and concluding remarks follow in Section 5.

2 Event Rate Estimation

Suppose we observe a time series x_1, x_2, \dots, x_N where $x_i \in \{0, 1, \dots, n\}$ indicates the number of events (spikes) at the i th discrete time. This series can also be obtained from a sequence $\mathbf{t} = (t_1, t_2, \dots, t_{N_s})$, $t_1 < t_2 < \dots < t_{N_s}$ of N_s event times by finely discretizing the observation interval $[0 T]$ into N bins, each with width T/N and at most n event times inside. We assume that each variable x_i independently obeys the binomial distribution with parameter $\lambda_i \in [0 1]$. The likelihood of the event rate $\lambda = \{\lambda_i\}_{i=1}^N$ for the sequence $\mathbf{x} = \{x_i\}_{i=1}^N$ is $p(\mathbf{x}|\lambda) = \prod_{i=1}^N p(x_i|\lambda_i)$ where

$$p(x|\lambda) = \binom{n}{x} \lambda^x (1-\lambda)^{n-x}. \quad (1)$$

Transforming the event rate $\lambda \in [0 1]$ to $y \in R$ by

$$y(\lambda) = \log \frac{\lambda}{1-\lambda} \quad (2)$$

yields the following model from eq.(1),

$$p(x|y) = \binom{n}{x} \exp\{xy - n \log(1 + e^y)\}. \quad (3)$$

We also call the variable $\mathbf{y} = \{y_i = y(\lambda_i)\}_{i=1}^N$ event rate.

We assume the Gaussian distribution as a prior distribution for the Bayesian estimation.

$$p(\mathbf{y}) = \sqrt{\frac{|\beta \mathbf{S}_0|}{(2\pi)^N}} \exp\left\{-\frac{\beta}{2}(\mathbf{y} - \mu_0)^T \mathbf{S}_0 (\mathbf{y} - \mu_0)\right\}, \quad (4)$$

where \mathbf{S}_0 is an $N \times N$ matrix, $\mu_0 \in R^N$ and $\beta > 0$. We fix \mathbf{S}_0 , μ_0 and estimate β as a hyperparameter. The matrix \mathbf{S}_0 specifies the smoothness between the rates y_i and y_j as we will see in eq. (19) as an example.

3 Approximate Bayesian Estimation

We consider the Bayesian estimation of the discretized firing rate $\mathbf{y} = \{y_i = y(\lambda_i)\}_{i=1}^N$ from the sequence $\mathbf{x} = \{x_i\}_{i=1}^N$ of counts using the likelihood (3) and the prior distribution (4). The Bayesian posterior distribution of the firing rate \mathbf{y} is defined by

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{\int p(\mathbf{x}|\mathbf{y})p(\mathbf{y})d\mathbf{y}}. \quad (5)$$

The rate \mathbf{y} is estimated by, for example, the mean of the posterior distribution.

Computation of the posterior distribution involves the analytically intractable integration over N variables and hence requires vast computational costs. In this paper, we approximate the Bayesian posterior distribution (5) by using the technique introduced by Jaakkola and Jordan [6] for the Bayesian logistic regression model, which enables the analytical calculation of the posterior distribution. The derivation until Section 3.3 follows [6] except that we consider general case of n while [6] focuses on the case of $n = 1$, that is, the Bernoulli distribution. The hyperparameter estimation discussed in Section 3.4 is a new development tailored to the event rate estimation method.

3.1 Approximate Posterior Distribution

We approximate the exponent in eq. (3) by a quadratic function of y in order to obtain a Gaussian distribution as an approximating distribution. Let $f(y) = -\log(e^{y/2} + e^{-y/2})$. Since $f(y)$ is a convex function with respect to y^2 , the tangent line of $f(y)$ at $y^2 = \xi^2$ derives the following inequality,

$$f(y) \geq f(\xi) - l(\xi)(y^2 - \xi^2),$$

where we have defined $l(\xi) = \frac{1}{4\xi} \tanh(\frac{\xi}{2})$. Since $-\log(1 + e^y) = f(y) - y/2$, we have the following inequality regarding the model (3),

$$p(x|y) \geq \binom{n}{x} \exp\left\{\frac{(2x-n)y}{2} + nf(\xi) - nl(\xi)(y^2 - \xi^2)\right\} \quad (6)$$

$$\equiv \underline{p}_\xi(x|y). \quad (7)$$

We introduce the variational parameter ξ_i for $i = 1, 2, \dots, N$ and denote the approximate likelihood by $\underline{p}_\xi(\mathbf{x}|\mathbf{y}) = \prod_{i=1}^N \underline{p}_{\xi_i}(x_i|y_i)$. Then it follows that

$$p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) \geq \underline{p}_\xi(\mathbf{x}|\mathbf{y})p(\mathbf{y}) \quad (8)$$

The true posterior distribution (5) is obtained by normalizing the left hand side of the above inequality. We approximate it by the distribution of \mathbf{y} obtained by normalizing the right hand side, that is,

$$\underline{p}_\xi(\mathbf{y}|\mathbf{x}) = \frac{\underline{p}_\xi(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{\int \underline{p}_\xi(\mathbf{x}|\mathbf{y})p(\mathbf{y})d\mathbf{y}} \quad (9)$$

Since the exponent in eq.(6) is a quadratic function of y , from the conjugacy of the prior distribution (4), the approximating posterior distribution $\underline{p}_\xi(\mathbf{y}|\mathbf{x})$ is also a normal distribution. Its variance-covariance matrix \mathbf{S}^{-1} and mean \mathbf{m} are specifically given by the following expressions.

$$\mathbf{S}^{-1} = (\beta \mathbf{S}_0 + 2n\Lambda)^{-1}, \quad (10)$$

$$\mathbf{m} = \mathbf{S}^{-1}\{\beta \mathbf{S}_0 \mu_0 + (\mathbf{x} - n\mathbf{1}/2)\}, \quad (11)$$

where $\mathbf{1}$ is an N -dimensional vector whose elements are all 1 and Λ is the $N \times N$ diagonal matrix with diagonal elements $l(\xi_i)$.

3.2 Optimization of Variational Parameter

The approximating posterior distribution $\underline{p}_\xi(\mathbf{y}|\mathbf{x})$ we have introduced in the previous section depends on the variational parameters $\xi = \{\xi_i\}_{i=1}^N$. They can be optimized so as to improve the approximation to the true posterior distribution. More specifically, we can improve the approximation by optimizing the variational parameters ξ so as to maximize the right hand side of the inequality,

$$\int p(\mathbf{x}|\mathbf{y})p(\mathbf{y})d\mathbf{y} \geq \int \underline{p}_\xi(\mathbf{x}|\mathbf{y})p(\mathbf{y})d\mathbf{y}, \quad (12)$$

which follows from eq.(8). The left hand side $Z \equiv \int p(\mathbf{x}|\mathbf{y})p(\mathbf{y})d\mathbf{y}$ is called the marginal likelihood and is the denominator of the posterior distribution (5). The above inequality provides a lower bound of Z which we denote by $\underline{Z}(\xi) \equiv \int \underline{p}_\xi(\mathbf{x}|\mathbf{y})p(\mathbf{y})d\mathbf{y}$.

We can interpret the lower bound $\underline{Z}(\xi)$ as the likelihood function of ξ by viewing the firing rate \mathbf{y} as an unobservable variable. Therefore we can make use of the Expectation-Maximization (EM) algorithm [5] to update ξ so as to increase the lower bound $\underline{Z}(\xi)$. Given the current estimates of the variational parameters $\tilde{\xi}$, the EM algorithm maximizes the function,

$$Q(\xi|\tilde{\xi}) = E_{\underline{p}_\xi}[\log \underline{p}_\xi(\mathbf{x}|\mathbf{y})p(\mathbf{y})] \quad (13)$$

with respect to ξ and update $\tilde{\xi}$ to ξ that attains the maximum. Here $E_{\underline{p}_\xi}[\cdot]$ denotes the expectation over the approximating posterior distribution (9) with the variational parameters $\tilde{\xi}$. This update is guaranteed to increase $\underline{Z}(\xi)$.

From eqs.(6) and (4), we have

$$\frac{\partial Q}{\partial \xi_i} = -l'(\xi_i)(E_{\underline{p}_{\tilde{\xi}}}[\bar{y}_i^2] - \xi_i^2).$$

Since $l'(\xi_i) < 0$, we obtain the variational parameter ξ_i that attains the maximum of Q ,

$$\xi_i^2 = E_{\underline{p}_{\xi}}[y_i^2] = (\beta \mathbf{S}_0 + 2n\mathbf{\Lambda})_{ii}^{-1} + m_i^2. \quad (14)$$

3.3 Variational Free Energy

Here we define the free energy F and the variational free energy $\overline{F}(\xi)$ by $F = -\log Z$ and $\overline{F}(\xi) = -\log \underline{Z}(\xi)$. Since $\overline{F}(\xi) \geq F$ follows from eq.(12), the difference between $\overline{F}(\xi)$ and F corresponds to the accuracy of approximation. The variational free energy is explicitly given by

$$\begin{aligned} \overline{F}(\xi) &= -\log \int \underline{p}_{\xi}(\mathbf{y}|\mathbf{x})p(\mathbf{y})d\mathbf{y} \\ &= \frac{1}{2} \log |\beta \mathbf{S}_0 + 2n\mathbf{\Lambda}| - \frac{1}{2} \log |\beta \mathbf{S}_0| - \sum_{i=1}^N \log \binom{n}{x_i} + \frac{\beta \mu_0^T \mathbf{S}_0 \mu_0}{2} \\ &\quad - \frac{\mathbf{m}^T (\beta \mathbf{S}_0 + 2n\mathbf{\Lambda}) \mathbf{m}}{2} - n \sum_{i=1}^N \left\{ \frac{\xi_i}{4} \tanh \frac{\xi_i}{2} - \log(2 \cosh \frac{\xi_i}{2}) \right\}, \end{aligned} \quad (15)$$

which we can monitor during the optimization process. It also provides an estimate (upper bound) of the true free energy F . This is an advantage of the usage of the variational free energy although we can employ another objective function to determine the variational parameters as long as it is easy to obtain.

3.4 Hyperparameter Estimation

Following a similar argument as in Section 3.2, we can update hyperparameter β so as to maximize the lower bound $\underline{Z}(\xi)$ by maximizing Q with respect to β . The inequality (12) implies that this procedure also increases the true marginal likelihood Z . Since

$$\frac{\partial Q}{\partial \beta} = \frac{N}{2\beta} - \frac{1}{2} [\text{Tr}\{\mathbf{S}_0 E_{\underline{p}_{\xi}}[\mathbf{y}\mathbf{y}^T]\} - 2\mu_0^T \mathbf{S}_0 E_{\underline{p}_{\xi}}[\mathbf{y}] + \mu_0^T \mathbf{S}_0 \mu_0] \quad (16)$$

$$= \frac{N}{2\beta} - \frac{1}{2} \text{Tr}[\mathbf{S}_0 \{\mathbf{S}^{-1} + (\mathbf{m} - \mu_0)(\mathbf{m} - \mu_0)^T\}], \quad (17)$$

we obtain the following update rule for the hyperparameter β ,

$$\beta = N/\text{Tr}[\mathbf{S}_0 \{\mathbf{S}^{-1} + (\mathbf{m} - \mu_0)(\mathbf{m} - \mu_0)^T\}]. \quad (18)$$

3.5 Estimation Algorithm

The resulting algorithm is summarized as follows:

1. Initialize $\xi = \{\xi_i\}_{i=1}^N$, β (for example, $\xi_i = 1$), and repeat 2, 3, 4 until convergence.
2. Update \mathbf{S} and \mathbf{m} by eqs. (10) and (11). (update of the posterior)

3. Update the variational parameters ξ by eq.(14). (improvement of the approximation)
4. Update β by eq. (18). (estimation of the hyperparameter)

At each iteration, we can monitor the variational free energy (15) which is guaranteed to decrease monotonically.

4 Example

In this section, we present an example where we apply the smoothness prior

$$p(\mathbf{y}) \propto \exp\left(-\frac{\beta}{2} \sum_{i=1}^{N-1} (y_{i+1} - y_i)^2\right) \quad (19)$$

which favors the smooth variation of the firing rate. Here $\mu_0 = \mathbf{0}$ and the matrix \mathbf{S}_0 is defined by

$$\mathbf{S}_0 = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}.$$

We generated a binary sequence ($n = 1$) of length $N = 300$ according to the true distribution $p(\mathbf{x}|\mathbf{y}^*)$ where $\mathbf{y}^* = \{y_i^* = y(\lambda_i^*)\}_{i=1}^N$ and $\lambda_i^* = 0.3 + 0.25 \sin(2\pi i/N)$. Using the sequence, we estimated the firing rate simultaneously optimizing the hyperparameter β . Figure 1 shows the binary sequence, the true rate $\lambda^* = \{\lambda_i^*\}_{i=1}^N$ and the estimated firing rate $\{y^{-1}(m_i)\}_{i=1}^N$ where m_i is the posterior mean.

We compare the computational time of the approximate Bayesian estimation method with that of the true Bayesian approach. We implemented the true Bayesian estimation by using a transfer matrix method[9] and calculated the true free energy F which we compared with the variational (approximate) free energy $\bar{F}(\xi)$ defined in Section 3.3.

Table 1 shows the computational time and the free energy obtained by the three approaches:

1. Approximate Bayesian estimation simultaneously optimizing the hyperparameter β
2. Approximate Bayesian estimation with the hyperparameter β fixed
3. True Bayesian estimation with the hyperparameter β fixed.

In 2 and 3, the hyperparameter β was fixed to the value obtained in 1.

We can see three things. Firstly, the computational time required for simultaneous optimization of the hyperparameter is several times that without hyperparameter optimization. Secondly, the computational time of the approximate Bayesian estimation is much faster than that of the true Bayesian approach.

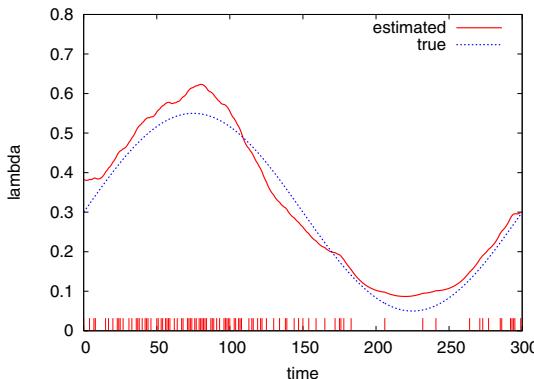


Fig. 1. An example of firing rate estimation: estimated rate (solid line), true rate (dotted line), binary sequence (impulse)

Table 1. Computational time and free energy

	β estimated	β fixed	true Bayes
Computational time	1.71	0.57	1197.20
Free energy	166.84	166.84	165.89

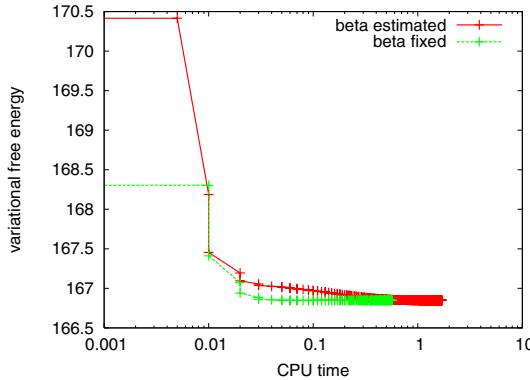


Fig. 2. An example of the behavior of the variational free energy as a function of CPU time (logarithmic scale) for the algorithm with the simultaneous estimation of β (solid line) and for that with fixed β (dashed line)

Lastly, the free energy obtained by the approximate Bayesian estimation is fairly close to that by the true Bayesian estimation, which implies the high accuracy of the approximation.

Figure 2 demonstrates the behavior of the variational free energy $\overline{F}(\xi)$ during the estimation process as a function of CPU time for the algorithm with

simultaneous estimation of β and that with fixed β . We can see that, for both algorithms, $\overline{F}(\xi)$ is monotonically decreasing and is already close to the limiting value after only a few iterations.

5 Conclusion

In this paper, we presented an approximate Bayesian estimation method for the event rate estimation from a sequence of event counts. We also provided an optimization procedure for estimating the hyperparameter which determines the smoothness of the event rate. Numerical simulation demonstrated the efficiency and the high accuracy of the proposed method. In future research, the performance of this method should be examined also in terms of the estimation accuracy of the true event rate.

Acknowledgements

This work was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research 20800012.

References

1. Arjas, E., Heikkilä, J.: An algorithm for nonparametric Bayesian estimation of a Poisson intensity. *Computational Statistics* 12, 385–402 (1997)
2. Bishop, C.: *Pattern Recognition and Machine Learning*. Springer, Heidelberg (2006)
3. Smith, A., Brown, E.: Estimating a state-space model from point process observations. *Neural Computation* 15, 965–991 (2003)
4. Davison, A., Ramesh, N.: Some models for discretized series of events. *Journal of the American Statistical Association* 91(434), 601–609 (1996)
5. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39-B, 1–38 (1977)
6. Jaakkola, T., Jordan, M.: Bayesian parameter estimation via variational methods. *Statistics and Computing* 10, 25–37 (2000)
7. Koyama, S., Shimokawa, T., Shinomoto, S.: Phase transitions in the estimation of event rate: a path integral analysis. *Journal of Physics A* 40, F383–F390 (2007)
8. Koyama, S., Shinomoto, S.: Empirical Bayes interpretations of random point events. *Journal of Physics A* 38, L531–L537 (2005)
9. Tanaka, H., Miura, K., Watanabe, K., Okada, M.: Estimation of an instantaneous spike rate with the transfer matrix method. Technical Report of IEICE, NC2007-96, vol. 107(413), pp. 55–60 (2008)

Sampling Curve Images to Find Similarities among Parts of Images

Kazunori Iwata and Akira Hayashi

Graduate School of Information Sciences, Hiroshima City University,
Hiroshima, 731-3194, Japan
kiwata@hiroshima-cu.ac.jp,
akira@hiroshima-cu.ac.jp

<http://www.robotics.im.hiroshima-cu.ac.jp/~kiwata/>,
<http://www.robotics.im.hiroshima-cu.ac.jp/~akira/>

Abstract. In statistical shape analysis, curve matching is often used to find correspondences between the sample points of a curve image and those of another curve image by using a dissimilarity measure of curve images. In this paper, we present a novel dissimilarity measure of curve images to be used in curve matching, together with a way of distributing sample points on each curve image. We prove that the dissimilarity measure has an asymptotic guarantee for finding a part of a curve image which is similar to a part of another one, with their respective sample points.

1 Introduction

Because shapes, line drawings, and characters consist of curve images in two dimensions, curve matching has frequently been used in statistical shape analysis for digital image processing, line drawing interpretation, and character handwriting recognition [12345678]. When a curve image is represented as an ordered sequence of points, it is called a directed curve image. To reduce computational cost for processing curve images, a curve image is often re-parameterized as a finite set of points sampled from it [1, Chap. 1] [4789]. Curve matching is used to find correspondences between the sample points of a curve image and those of another curve image by using a dissimilarity measure of curve images. In most cases, the curve images are alike in shape, but different because of various kinds of deformations such as articulation, occlusion, and jaggy [8].

In this paper, we concentrate on a specific but meaningful partial deformation defined by a similar relation. We present a novel dissimilarity measure to be used in curve matching, together with a way of sampling points from each directed curve image. We show that there is an asymptotic guarantee for finding a part of a directed curve image similar to a part of another one, with their respective sample points.

We describe some basics on curve matching in Sect. 2, and prove the main result in Sect. 3 followed by a summary of the paper in Sect. 4.

2 Basics on Curve Matching

Let \mathbb{Z} be the integers and let \mathbb{R} be the real numbers. The nonnegative and positive elements in \mathbb{Z} are denoted as \mathbb{Z}_0^+ and \mathbb{Z}^+ , respectively. For any $i, j \in \mathbb{Z}$ where $i \leq j$, the set of integers between i and j is denoted as $\mathbb{Z}_i^j \triangleq \{z \in \mathbb{Z} \mid i \leq z \leq j\}$. The nonnegative and positive elements in \mathbb{R} are denoted as \mathbb{R}_0^+ and \mathbb{R}^+ , respectively. The norm of a vector in Euclidean space is denoted as $\|\cdot\|$.

2.1 Curves, Images and Sample Points

We distinguish curve images from curves for the development of theory.

Definition 1 (Regular Curve). Let $I = [a, b] \subset \mathbb{R}$ be a closed interval, where $a < b$. A plane curve is denoted as a twice differentiable map $C_I : I \rightarrow \mathbb{R}^2$, with

$$C_I(t) \triangleq (x_I(t), y_I(t)). \quad (1)$$

When a time-parameter $t \in I$ increases from a to b , we obtain the directed trajectory of $C_I(t)$,

$$C_I(I) \triangleq \{C_I(t) \mid t \in I\}, \quad (2)$$

where the ordering of coordinates in $C_I(I)$ preserves that of $t \in I$. The curve image which is an ordered set of coordinates with respect to t is simply called an image. A plane curve C_I which satisfies

$$\frac{dC_I(t)}{dt} \triangleq \dot{C}_I(t) \neq 0, \quad (3)$$

for all $t \in (a, b)$ is said to be regular, and its image is called a regular image.

Note that the ordering of coordinates in an image $C_I(I)$ where $I = [a, b]$ starts from $C_I(a)$ and ends at $C_I(b)$.

Definition 2 (Image Set). \mathbb{S} denotes the set of images whose length is positive.

To reduce computational cost for processing the images, an image is often re-parameterized as an ordered set of points sampled from it [4, 7, 8, 9]. In addition, this serves to avoid sensitivity to the local perturbation of the images.

Definition 3 (Sample Points). For any interval $I = [a, b]$ and any $N \in \mathbb{Z}^+$, let

$$\gamma_N(I) \triangleq \{\{t_0, t_1, \dots, t_{N-1}, t_N\} \in I^{N+1} \mid a = t_0 < t_1 < \dots < t_{N-1} < t_N = b\}. \quad (4)$$

For any sequence $T_N = \{t_0, \dots, t_N\} \in \gamma_N(I)$,

$$C_I(T_N) \triangleq \{C_I(t_i) \in C_I(I) \mid i \in \mathbb{Z}_0^N\}, \quad (5)$$

is called the sample points of $C_I(I)$, where the ordering of coordinates in $C_I(T_N)$ preserves that of $t_i \in I$.

Definition 4 (Re-parameterization). We denote the set of possible sample points on an image $C_I(I) \in \$$ by

$$\Gamma_N(C_I(I)) \triangleq \left\{ C_I(T_N) \in C_I(I)^{N+1} \mid T_N \in \gamma_N(I) \right\}. \quad (6)$$

For any sequence $T_N = \{t_0, \dots, t_N\} \in \gamma_N(I)$, the sample points on the image are simply denoted as

$$P_N \triangleq C_I(T_N). \quad (7)$$

For all $i \in \mathbb{Z}_0^N$, the i -th element of P_N is denoted as

$$p_i \triangleq C_I(t_i), \quad (8)$$

and the components of the i -th element are written as

$$(x_i, y_i) \triangleq p_i. \quad (9)$$

For all $i \in \mathbb{Z}_0^{N-1}$, the finite difference at p_i is defined as

$$\Delta p_i = (\Delta x_i, \Delta y_i), \quad (10)$$

$$\triangleq (x_{i+1} - x_i, y_{i+1} - y_i). \quad (11)$$

For all $i \in \mathbb{Z}_0^{N-2}$, the second-order finite difference at p_i is written as

$$\Delta^2 p_i = (\Delta^2 x_i, \Delta^2 y_i), \quad (12)$$

$$\triangleq (\Delta x_{i+1} - \Delta x_i, \Delta y_{i+1} - \Delta y_i). \quad (13)$$

For all $i \in \mathbb{Z}_0^{N-1}$, the unit tangent and the unit normal vectors at p_i are defined as

$$e_{P_N}^{(1)}(p_i) \triangleq \frac{1}{\|\Delta p_i\|} (\Delta x_i, \Delta y_i), \quad (14)$$

$$e_{P_N}^{(2)}(p_i) \triangleq \frac{1}{\|\Delta p_i\|} (-\Delta y_i, \Delta x_i), \quad (15)$$

respectively. For all $i \in \mathbb{Z}_0^{N-2}$, the curvature at p_i is defined as

$$\kappa_{P_N}(p_i) \triangleq \frac{\Delta x_i \Delta^2 y_i - \Delta^2 x_i \Delta y_i}{\|\Delta p_i\|^3}. \quad (16)$$

In this sense, the image $C_I(I)$ is re-parameterized with respect to sample points P_N . For notational simplicity, an image $C_I(I)$ is denoted as \mathcal{C} whenever we do not need to show its curve explicitly.

Definition 5 (Equipartition Sample Points). Let \mathcal{C} be an image in $\$$. For any sample points $P_N \in \Gamma_N(\mathcal{C})$, let p_i denote the i -th element of P_N . If for all $i \in \mathbb{Z}_0^{N-1}$, the finite difference at p_i is

$$\|\Delta p_i\| = r_N > 0, \quad (17)$$

then P_N is referred to as equipartition sample points of \mathcal{C} . For any $N \in \mathbb{Z}^+$, the set of such sample points on \mathcal{C} is simply denoted as

$$\Gamma_N^*(\mathcal{C}) \triangleq \left\{ P_N \in \mathcal{C}^{N+1} \mid \|\Delta p_i\| = r_N, i \in \mathbb{Z}_0^{N-1} \right\}. \quad (18)$$

Note that r_N depends only on N but not on i .

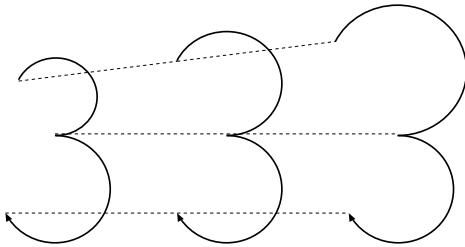


Fig. 1. Images in partially similar relation

2.2 Partially Similar Relation

When an image is uniformly enlarged or reduced, the resulting images are similar to the original image in the following sense.

Definition 6 (Similar Relation). Let \mathcal{C} and \mathcal{C}' be images in \mathbb{S} . If there exist a map $\zeta : \mathcal{C} \rightarrow \mathcal{C}'$ and a constant $\lambda \in \mathbb{R}^+$ such that for all $c_1, c_2 \in \mathcal{C}$,

$$\|\zeta(c_1) - \zeta(c_2)\| = \lambda \|c_1 - c_2\|, \quad (19)$$

then \mathcal{C} and \mathcal{C}' are similar images and this binary relation on \mathbb{S} is expressed as

$$\mathcal{C} \sim \mathcal{C}'. \quad (20)$$

Similarity plays an important role in human recognition of images, because similar images appear to be the same in shape, though they are different in scale. For example, a small digit “3” and a large digit “3” are recognized as the same digit. In curve matching, we sometimes need to find a part of an image which is similar to a part of another image. This relation is formulated in Definition 7.

Definition 7 (Partially Similar Relation). Let \mathcal{C} and \mathcal{C}' be images in \mathbb{S} . If there exist their parts $\underline{\mathcal{C}} \subseteq \mathcal{C}$ and $\underline{\mathcal{C}'} \subseteq \mathcal{C}'$ in \mathbb{S} such that

$$\underline{\mathcal{C}} \sim \underline{\mathcal{C}'}, \quad (21)$$

then \mathcal{C} and \mathcal{C}' are called partially similar images.

The purpose of this paper is to provide an effective measure for finding similar parts of images by sampling points from their respective images.

Example 1 (Partially Similar Relation). The center in Fig. 1 is an image of the digit “3”. On the left of the figure is an image deformed by uniformly making the upper part smaller. On the right of the figure is an image deformed by uniformly making the upper part larger. Accordingly, they are partially similar images.

3 Main Result

Our dissimilarity measure is based on the curvature-based measure of an image, defined next.

Definition 8 (Curvature-based Measure). Let \mathcal{C} be an image in \mathbb{S} . Let $\underline{\mathcal{C}} \subseteq \mathcal{C}$ denote a part of the image in \mathbb{S} . For any sample points $P_N \in \Gamma_N(\mathcal{C})$, the measure $\alpha_{P_N} : \mathbb{S} \rightarrow \mathbb{R}$ is defined as

$$\alpha_{P_N}(\underline{\mathcal{C}}) \triangleq \sum_{p_i \in \underline{\mathcal{C}} \cap P_N, i \in \mathbb{Z}_0^{N-2}} \kappa_{P_N}(p_i), \quad (22)$$

where κ_{P_N} is the curvature defined in (16).

Definition 9 (Dissimilarity Measure of Images). Let \mathcal{C} and \mathcal{C}' be images in \mathbb{S} . Let $\underline{\mathcal{C}} \subseteq \mathcal{C}$ and $\underline{\mathcal{C}'} \subseteq \mathcal{C}'$ denote their parts in \mathbb{S} . For any sample points $P_N \in \Gamma_N(\mathcal{C})$ and $Q_M \in \Gamma_M(\mathcal{C}')$, the dissimilarity measure $\mu_{P_N, Q_M} : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}_0^+$ of them is defined as

$$\mu_{P_N, Q_M}(\underline{\mathcal{C}}, \underline{\mathcal{C}'}) \triangleq |\alpha_{P_N}(\underline{\mathcal{C}}) - \alpha_{Q_M}(\underline{\mathcal{C}'})|. \quad (23)$$

The proof that this dissimilarity measure is a metric is routine. Note that computing the dissimilarity measure is simple, since it needs only the curvatures.

We are now in a position to give our main result.

Theorem 1. Let \mathcal{C} and \mathcal{C}' be images in \mathbb{S} . Let $\underline{\mathcal{C}} \subseteq \mathcal{C}$ and $\underline{\mathcal{C}'} \subseteq \mathcal{C}'$ denote their parts which are regular images in \mathbb{S} . If $\underline{\mathcal{C}} \sim \underline{\mathcal{C}'}$ and the following constraints on their sample points hold,

1. $P_N = \{p_0, \dots, p_N\} \in \Gamma_N^*(\mathcal{C})$,
2. $Q_M = \{q_0, \dots, q_M\} \in \Gamma_M^*(\mathcal{C}')$, and
3. for any $\epsilon \in \mathbb{R}^+$, there exist $N_0 \in \mathbb{Z}^+$ and $M_0 \in \mathbb{Z}^+$ such that for all $N \geq N_0$ and all $M \geq M_0$, all $i \in \mathbb{Z}_0^{N-1}$, and all $j \in \mathbb{Z}_0^{M-1}$,

$$|1/\|\Delta p_i\| - 1/\|\Delta q_j\|| < \epsilon, \quad (24)$$

then

$$\lim_{N, M \rightarrow \infty} \mu_{P_N, Q_M}(\underline{\mathcal{C}}, \underline{\mathcal{C}'}) = 0. \quad (25)$$

Proof. Let c_a and c_b denote the first and last coordinates of $\underline{\mathcal{C}}$, respectively. Let ℓ_a and ℓ_b be the tangential lines at the first and last coordinates of $\underline{\mathcal{C}}$, respectively. Let O be the intersection point of the tangential lines. We use θ_a, θ_b to denote the angles between lines ℓ_a, ℓ_b and the horizontal axis respectively, as shown in Fig. 2. Since the sum of the interior angles of any triangle is π , we have

$$\pi - \angle c_a O c_b = \theta_b - \theta_a. \quad (26)$$

Since P_N denotes equipartition sample points (constraint 1), for all $i \in \mathbb{Z}_0^{N-1}$, let

$$\|\Delta p_i\| = r_N. \quad (27)$$

We define θ_{P_N} such that for all $i \in \mathbb{Z}_0^{N-1}$, the unit tangent vector at $p_i \in P_N$ is

$$e_{P_N}^{(1)}(p_i) = (\cos \theta_{P_N}(p_i), \sin \theta_{P_N}(p_i)). \quad (28)$$

Because constraint 1 gives

$$\lim_{N \rightarrow \infty} \inf \left\{ |\theta_{P_N}(p_i) - \theta_a| \in \mathbb{R}_0^+ \mid p_i \in \underline{\mathcal{C}} \cap P_N, i \in \mathbb{Z}_0^{N-2} \right\} = 0, \quad (29)$$

$$\lim_{N \rightarrow \infty} \inf \left\{ |\theta_{P_N}(p_i) - \theta_b| \in \mathbb{R}_0^+ \mid p_i \in \underline{\mathcal{C}} \cap P_N, i \in \mathbb{Z}_0^{N-2} \right\} = 0, \quad (30)$$

we see that

$$\lim_{N \rightarrow \infty} \sum_{p_i \in \underline{\mathcal{C}} \cap P_N, i \in \mathbb{Z}_0^{N-2}} \Delta \theta_{P_N}(p_i) = \theta_b - \theta_a, \quad (31)$$

where

$$\Delta \theta_{P_N}(p_i) \triangleq \theta_{P_N}(p_{i+1}) - \theta_{P_N}(p_i). \quad (32)$$

Since for all $i \in \mathbb{Z}_0^{N-2}$,

$$\lim_{N \rightarrow \infty} \kappa_{P_N}(p_i) = \lim_{N \rightarrow \infty} \frac{\Delta \theta_{P_N}(p_i)}{r_N}, \quad (33)$$

we obtain

$$\lim_{N \rightarrow \infty} \alpha_{P_N}(\underline{\mathcal{C}}) = \lim_{N \rightarrow \infty} \sum_{p_i \in \underline{\mathcal{C}} \cap P_N, i \in \mathbb{Z}_0^{N-2}} \frac{\Delta \theta_{P_N}(p_i)}{r_N}. \quad (34)$$

Next, since $\underline{\mathcal{C}} \sim \underline{\mathcal{C}'}$, from Frenet-Serret formulae [10, Sect. 2], we can displace $\underline{\mathcal{C}'}$ by transform $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ of parallel displacement and rotation such that it is tangent to both ℓ_a and ℓ_b as shown in Fig. 2. Let

$$\underline{\mathcal{C}''} \triangleq \{ \phi(c') \in \mathbb{R}^2 \mid c' \in \underline{\mathcal{C}'} \}, \quad (35)$$

where ϕ preserves the ordering of coordinates in $\underline{\mathcal{C}'}$. Those points of tangency are c_a'' and c_b'' . The angle between Oc_a'' and Oc_b'' is also

$$\pi - \angle c_a'' O c_b'' = \theta_b - \theta_a. \quad (36)$$

From constraint 2, for all $j \in \mathbb{Z}_0^{M-1}$, let

$$\|\Delta q_j\| = r'_M. \quad (37)$$

In the same way as above, letting

$$\Delta \theta'_{Q_M}(q_j) \triangleq \theta'_{Q_M}(q_{j+1}) - \theta'_{Q_M}(q_j), \quad (38)$$

we obtain

$$\lim_{M \rightarrow \infty} \sum_{q_j \in \underline{\mathcal{C}'} \cap Q_M, j \in \mathbb{Z}_0^{M-2}} \Delta \theta'_{Q_M}(q_j) = \theta_b - \theta_a, \quad (39)$$

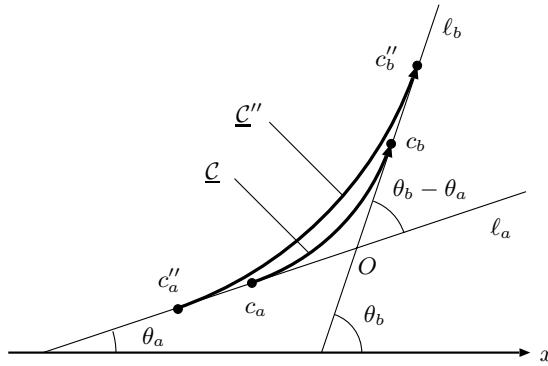


Fig. 2. Similar Parts of Images

because ϕ does not affect $\Delta\theta'_{Q_M}$. We also have

$$\lim_{M \rightarrow \infty} \alpha_{Q_M} (\underline{\mathcal{C}'}) = \lim_{M \rightarrow \infty} \sum_{q_j \in \underline{\mathcal{C}'} \cap Q_M, j \in \mathbb{Z}_0^{M-2}} \frac{\Delta\theta'_{Q_M}(q_j)}{r'_M}. \quad (40)$$

Because constraint 3 means

$$\lim_{N, M \rightarrow \infty} |1/r_N - 1/r'_M| = 0, \quad (41)$$

equations (31) and (39) can give

$$\lim_{N, M \rightarrow \infty} |\alpha_{P_N} (\mathcal{C}) - \alpha_{Q_M} (\underline{\mathcal{C}'})| = 0. \quad (42)$$

Using Definition 9, this may be rewritten to provide the conclusion (25).

Theorem 1 means that under the given constraints, the dissimilarity measure has an asymptotic guarantee for finding similar parts of images with their respective sample points. This is because if two images are partially similar, then we can find similar image parts such that the dissimilarity measure of them tends to zero as shown in (25). The constraints show a concrete way of sampling points from their respective images.

4 Summary

We discussed a partial deformation given by a similarity relation. We proposed a dissimilarity measure to be used in curve matching, together with a way of sampling points from directed curve images. We described an asymptotic guarantee that it finds a part of directed curve image, which is similar to a part of another one, with their respective sample points.

Acknowledgments

This work was supported in part by Grants-in-Aid 18300078, 18700157, and 20500139 for scientific research from the Ministry of Education, Culture, Sports, Science, and Technology, Japan.

References

1. Dryden, I.L., Mardia, K.V.: Statistical Shape Analysis. Wiley series in probability and mathematical statistics. Wiley, Chichester (1998)
2. Gdalyahu, Y., Weinshall, D.: Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21(12), 1312–1328 (1999)
3. Ablameyko, S., Pridmore, T.: Machine interpretation of line drawing images: technical drawings, maps, and diagrams. Springer, Berlin (2000)
4. Sebastian, T.B., Klein, P.N., Kimia, B.B.: On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(1), 116–125 (2003)
5. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5), 550–571 (2004)
6. Srivastava, A., Joshi, S.H., Mio, W., Liu, X.: Statistical shape analysis: Clustering, learning, and testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(4), 590–602 (2005)
7. Belongie, S., Mori, G., Malik, J.: Matching with shape contexts. In: Krim, H., Yezzi Jr., A. (eds.) *Statistics and Analysis of Shapes. Modeling and Simulation in Science, Engineering and Technology*, pp. 81–105. Birkhäuser, Boston (2006)
8. Manay, S., Cremers, D., Hong, B.W., Yezzi Jr., A.J., Soatto, S.: Integral invariants for shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(10), 1602–1618 (2006)
9. Powell, M.J.D.: An optimal way of moving a sequence of points onto a curve in two dimensions. *Computational Optimization and Applications* 13(1–3), 163–185 (1999)
10. Mokhtarian, F., Mackworth, A.K.: A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(8), 789–805 (1992)

Improving the State Space Organization of Untrained Recurrent Networks

Michal Čerňanský¹, Matej Makula¹, and Ľubica Beňušková²

¹ Faculty of Informatics and Information Technologies, STU Bratislava, Slovakia

² Department of Computer Science, University of Otago, Dunedin, New Zealand

{cernansky,makula}@fiit.stuba.sk, lubica@cs.otago.ac.nz

Abstract. Recurrent neural networks are frequently used in cognitive science community for modeling linguistic structures. More or less intensive training process is usually performed but several works showed that untrained recurrent networks initialized with small weights can be also successfully used for this type of tasks. In this work we demonstrate that the state space organization of untrained recurrent neural network can be significantly improved by choosing appropriate input representations. We experimentally support this notion on several linguistic time series.

1 Introduction

Many commonly used real-world data with time structure can be expressed as a sequence of symbols from finite alphabet like symbolic time series. Since their emergence, neural networks were applied to symbolic time series analysis. Especially popular is to use connectionist models for processing of complex language structures.

Recurrent neural networks (RNNs) are frequently applied to processing symbolic sequences. Elman's simple recurrent network (SRN) [1] is the most frequently used architecture. Common algorithms usually used for RNN training are based on gradient minimization of the output error. Several advanced approaches have been proposed, especially popular are methods based on the Kalman filtration.

But thorough training process may not be necessary. In [2] we have studied the dynamics of untrained recurrent network randomly initialized with small weights. We have shown that for some tasks the contractive dynamics of such a recurrent network is sufficient and the structural organization of the state space can be used to create prediction machines having comparable performance to thoroughly trained SRNs. We have shown and explained strong correspondence between such a prediction machines and a class of Markov models – variable length Markov models (VLMMs) [3]. On the other side carefully trained RNNs can usually achieve significantly better performance.

A lot of attention is now being focused on connectionist models known under the name “reservoir computing”. The most prominent example of these approaches is a recurrent neural network architecture called an echo state network (ESN). ESNs were successfully applied in multiple sequence processing tasks [4] and recently several authors have also used ESNs for language modeling. In [5] ESN was used to the next symbol prediction task on the “Little Red Riding Hood” fairy tale. In [6] authors studied performance of ESN and SRN in modeling natural-like artificial languages. Authors

claim that ESN show similar predictive performance to SRN, superior to Markov models of various orders (n-grams). In [78] authors also used ESNs to process natural-like languages but in the context of studying systematicity, that is defined as the ability to process novel sometimes even ungrammatical sentences.

In [9] we have studied the state space organization of the recurrent neural network before and after training on three artificial languages. We found that the dynamics of the trained network is still based on the fixed point attractors, but these attractors do not correspond to the symbols representing the words as is the case of the untrained network. Instead attractors correspond to the word categories enabling the network to better manage its state space. This behavior was achieved by the training process by setting input weights sourced from the inputs of the same category to the similar values.

Input representations, i.e. weights spreading information from active input neuron are usually randomly initialized in models based on the contractive dynamics such as ESNs. In this work we study and evaluate word co-occurrence method of initialization of input weights on two models based on random contractive dynamics: Neural prediction machines (NPMs) and echo state networks (ESNs). We compare their predictive performance on two linguistic datasets and elaborate upon importance of input representations on the state space organization in these models. Such an enhanced ESNs were already studied in [10, 11], but in the context of the generalization and systematicity. Following [10] we denote models with enhanced input representations by the “plus” sign.

2 Neural Prediction Machines

Techniques frequently used in RNN state space analysis are clusterization of the recurrent units activities and attributing meaningful information to the extracted clusters. Neural prediction machines [2] are models using clusters extracted from RNN state space as predictive contexts. The symbol just presented to the network corresponds to some cluster if and only if the RNN state belongs to this cluster. The next symbol probability of a symbol $a \in \mathcal{A}$ for a cluster c is calculated by relating the number of times (counter N_c^a) when symbol a follows a symbol driving the RNN to a given cluster c with the number of times the RNN state belongs to the cluster c ($\sum_{b \in \mathcal{A}} N_c^b$). The next symbol probability distributions are smoothed by applying Laplace correction parameter γ , hence probability of predicting symbol a when in cluster c can be calculated as:

$$P_{NPM}(a|c) = \frac{P(a, c)}{P(c)} \doteq \frac{\gamma + N_c^a}{\gamma A + \sum_{b \in \mathcal{A}} N_c^b}. \quad (1)$$

We set the Laplace correction parameter γ to the value of A^{-1} . This can be seen as if we initialized counters to the value γ prior to counting any symbol occurrences thus attributing some probabilities also to symbols not present in the training sequence. The more important the context, the less smoothing is performed. On the other hand, the probability distribution of rare (statistically less convincing) context is smoothed more heavily.

3 Echo State Networks

Echo state network is formed of one huge recurrent layer composed of hundreds or even thousands sparsely interconnected units. Input and recurrent connections initialized randomly with small weights and are not modified in the training phase. Only output weights are trained usually using simple linear regression. Simple and fast training is the most appealing feature of ESNs.

Experiments with ESNs were done in similar way as described in [12]. Output units had linear activation function, recursive least squares were used to train the network. When $\mathbf{u}(t)$ is an input vector at time step t , activations of internal units were updated according to

$$\mathbf{x}(t) = f_{\text{hid}} (\mathbf{W}^{\text{in}} \cdot \mathbf{u}(t) + \mathbf{W} \cdot \mathbf{x}(t-1) + \mathbf{W}^{\text{back}} \cdot \mathbf{y}(t-1)), \quad (2)$$

where f is the internal unit's activation function, \mathbf{W} , \mathbf{W}^{in} and \mathbf{W}^{back} are hidden-hidden, input-hidden, and output-hidden connections' matrices, respectively. Activations of output units are calculated as

$$\mathbf{y}(t) = f_{\text{out}} (\mathbf{W}^{\text{out}} \cdot [\mathbf{u}(t), \mathbf{x}(t), \mathbf{y}(t-1)]), \quad (3)$$

where \mathbf{W}^{out} is hidden-output and output-output connections' matrix.

The next symbol probabilities $p(t)$ in time step t were calculated from activities on output units. First activities smaller than specified minimal value o_{\min} were set to o_{\min} and then probabilities were estimated by normalizing output activities:

$$\hat{o}_i(t) = \begin{cases} o_{\min} & \text{if } o_i(t) < o_{\min} \\ o_i(t) & \text{otherwise} \end{cases}, \quad (4)$$

$$p(t) = \frac{\hat{o}_i(t)}{\sum_j \hat{o}_j(t)}, \quad (5)$$

where $o_i(t)$ is the activity of the output unit i in time t . o_{\min} was set to 0.001 throughout all experiments.

4 Extracting Input Representations

An enhancement of ESN denoted as ESN+ was proposed in [10]. In this modification of ESN input weights were not initialized randomly but simple word co-occurrence statistics was used [13]. The value of each input weight $W_{i,j}^{\text{in}}$ was set to the ratio:

$$R_{i,j} = N \cdot \frac{N(i,j) + N(j,i)}{N(i) \cdot N(j)} \quad (6)$$

where $N(i,j)$ is the number of times symbol i and j occur next to each other and N is the length of training sequence.

Since dimensions of matrices \mathbf{R} and \mathbf{W}^{in} are usually different remaining elements of \mathbf{W}^{in} were set to zeros [10]. In our work we have slightly modified this approach and we initialized input weights as $\mathbf{W}^{\text{in}} = \mathbf{T}^{\text{rand}} \cdot \mathbf{R}$, where \mathbf{T}^{rand} is transformation matrix created with small random values from $(-0.01, 0.01)$. Better results than with original method were achieved.

5 Datasets

The first dataset we used was the so called “Elman’s Grammar” (EG) [14]. In [6] ESN’s ability to learn grammatical structures was studied using this language. Dataset alphabet A is composed 24 words including end-of-string marker. The training set was composed of 10000 sentences (55273 symbols) and the test set of independently generated 10000 sentences (54966 symbols). The entropy estimated on the test set is $H = 0.534$.

Language generated by complex stochastic context-free grammar (CG)¹ was used in experiments with models using self-organization [15]. Grammar was inspired by child-directed sentences with added recursive structures. The language vocabulary was composed of 72 words including the end-of-string marker. Similarly to [15] the training set was composed of 5000 randomly generated sequences (30985 symbols) and the test set comprises 10000 randomly generated sequences (62387 symbols). The estimated entropy is $H = 0.398$.

6 Experiments and Results

The predictive performance was evaluated using normalized negative log-likelihood (NNL). NNL was calculated as:

$$\text{NNL} = -\frac{1}{T} \sum_{t=1}^T \log_{|A|} P_t(s_t) \approx -\frac{1}{T} \sum_{t=1}^T \sum_{a \in A} G_t(a) \log_{|A|} P_t(a), \quad (7)$$

where the base of the logarithm is the alphabet size $|A|$, s_t is the symbol in the test set in time step t . $P_t(a)$ stands for the probability of predicting symbol a in the time step t and $G_t(a)$ is the ground true probability for symbol a in time step t . NNL can be seen as the compression ratio, NNL of 0.5 means that original sequence can be compressed to the half of its original size. Language entropy H can be estimated as:

$$H = \frac{1}{T} \sum_{t=1}^T \sum_{a \in A} G_t(a) \log_{|A|} G_t(a), \quad (8)$$

and can be seen as the best achievable NNL of a perfect model.

We tested ESNs with reservoirs of different sizes. Other parameters such as spectral radius, the sparsity of input and hidden weights, and input weight range are of much less importance when processing symbolic sequences. For ESN we provide results for spectral radius of 0.98. The probability of existence of input weight was 1.0 and their values were chosen randomly (uniform distribution) from interval (-1.0, 1.0). We found out that the probability p_{rec} of existence of recurrent weight (determining sparsity of the recurrent weight matrix) has very little influence on ESN performance. To ensure full connectivity for small networks and a reasonable computation time for large networks we set $p_{\text{rec}} = \min(1, 10/n)$, where n is the number of reservoir units.

ESN+ seemed to be more sensitive to parameter settings. For EG dataset p_{rec} was increased to $p_{\text{rec}} = \min(1, 50/n)$ and for CG dataset it was set to 1.0. Also the spectral

¹ Generously provided by Igor Farkaš.

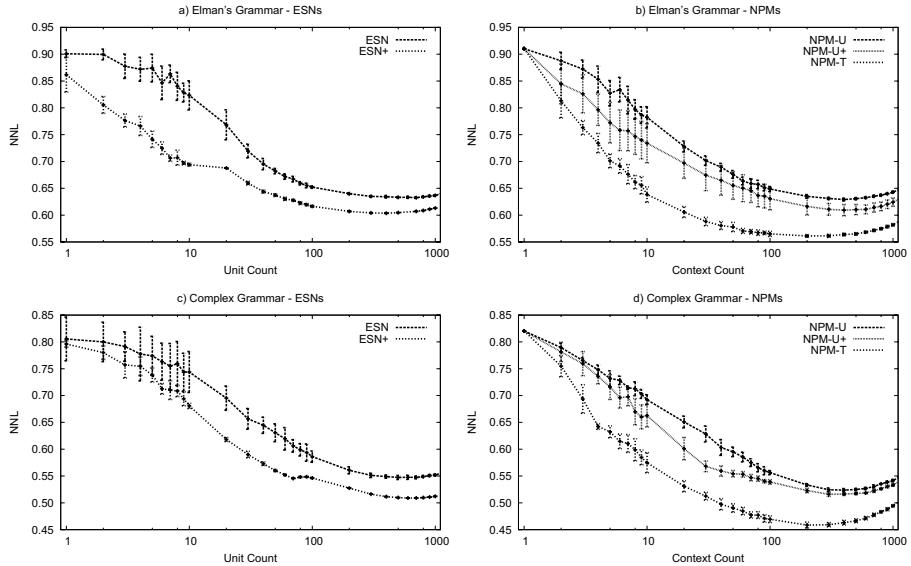


Fig. 1. Predictive performance of ESNs (left) and NPMs (right) on EG (top) and CG (bottom)

radius of ESN+ for CG dataset was changed to the value of 0.5. Results are presented in Fig. 1a and Fig. 1c. ESN+ significantly outperforms ESN model.

NPM models with context count varying from 1 to 1000 were tested. NPM-U was built over untrained and NPM-T was built over trained recurrent part of SRN with 16 hidden units. The model with modified input representations is denoted as NPM-U+. We present mean and standard deviations of 10 simulations performed with the same parameters. NPM-T was build over SRNs carefully trained using Kalman filtration [16].

Not surprisingly NPM-Ts performed significantly better than NPM-Us. Careful adaptation process affected the RNN state space organization through adaptation of both input and recurrent weights and the trained SRNs (or models built over the trained SRNs) outperform other models. But improvement of NPM-U+ over NPM-U is also visible (mostly the case of EG dataset). Results of NPM models are presented in Fig. 1b and Fig. 1d.

7 Discussion and Conclusion

Results reveal an improvement of predictive performance of ESNs and NPM-Us can be achieved by creating proper input representations. For both EG and CG datasets ESN+ and NPM-U+ models performed significantly better than their counterparts consistently for all numbers of predictive contexts. Welch modification of unpaired Student's t-test was used to evaluate statistical significance and almost all comparisons proved to be significant at $p = 0.05$ and great majority of comparisons of models with context count above 10 proved to be significant at $p = 0.001$. In the following we attempt to analyze state space representations of studied models to justify these preliminary results. We

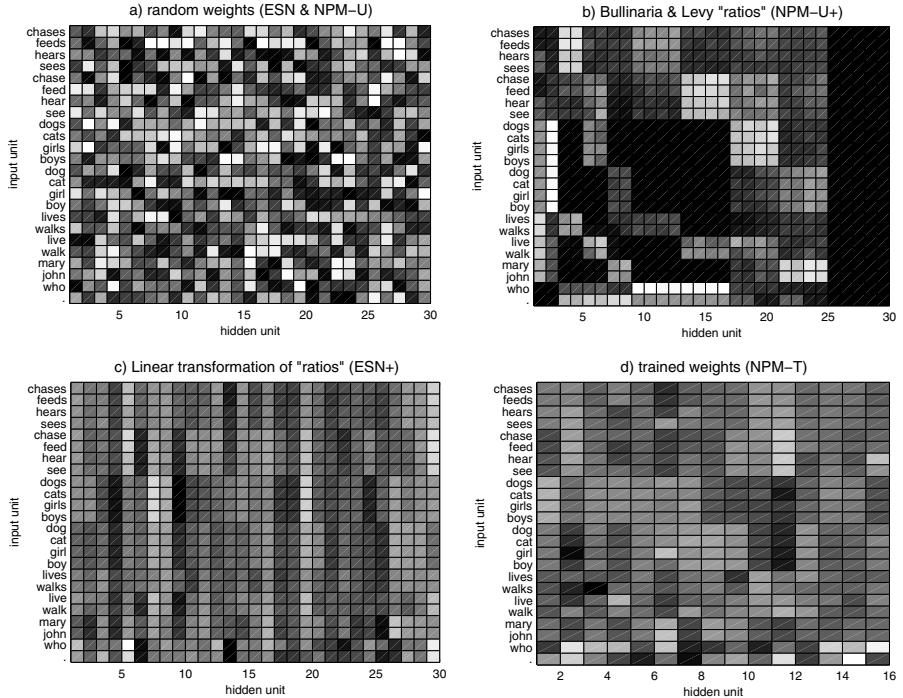


Fig. 2. Input representations of words from EG dataset created: a) randomly; b) Bullinaria & Levy "ratios" as used in [10]; c) linear transformation of "ratios" used in this work; d) by RNN training

start by showing the impact of proper input representations by visualizing input weights initialized by different methods used in this paper (Fig. 2).

Standard ESN and NPM-U are built over untrained network, where input weights are chosen randomly (Fig. 2a). In the case of ESN+ and NPM-U+ co-occurrence statistics according to Bulinaria & Levy [13] is used (Fig. 2b). This results in similar input vectors for words from the same grammatical category (e.g. walks and lives, john and mary, boy and girl, etc.). In the case of ESN+ we apply linear transformation to scale down input weight values (Fig. 2c). Similar representation of input words can be also seen in NPM-T, i.e. NPM built over network trained by extended Kalman filter (Fig. 2d).

However, even when input representations in ESN+, NPM-U+ and NPM-T are identical, predictive performance of NPM-T is superior. To analyze impact of input representation on the network state space organization more precisely we locate and analyze fixed point in each model used in this paper. There were three typical configurations of fixed points as shown in Fig. 3.

The contractive dynamics of ESN and NPM-U models is simple. With fixed input, activities of hidden units converge to a single attractive fixed point in the network state space. Different input codes of words from \mathcal{A} serve as bias which pushes attractive points in different directions. If input codes are initialized randomly, distances between attractors are random (Fig. 3a). When co-occurrence statistics is used, similar input

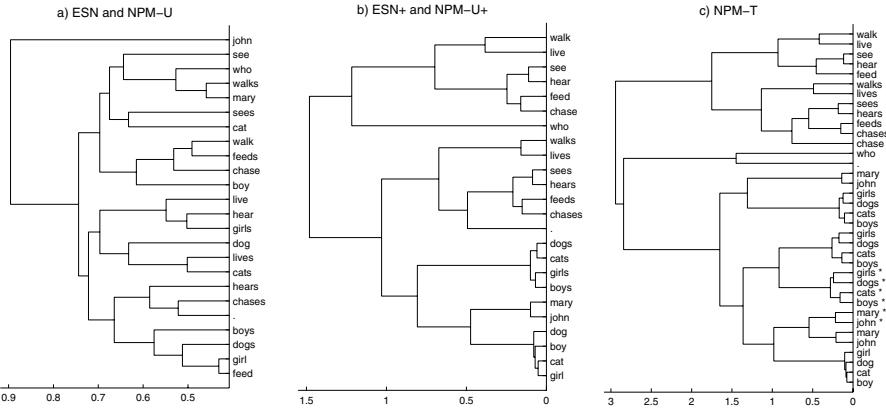


Fig. 3. Distance of fixed points in the network dynamics: a) ESN and NPM-U; b) ESN+ and NPM-U+ c) NPM-T. Several input words in NPM-T employ multiple fixed point in their dynamics. (e.g. dynamics of words *john* and *mary* is composed of two attractors and one unstable saddle.)

vectors push attractive fixed points in the same direction. It means that the positions of attractors are still random, but now they are clustered according similar input words (Fig. 3b).

Finally, analysis of NPM-T revealed that the dynamics of underlying trained networks was not based on single attractive points. In Fig. 3c dynamics of several input words is composed of two attractors and one saddle combination (e.g. john, mary, girls, boys, etc.). Adaptation of both input and recurrent weights allows network to change contractive character of network dynamics and thus outperform other models.

In this work we build on results found in [9], where SRNs achieved better predictive performance by creating proper input representations during training process. Here we studied performance of models using contrastive dynamics of untrained recurrent networks with inputs representations extracted from word co-occurrence statistics as described in [13]. Extraction method may not be optimal and more extensive study should be performed. Nevertheless models with extracted input representations perform significantly better than their counterparts with random input representations. Similar results but in the context of systematicity study were achieved in [10][11].

Acknowledgments. This work was supported by the grants Vega 1/0848/08 and Vega 1/0822/08.

References

1. Elman, J.L.: Finding structure in time. *Cognitive Science* 14(2), 179–211 (1990)
 2. Tiňo, P., Černánský, M., Beňušková, Ľ.: Markovian architectural bias of recurrent neural networks. *IEEE Transactions on Neural Networks* 15(1), 6–15 (2004)
 3. Ron, D., Singer, Y., Tishby, N.: The power of amnesia. *Machine Learning* 25, 117–149 (1996)
 4. Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* 304(5667), 78–80 (2004)

5. Jaeger, H.: Short term memory in echo state networks. Technical Report GMD 152, German National Research Center for Information Technology (2001)
6. Tong, M.H., Bickett, A.D., Christiansen, E.M., Cottrell, G.W.: Learning grammatical structure with Echo State Networks. *Neural Networks* 20, 424–432 (2007)
7. Frank, S.L.: Strong systematicity in sentence processing by an Echo State Network. In: Kollias, S., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 505–514. Springer, Heidelberg (2006)
8. Frank, S.L.: Learn more by training less: systematicity in sentence processing by recurrent networks. *Connection Science* 18, 287–302 (2006)
9. Tiňo, P., Čerňanský, M., Beňušková, Ľ.: Organization of the state space of a simple recurrent neural network before and after training on recursive linguistic structures. *Neural Networks* 20, 236–244 (2007)
10. Frank, S.L., Čerňanský, M.: Generalization and systematicity in echo state networks. In: Proceedings of the 30th Cognitive Science Conference, Washington, DC, USA, pp. 733–738 (2008)
11. Frank, S.L., Jacobsson, H.: Sentence processing in echo state networks: a qualitative analysis by finite state machine extraction. *Journal of Algorithms* (2008) (submitted)
12. Jaeger, H.: The “echo state” approach to analysing and training recurrent neural networks. Technical Report GMD 148, German National Research Center for Information Technology (2001)
13. Bullinaria, J.A., Levy, J.P.: Extracting semantic representations from word co-occurrence statistics: a computational study. *Behavior Research Methods* 39, 510–526 (2007)
14. Elman, J.: Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning* 7, 195–225 (1991)
15. Farkaš, I., Crocker, M.: Recurrent networks and natural language: exploiting self-organization. In: Proceedings of the 28th Cognitive Science Conference, Vancouver, Canada, pp. 1275–1280 (2006)
16. Čerňanský, M., Beňušková, Ľ.: Simple recurrent network trained by RTRL and extended Kalman filter algorithms. *Neural Network World* 13(3), 223–234 (2003)

Online Multibody Factorization Based on Bayesian Principal Component Analysis of Gaussian Mixture Models

Kentarou Hitomi^{1,2}, Takashi Bando², Naoki Fukaya², Kazushi Ikeda¹,
and Tomohiro Shibata¹

¹ Graduate School of Information Science, Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara, Japan
{kenta-hi,kazushi,tom}@is.naist.jp

² DENSO CORPORATION, 1-1, Showa-cho, Kariya, Aichi 4488661, Japan
{kentarou.hitomi,takashi.bandou,naoki.fukaya}@denso.co.jp

Abstract. An online multibody factorization method for recovering the shape of each object from a sequence of monocular images is proposed. We formulate multibody factorization problem of data matrix of feature positions as the parameter estimation of the mixtures of probabilistic principal component analysis (MPPCA) and use the variational inference method as an estimation algorithm that concurrently performs classification of each feature points and the three-dimensional structures of each object. We also apply the online variational inference method make the algorithm suitable for real-time applications.

1 Introduction

The structure from motion (SFM) has attracted much attention in the computer vision community. SFM is a problem of estimating the three-dimensional structures and the camera motion from a sequence of monocular images. The SFM method satisfies a social requirement because it is simple and low-cost. For instance, it is applicable to mobile phones and vehicle video systems.

One of the most popular solutions for the SFM problem is the factorization method proposed by Tomasi and Kanade [1] that decomposes a data matrix \mathbf{x} comprising of two-dimensional feature positions tracked over frames into two matrices each of which corresponds to the target shape and the camera motion, respectively. The key benefit of the factorization method is that it is based on Singular Value Decomposition (SVD) and thus numerically stable. A series of developments on factorization methods have been conducted. Costeira et al. [2] developed a multibody factorization method that classifies feature positions out of multiple objects and estimates the structure of each object by introducing the *shape interaction matrix* which again makes use of SVD. Because the factorization method is a batch method that works only after all images are acquired and that requires a high computational cost, Morita et al. developed a sequential factorization method [3] that updates its estimates at each frame by computing

shape space, the three most dominant eigenvectors of the covariance matrix of feature positions.

In contrast to those algebraic developments, Okatani [4] indicated that a probabilistic model called mixtures of probabilistic principal component analysis (MPPCA) enables straightforward subspace estimation with feature position uncertainty in the data incorporated. PPCA is originally a formulation of PCA as a parameter estimation problem of a probabilistic generative model that projects latent variables to a data space. Bishop [5] has developed a PPCA solution with variational inference. PPCA has been extended to MPPCA model where data is generated from multiple latent spaces. Solving MPPCA concurrently performs classification of mixed data and fitting the subspaces into each latent space. The EM algorithm [6] and the variational inference method [7] were developed for the MPPCA modeling. Variational inference [8] is an approximate calculation of the computationally intractable posterior distribution. As an extension of the variational inference method, Sato [9] proposed its online version that computes the approximated posterior distribution at each observation.

This article proposes an online multibody factorization method with variational inference employing MPPCA model. As indicated by Okatani, multibody factorization can be achieved by an MPPCA modeling; Affine camera projection of multiple objects corresponds to the projection from multiple subspaces to data space. The online variational inference enables incremental acquisition of the multibody structures as well as camera motion, which is suitable for real-time applications.

2 Online Multibody Factorization

2.1 Factorization with the MPPCA Model

The factorization method [1] assumes the affine camera model which is a linear projection from a 3D space to a two-dimensional image plane. The three-dimensional structure, shape of an object, is represented by the three-dimensional positions of P features $\mathbf{z} = [\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(P)}] \in \mathbb{R}^{3 \times P}$. Let $\mathbf{x}_f = [\mathbf{x}_f^{(1)}, \dots, \mathbf{x}_f^{(P)}] \in \mathbb{R}^{2 \times P}$ be feature positions on the image plane at frame f . Then the generative model of $\mathbf{x}_f^{(p)}$ with the observation noise $\boldsymbol{\varepsilon}_f^{(p)} \in \mathbb{R}^{2 \times 1}$ can be written as $\mathbf{x}_f^{(p)} = [\mathbf{W}_f, \boldsymbol{\mu}_f] [\mathbf{z}^{(p)T}, 1]^T + \boldsymbol{\varepsilon}_f^{(p)} = \mathbf{W}_{e,f} \mathbf{z}_e^{(p)} + \boldsymbol{\varepsilon}_f^{(p)}$, where $\mathbf{W}_f \in \mathbb{R}^{3 \times 2}$ is a projection matrix and $\boldsymbol{\mu}_f \in \mathbb{R}^{3 \times 1}$ is a translation vector. Observing P feature points over F frames, generative model for data matrix $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_P^T]^T \in \mathbb{R}^{2F \times P}$ is written as

$$\mathbf{x} = \mathbf{W}_e \mathbf{z}_e + \boldsymbol{\varepsilon} \quad (1)$$

where $\mathbf{W}_e = [\mathbf{W}_{e,1}^T \cdots \mathbf{W}_{e,F}^T]^T \in \mathbb{R}^{2f \times 4}$ is called the motion matrix and $\mathbf{z}_e = [\mathbf{z}_e^{(1)} \cdots \mathbf{z}_e^{(P)}] \in \mathbb{R}^{4 \times P}$ is called the shape matrix. The factorization method factorizes \mathbf{x} into two matrices so that $\mathbf{x} \doteq \hat{\mathbf{W}}_e \hat{\mathbf{z}}_e$, where $\hat{\mathbf{W}}_e = [\hat{\mathbf{W}}, \hat{\boldsymbol{\mu}}]$, $\hat{\mathbf{z}}_e = [\hat{\mathbf{z}}_e^{(1)}, \dots, \hat{\mathbf{z}}_e^{(P)}]$, $\hat{\mathbf{z}}_e^{(p)} = [\hat{\mathbf{z}}^{(p)T}, 1]^T$. However, this decomposition is not unique.

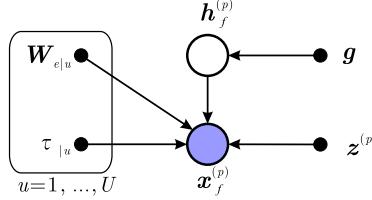


Fig. 1. A graphical model of MPPCA. p -th feature position $\mathbf{x}_f^{(p)}$ at frame f is generated. Circle nodes are stochastic variables and dot nodes are model parameters. Only $\mathbf{x}_f^{(p)}$ is observable.

In fact, if \mathbf{T} is any invertible 3×3 matrix, $\hat{\mathbf{W}}_e \mathbf{T}$, and $\mathbf{T}^{-1} \hat{\mathbf{z}}_e$ are also a valid decomposition of \mathbf{x} . To calculate the true motion and shape matrices, \mathbf{T} is determined under the constraint imposed by the camera model (Ref. [2] for orthogonal camera model, for instance). PPCA [10] can be employed for modeling the linear projection expressed by (1). PPCA has an extension to MP-PCA that can be used to model class-conditional densities and hence be applied to classification of feature points in the multibody case. The prior distribution over each elements of ε is assumed to be mutually independent, and follows the univariate normal distribution with mean 0 and variance τ^{-1} . Then, $p(\mathbf{x}_f^{(p)} | \mathbf{z}^{(p)}) = N(\mathbf{x}_f^{(p)} | \mathbf{W}_e \mathbf{z}_e^{(p)}, \tau^{-1} \mathbf{I}_2)$, where $N(\mathbf{x} | \mathbf{m}, \Sigma)$ denotes a multivariate normal distribution function over \mathbf{x} with mean \mathbf{m} and covariance matrix Σ .

Probabilistic multibody factorization can be modeled by the MPPCA [6, 7] where a projection is stochastically selected from U linear projection units (1). To formulate the selection of a projection, MPPCA introduces an additional U -dimensional binary latent variable $\mathbf{h}_f^{(p)} = (h_{f,1}^{(p)}, \dots, h_{f,U}^{(p)})^T$ having 1-of- K coding in which a particular element $h_{f,u}^{(p)}$ is equal to 1 and all other element is equal to 0 describing which of the U units in the mixture is responsible for generating $\mathbf{x}_f^{(p)}$. The discrete distribution over $\mathbf{h}_f^{(p)}$ is specified in terms of the mixing proportion $\mathbf{g} = (g_1, \dots, g_U)^T$, such that $p(h_{f,u}^{(p)} = 1) = g_u$. Fig. 1 is a graphical model for MPPCA. Let $\mathbf{h} = [\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(P)}]$ and $\mathbf{W}_{e|u}$ be projection matrix of unit u . MPPCA gives estimates of model parameter $\boldsymbol{\theta} = \{\{\mathbf{W}_{e|u}, \tau_{|u}\}_{u=1}^U, \mathbf{g}\}$ and stochastic variables \mathbf{h}, \mathbf{z} , by computing the posterior distribution given the data,

$$p(\boldsymbol{\theta}, \mathbf{z}, \mathbf{h} | \mathbf{x}) = p(\mathbf{x} | \boldsymbol{\theta}, \mathbf{z}, \mathbf{h}) p_0(\boldsymbol{\theta}, \mathbf{z}, \mathbf{h}) / p(\mathbf{x}) . \quad (2)$$

As in [7], the prior distributions $p_0(\mathbf{z})$ and $p_0(\boldsymbol{\theta}) = p_0(\mathbf{g}) \prod_{u=1}^U p_0(\mathbf{W}_{e|u}) p_0(\tau_{|u})$ are given by

$$p_0(\mathbf{g}) = \text{Dir}(\mathbf{g} | \gamma_{\mathbf{g},0}), \quad p_0(\mathbf{z}^{(p)}) = N(\mathbf{z}^{(p)} | \mathbf{0}_{3 \times 1}, \mathbf{I}_3),$$

$$p_0(\mathbf{W}_{e|u}) = \prod_{i=1}^{2F} p_0(\tilde{\mathbf{w}}_{e,i|u}) = \prod_{i=1}^{2F} N(\tilde{\mathbf{w}}_{e,i|u} | \mathbf{m}_{\tilde{\mathbf{w}}_{e,i,0|u}}, \tau_{|u} \text{diag}(\boldsymbol{\alpha}_{e|u})^{-1}),$$

$$p_0(\tau_{|u}) = \text{Gam}(\tau_{|u} | \gamma_{\tau_{|u},0}, \gamma_{\tau_{|u},0} / m_{\tau_{|u},0}),$$

where $\boldsymbol{\gamma}_{\mathbf{g},0} = (\gamma_{g_1}, \dots, \gamma_{g_U})$ and $\tilde{\mathbf{w}}_{e,i|u}$ is a column vector corresponding to the i -th row of $\mathbf{W}_{e|u}$. Here, $\text{Gam}(x | a, b)$ denotes a Gamma distribution over x and $\text{Dir}(\mathbf{x} | \boldsymbol{\gamma}_{\mathbf{x}})$ is a Dirichlet distribution over \mathbf{x} which are given by

$$\text{Gam}(x | a, b) = \frac{1}{\Gamma(a)} b^a x^{a-1} e^{-bx}, \text{Dir}(\mathbf{x} | \boldsymbol{\gamma}_{\mathbf{x}}) = \frac{\Gamma(U + \sum_{u=1}^U \gamma_{x_u})}{\Gamma(\gamma_{x_1} + 1) \cdots \Gamma(\gamma_{x_U} + 1)} \prod_{u=1}^U x_u^{\gamma_{x_u}}$$

where $\boldsymbol{\gamma}_{\mathbf{x}} = (\gamma_{x_1}, \dots, \gamma_{x_U})$. However, computing the posterior over \mathbf{z} and $\boldsymbol{\theta}$ is intractable. The variational inference method introduced in next section can give an approximate posterior distribution.

2.2 Variational Inference

Here, the test distribution $q(\boldsymbol{\theta}, \mathbf{h}, \mathbf{z}) = q(\boldsymbol{\theta})q(\mathbf{h}, \mathbf{z})$ is introduced. It is shown that minimization of Kullback-Leibler divergence from q to the true posterior distribution $p(\boldsymbol{\theta}, \mathbf{h}, \mathbf{z} | \mathbf{x})$ is equivalent to maximizing the free energy \mathcal{F} ,

$$\mathcal{F}(\mathbf{x}, q(\boldsymbol{\theta}, \mathbf{h}, \mathbf{z})) = \langle \ln \frac{p_0(\mathbf{h}, \mathbf{z}, \mathbf{x} | \boldsymbol{\theta})}{q(\mathbf{h}, \mathbf{z})} \rangle_{q(\boldsymbol{\theta}), q(\mathbf{h}, \mathbf{z})} + \langle \ln \frac{p_0(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \rangle_{q(\boldsymbol{\theta})} \quad (3)$$

Let $\mathbf{J} = \{\boldsymbol{\theta}, \mathbf{h}, \mathbf{z}, \mathbf{x}\}$. Variational inference introduces the following constraints on q which maximizes \mathcal{F} .

$$q(\mathbf{h}, \mathbf{z}) = \frac{\exp(\langle \ln p_0(\mathbf{J}) \rangle_{q(\boldsymbol{\theta})})}{\int \exp(\langle \ln p_0(\mathbf{J}) \rangle_{q(\boldsymbol{\theta})}) d\mathbf{h} d\mathbf{z}}, q(\boldsymbol{\theta}) = \frac{\exp(\langle \ln p_0(\mathbf{J}) \rangle_{q(\mathbf{h}, \mathbf{z})})}{\int \exp(\langle \ln p_0(\mathbf{J}) \rangle_{q(\mathbf{h}, \mathbf{z})}) d\boldsymbol{\theta}} \quad (4)$$

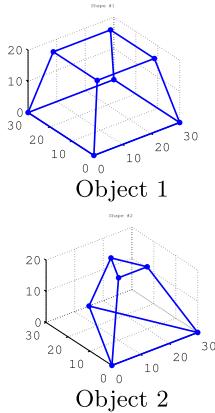
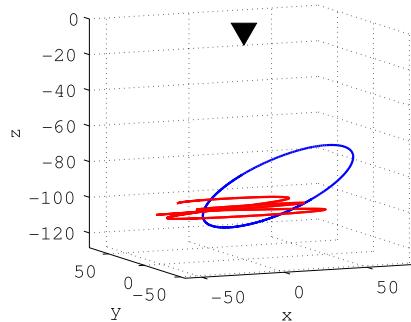
The closed-form solutions result from these constraints are implicit for each factor since the moment of $q(\mathbf{h}, \mathbf{z})$ and $q(\boldsymbol{\theta})$ are mutually dependent. Sato [9] has shown, however, that iterative calculations of constraints of (4) are equivalent to conducting a gradient method with respect to the free energy (3). Thus the batch factorization method of PPCA based on variational inference ,VPCA, can be derived.

2.3 Online VPCA

An online version of VPCA is derived by replacing the batch free energy (3) with the one defined as follows [9], with forgetting factor λ and normalization constant $\eta_{f_c} = \{\sum_{f=1}^{f_c} \lambda^{f_c-f}\}^{-1}$,

$$\begin{aligned} \mathcal{F}^\lambda(\{\mathbf{x}_f\}_{f=1}^{f_c}, \{q_f(\boldsymbol{\theta}, \mathbf{h}_f, \mathbf{z}_f)\}_{f=1}^{f_c}) \\ \equiv \eta_{f_c} \sum_{f=1}^{f_c} \left\{ \lambda^{f_c-f} \langle \ln \frac{p_0(\mathbf{h}_f, \mathbf{z}_f, \mathbf{x}_f | \boldsymbol{\theta})}{q_f(\mathbf{h}_f, \mathbf{z}_f)} \rangle_{q_{f_c}(\boldsymbol{\theta}), q_f(\mathbf{h}, \mathbf{z})} \right\} + \langle \ln \frac{p_0(\boldsymbol{\theta})}{q_{f_c}(\boldsymbol{\theta})} \rangle_{q_{f_c}(\boldsymbol{\theta})}, \end{aligned} \quad (5)$$

where $q_f(\boldsymbol{\theta}, \mathbf{h}_f, \mathbf{z}_f) = q_f(\boldsymbol{\theta})q_f(\mathbf{h}_f, \mathbf{z}_f)$ is the test distribution based on incremental observations up to frame f and f_c is the current frame. Comparing the online free energy (5) with the batch one (3), each statistics $\mathbf{r}(\mathbf{h}, \mathbf{z}, \mathbf{x})$ in (3) is

**Fig. 2.** Object shapes**Fig. 3.** Centroids of the two objects in the 3D space. Blue and red lines are trajectories of Objects 1 and 2.

replaced with the weighted sum of $\mathbf{r}_f = \mathbf{r}(\mathbf{h}_f, \mathbf{z}_f, \mathbf{x}_f)$ for $f = 1, \dots, f_c$, where \mathbf{r}_f is calculated from $q_f(\mathbf{h}_f, \mathbf{z}_f)$. It follows that the gradient method with respect to the online free energy is obtained by the same replacement of statistics in the constraints shown in Fig. 2. The weighted sum of sufficient statistics at each frame is calculated incrementally as follows.

$$\langle\langle \mathbf{r} \rangle\rangle_{f_c} = (1 - \eta_{f_c}) \langle\langle \mathbf{r} \rangle\rangle_{f_c-1} + \eta_{f_c} \mathbf{r}_{f_c} \quad (6)$$

The normalization constant is also calculated incrementally as $\eta_{f_c} = \{1 + \lambda \eta_{f_c-1}\}^{-1}$. This online variational inference of MPPCA realizes online multibody factorization.

3 Experiment

The aim of this section is to compare the performances of Batch VPCA, Online VPCA, and Costeira's SVD-based multibody factorization method.

3.1 Synthetic Data

The synthetic data of 300 frames was prepared as follows. The scene consists of two independently-moving objects with no background. These objects are truncated pyramids shown in Fig. 2. The feature points in images were generated with the paraperspective camera model. Fig. 3 shows the trajectories of the centroids of the two objects. All feature points can be always observed transparently even when the self-body or the other's body occludes some feature points. To test that each feature is correctly classified into its true object when such occlusions occur temporarily on the image plane, the synthetic trajectories shown in Fig. 4

were prepared. The blue and red lines depict the trajectories of Objects 1 and 2, respectively. The observation noise $\varepsilon_f^{(p)} \sim N(0, 0.1^2)$ are exemplified in Fig. 5. This figure shows examples of true the positions of all the feature points and their noisy observations from 50-th to 53rd frame, where the true positions of Objects 1 and 2 are shown by stars and triangles, and noisy observations by dots. In the experiment of Online VPCA, the observation of 60 frames consisting of data of the current frame and preceding 59 frames were used to estimate the posterior.

3.2 Costeira's Multibody Factorization Method

Costeira's method [2] classifies feature points based on the shape interaction matrix $\mathbf{Q} = \mathbf{V}_{:,1:r} \mathbf{V}_{:,1:r}^T$ where $\mathbf{V}_{:,1:r}$ is a matrix comprising r columns of \mathbf{V} which is acquired by SVD as $\mathbf{x} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$. Though r is the rank of data matrix \mathbf{x} determined depending on the uncertainty provided by a statistically modeled feature tracker, we set $r = 8$ since feature tracking is not dealt here.

3.3 Results

Fig. 6 shows the estimated shapes by all methods. Note that the result of Online VPCA was acquired after observing 300 frames. Squares, stars, and circles depict the results obtained by Costeira's method, Batch VPCA method, and Online VPCA method, respectively. To generate this figure, each shape \hat{s} is calculated by affine transform of \hat{z} by least square fitting of \hat{s} to each true shape s according to $\hat{s} = s\hat{z}^T(\hat{z}\hat{z}^T)^{-1} \cdot \hat{z}$. Fig. 7 shows the root-mean-square (RMS) error of Online VPCA estimation, where the estimation error decreases as observations increases. Fig. 8 compares the means and the standard deviations of the RMS errors averaged over 100 trials. The means of the RMS errors of the two objects were about 6.97 and 0.84 by the Costeira's method, and about 0.31 and 0.87 by Batch VPCA method, that is, the performance of Batch VPCA

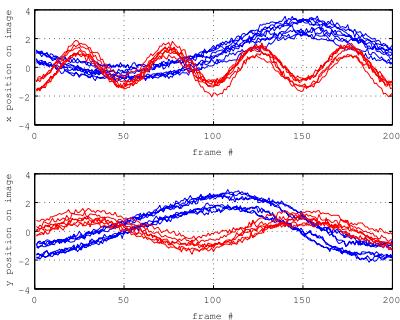


Fig. 4. Trajectories of feature points on image plane

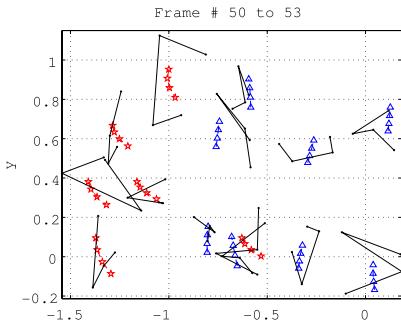


Fig. 5. Examples of the true feature positions and their noisy observations on image plane

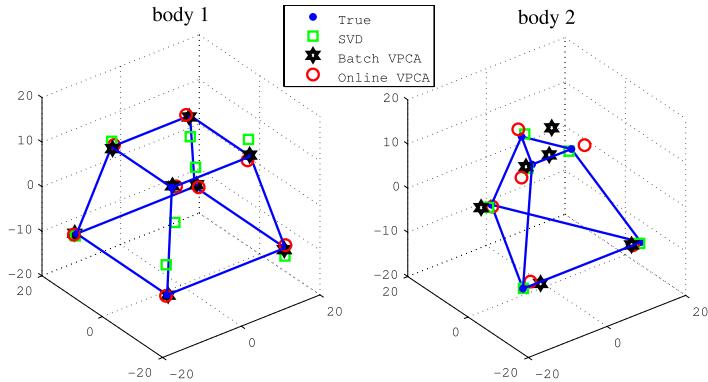


Fig. 6. Recovered 3D structuresD

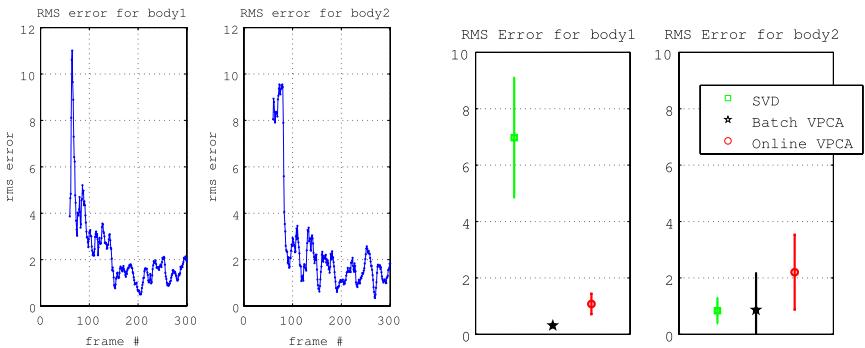


Fig. 7. RMS error by Online VPCA

Fig. 8. Means and standard deviations of RMS error

method is equivalent to, or could be higher than the Costeira's method. The performance of Online VPCA achieved an accuracy of the RMS error, where the means are about 1.07 and 2.20 for each object, which is comparable to Batch VPCA method.

4 Conclusion

In this article, we formulated the multibody structure-from-motion (SfM) problem as a parameter estimation of the mixture probabilistic principle component analysis (MPPCA) with variational inference. This probabilistic modeling approach enabled simultaneous processing of the multibody classification and factorization of each object in a principled way, and achieved an equivalent or even higher accuracy to the Costeira's method based on a non-probabilistic matrix factorization in our simulation. We further implemented an online version of

the variational inference which realized online multibody factorization with a comparable accuracy to the batch version.

Armed with the Bayesian PPCA framework, the following two improvements can, at least, be done in the near future. First, an incremental update of the number of objects can be achieved as Sato [9] indicated, although the number of objects was assumed to be known in this study. Second, the problem of missing observations is unavoidable in the SFM because of noise, occlusion, and/or frame out. Estimation of the missing observations can be accomplished as Oba et al. [11] showed.

References

1. Tomasi, C., Kanade, T.: Shape from motion from image streams under orthography: A factorization method. *IJCV* 9(2), 137–154 (1992)
2. Costeira, J.P., Kanade, T.: A multibody factorization method for independently moving objects. *IJCV* 29(3), 159–179 (1998)
3. Morita, T., Kanade, T.: A sequential factorization method for recovering shape and motion from image streams. *PAMI* 19(8), 858–867 (1997)
4. Okatani, T.: A probabilistic approach to linear subspace fitting for computer vision problems. In: *CVPR Workshops* (2004)
5. Bishop, C.M.: Variational principal components. In: *ICANN*, vol. 1, pp. 509–514 (1999)
6. Tipping, M.E., Bishop, C.M.: Mixtures of probabilistic principal component analysers. *Neur. Comp.* 11(2), 443–482 (1999)
7. Bishop, C.M., Winn, J.M.: Non-linear bayesian image modelling. In: Vernon, D. (ed.) *ECCV 2000. LNCS*, vol. 1842, pp. 3–17. Springer, Heidelberg (2000)
8. Attias, H.: A variational bayesian framework for graphical models. In: *NIPS*, vol. 12, pp. 206–212 (2000)
9. Sato, M.: Online model selection based on the variational bayes. *Neur. Comp.* 13, 1649–1681 (2001)
10. Tipping, M.E., Bishop, C.M.: Probabilistic component analysis. *Journal of the Royal Statistical Society Series B* 61, Part 3, 611–622 (1999)
11. Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., Ishii, S.: A bayesian missing value estimation method for gene expression profile data. *Bioinformatics* 19(16), 2088–2096 (2003)

A Appendix; Online Variational Solution for MPPCA

Batch update equations are obtained calculating constraints (4) of variational inference based on prior joint distribution $p_0(\boldsymbol{\theta}, \mathbf{h}, \mathbf{z}, \mathbf{x}) = p_0(\mathbf{x}, \mathbf{z}, \mathbf{h}|\boldsymbol{\theta}) \cdot p_0(\boldsymbol{\theta})$. The test distribution for parameter $\boldsymbol{\theta}$ becomes $q(\boldsymbol{\theta}) = q(\mathbf{g}) \cdot \prod_{u=1}^U \{q(\mathbf{W}_{e|u}|\tau_{|u}) \cdot q(\tau_{|u})\}$ where $q(\mathbf{g})$ is Dirichlet distribution, $q(\mathbf{W}_{e|u}|\tau_{|u})$ is Gaussian distribution, and $q(\tau_{|u})$ is Gamma distribution. Update equations based on the fist constraint in Eq.(4) is called E-Step, and update equations based on the second constraint is called M-Step. Update equations of online version is obtained by replacement of terms which are dependent on \mathbf{h} , \mathbf{z} or \mathbf{x} with

weighted summation of corresponding statistics that decays older estimation. Equations of online algorithm are shown below.

E-Step : Parameterizing each test distributions as

$$q(\mathbf{z}^{(p)} | \mathbf{h}^{(p)}) = \prod_{u=1}^U N(\mathbf{z}_{|u}^{(p)} | \mathbf{m}_{\mathbf{z}_{|u}^{(p)}}, \boldsymbol{\Sigma}_{\mathbf{z}_{|u}^{(p)}})^{h_u^{(p)}}, q(\mathbf{h}^{(p)}) = \exp(c^{(p)}) \cdot \exp\left(\sum_{u=1}^U h_u^{(p)} c_u^{(p)}\right),$$

these parameters are subject to following constraints.

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{z}_{|u}} &= (m_{\tau_{|u}} \langle \mathbf{W}_{|u}^T \mathbf{W}_{|u} \rangle_{q(\boldsymbol{\theta})} + \mathbf{I}_{q_{|u}})^{-1} \\ \mathbf{m}_{\mathbf{z}_{|u}} &= \boldsymbol{\Sigma}_{\mathbf{z}_{|u}} (m_{\tau_{|u}} \langle \mathbf{W}_{|u} \rangle^T \mathbf{x}^{(p)} - m_{\tau_{|u}} \langle \mathbf{W}_{|u}^T \boldsymbol{\mu}_{|u} \rangle_{q(\boldsymbol{\theta})}) \\ c_u^{(p)} &= -\frac{d}{2} \ln 2\pi + \frac{d}{2} \langle \ln \tau_{|u} \rangle_{\boldsymbol{\theta}} - \frac{1}{2} m_{\tau_{|u}} \langle \mathbf{x}^{(p)} \rangle^T \mathbf{x}^{(p)} + m_{\tau_{|u}} \text{Tr}(\langle \boldsymbol{\mu}_{|u} \rangle_{q(\boldsymbol{\theta})}^T \mathbf{x}^{(p)}) \\ &\quad - \frac{1}{2} m_{\tau_{|u}} \langle \boldsymbol{\mu}_{|u}^T \boldsymbol{\mu}_{|u} \rangle_{q(\boldsymbol{\theta})} + \frac{1}{2} \text{Tr}(\mathbf{m}_{\mathbf{z}_{|u}^{(p)}} \mathbf{m}_{\mathbf{z}_{|u}^{(p)}}^T \boldsymbol{\Sigma}_{\mathbf{z}_{|u}}^{-1}) + \frac{1}{2} \ln |\boldsymbol{\Sigma}_{\mathbf{z}_{|u}}| + \langle \ln g_u \rangle_{\mathbf{g}} \\ c^{(p)} &= -\ln \sum_{u=1}^U \exp(c_u^{(p)}) \end{aligned}$$

Then it follows that $\langle \mathbf{h}^{(p)} \rangle_{q(\mathbf{h}^{(p)})} = \exp(c^{(p)}) \cdot (\exp(c_1^{(p)}), \dots, \exp(c_U^{(p)}))^T$.

M-Step : Parameterizing each test distributions as

$$q(\mathbf{g}) = \text{Dir}(\mathbf{g} | \boldsymbol{\gamma}_{\mathbf{g}}), \boldsymbol{\gamma}_{\mathbf{g}} = (\gamma_{g_1}, \dots, \gamma_{g_U}), q(\mathbf{W}_{e|u} | \tau_{|u}) = \prod_{i=1}^d N(\tilde{\mathbf{w}}_{e,i|u} | \mathbf{m}_{\tilde{\mathbf{w}}_{e,i|u}}, \boldsymbol{\Sigma}_{\tilde{\mathbf{w}}_{e,i|u}}),$$

$$q(\tau_{|u}) = \text{Gam}(\tau_{|u} | \gamma_{\tau_{|u}}, \gamma_{\tau_{|u}} / m_{\tau_{|u}}),$$

these parameters are subject to following constraints.

$$\begin{aligned} \gamma_{g_u} &= \langle \langle \sum_{p=1}^P \langle h_{f,u}^{(p)} \rangle_{q_f(\mathbf{h})} \rangle \rangle_{f_c} + \gamma_{g_u,0} \\ \boldsymbol{\Delta}_{|u} &= \text{diag}(\langle \boldsymbol{\alpha}_{e,u} \rangle_{q(\boldsymbol{\alpha})}) + \langle \langle \sum_{p=1}^P \langle h_{f,u}^{(p)} \mathbf{z}_{e,f|u}^{(p)} \mathbf{z}_{e,f|u}^{(p)T} \rangle_{q_f(\mathbf{h}, \mathbf{z})} \rangle \rangle_{f_c} \\ \mathbf{M}_{\mathbf{W}_{e|u}} &= \{ \langle \langle \sum_{p=1}^P \mathbf{x}^{(p)} \langle h_u^{(p)} \mathbf{z}_{e|u}^{(p)T} \rangle_{q_f(\mathbf{h}, \mathbf{z})} \rangle \rangle_{f_c} + \mathbf{M}_{\mathbf{W}_{e|u},0} \text{diag}(\langle \boldsymbol{\alpha}_{e,u} \rangle_{\boldsymbol{\alpha}}) \} \boldsymbol{\Delta}_{|u}^{-1} \\ \boldsymbol{\Sigma}_{\tilde{\mathbf{w}}_{e|u}} &= \tau_{|u}^{-1} \boldsymbol{\Delta}_{|u}^{-1}, \quad \gamma_{\tau_{|u}} = \frac{d}{2} \langle \langle \sum_{p=1}^P \langle h_u^{(p)} \rangle_{q_f(\mathbf{h})} \rangle \rangle_{f_c} + \gamma_{\tau_{|u},0} \\ \gamma_{\tau_{|u}} / m_{\tau_{|u}} &= \frac{1}{2} \langle \langle \sum_{p=1}^P \langle h_u^{(p)} \rangle_{q_f(\mathbf{h})} \mathbf{x}^{(p)T} \mathbf{x}^{(p)} \rangle \rangle_{f_c} - \frac{1}{2} \text{Tr}\{\mathbf{M}_{\mathbf{W}_{e|u}}^T \mathbf{M}_{\mathbf{W}_{e|u}} \boldsymbol{\Delta}_{|u}\} \\ &\quad + \frac{1}{2} \text{Tr}\{\mathbf{M}_{\mathbf{W}_{e|u},0}^T \mathbf{M}_{\mathbf{W}_{e|u},0} \text{diag}(\langle \boldsymbol{\alpha}_{e,u} \rangle_{\boldsymbol{\alpha}_u})\} + \frac{\gamma_{\tau_{|u},0}}{m_{\tau_{|u},0}} \end{aligned}$$

The weighted summation $\langle \langle \cdot \rangle \rangle_{f_c}$ is updated incrementally according to Eq.(6).

Experimental Study of Ergodic Learning Curve in Hidden Markov Models

Masashi Matsumoto¹ and Sumio Watanabe²

¹ Department of Computational Intelligence and Systems Science,
Tokyo Institute of Technology,
4259, Nagatsuta, Midori-ku, Yokohama, 226-8503 Japan
masashimatzumoto@cs.pi.titech.ac.jp

² Precision and Intelligence Laboratory, Tokyo Institute of Technology,
4259, Nagatsuta, Midori-ku, Yokohama, 226-8503 Japan
swatanab@pi.titech.ac.jp

Abstract. A number of learning machines used in information science are not regular, but rather singular, because they are non-identifiable and their Fisher information matrices are singular. Even for singular learning machines, the learning theory was developed for the case in which training samples are independent. However, if training samples have time-dependency, then learning theory is not yet established. In the present paper, we define an ergodic generalization error for a time-dependent sequence and study its behavior experimentally in hidden Markov models. The ergodic generalization error is clarified to be inversely proportional to the number of training samples, but the learning coefficient depends strongly on time-dependency.

1 Introduction

Hidden Markov models are widely used in time sequence recognition and nonlinear text analysis. In speech recognition and gene analysis in particular, they are among the most appropriate statistical models. Therefore, several information systems have been constructed that employ hidden Markov models. Although their effectiveness in practical applications is well known, the mathematical properties of hidden Markov models remain unknown, because they are singular learning machines. A learning machine is said to be regular if the mapping from the parameter to the probability distribution is one-to-one and its Fisher information matrix is always positive definite. If a learning machine is not regular, then it is said to be singular. A number of learning machines and statistical models are singular, including normal mixtures, artificial neural networks, reduced rank regressions, and Boltzmann machines. In general, if a learning machine has hidden variables or hierarchical structures, then it is not regular, but singular. A hidden Markov model is a typical singular learning machine.

In singular learning machines, neither the maximum likelihood estimator nor the Bayes *a posteriori* distribution converges to a normal distribution, which means that the conventional statistical learning theory does not hold. Recently

a new learning theory of singular learning machines was constructed under the condition that training samples are independent [2]. If training samples are independently taken from a true information source, then the average generalization error G is asymptotically equal to

$$G = \frac{\lambda(w^*)}{T}, \quad (1)$$

where T is the number of training samples, and the learning coefficient $\lambda(w^*) > 0$ is equal to the largest pole of the zeta function of the Kullback-Leibler information, which is a function of the true parameter w^* . It was shown that the value of the learning coefficient $\lambda(w^*)$ can be obtained by an algebraic geometrical method [3], by which learning coefficients of layered neural networks and normal mixtures were clarified [1] [4]. In addition, in hidden Markov models, if we have several independent sequences in training, the upper bound of the learning coefficient is evaluated [5]. However, if we have only one time-dependent sequence, the singular learning theory for independent samples cannot be applied to hidden Markov models, because learning theory has not yet been established for time-dependent sequences.

In the present paper, we investigate the learning curve and the learning coefficient $\lambda(w^*)$ under the condition that the sample sequence used in training has time-dependency.

We propose that the generalization error is defined by the expectation value of the Kullback-Leibler information from the ergodic probability distribution to the Bayes predictive distribution. This definition is a generalized concept of the conventional generalization error. Numerical experiments reveal the following: (1) The learning coefficients for time-dependent sequences depend strongly on the true parameter. (2) If the time-dependency is made to be zero, the learning coefficient converges to a constant. (3) If the time-dependency is made to be stronger, then the learning coefficient became increasingly different from the case in which the time-dependency is zero.

2 Ergodic Learning Curve

2.1 Definition of Hidden Markov Model

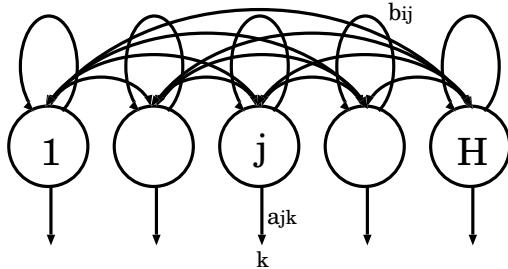
First, we define two sets of time series. Let T be a natural number that shows the length of the time sequence used in training. Let $Y = \{1, 2, \dots, H\}$ and $X = \{1, 2, \dots, K\}$ be a set of hidden states and a set of alphabets, respectively. The sets of all time series on Y and X are defined, respectively, by

$$Y^T = \{y = (y_1, y_2, \dots, y_T) ; y_t \in Y \ (t = 1, 2, \dots, T)\},$$

and

$$X^T = \{x = (x_1, x_2, \dots, x_T) ; x_t \in X \ (t = 1, 2, \dots, T)\}.$$

The transition probability from the hidden state $i \in Y$ to $j \in Y$ is defined by b_{ij} which satisfies $0 \leq b_{ij} \leq 1$ and $\sum_{j \in Y} b_{ij} = 1$. The set of all transition

**Fig. 1.** Hidden Markov Model

probabilities is denoted by $b = \{b_{ij}\}$. In the present paper, we assume that the initial probability distribution at $t = 1$ is given by the H -dimensional vector $\pi = (\pi_1, \pi_2, \dots, \pi_H)$. The output probability of $k \in X$ when the hidden state is $j \in Y$ is defined as a_{jk} which satisfies $0 \leq a_{jk} \leq 1$ and $\sum_{k \in X} a_{jk} = 1$. The set of all output probabilities is denoted by $a = \{a_{jk}\}$. Let $w = (a, b)$ be a set of parameters. Under the condition that w is determined, the simultaneous probability distribution $p(x, y|w)$ is given by

$$p(x, y|w) = \pi_{y_1} a_{y_1 x_1} b_{y_1 y_2} a_{y_2 x_2} b_{y_2 y_3} a_{y_3 x_3} \cdots b_{y_{T-1} y_T} a_{y_T x_T}.$$

The hidden Markov model (HMM) is defined by the marginal distribution,

$$p(x|w) = \sum_{y \in Y^T} p(x, y|w),$$

which can be rewritten as

$$p(x|w) = \pi \cdot AB_2B_3 \cdots B_T v, \quad (2)$$

where A is the diagonal matrix defined by

$$(A)_{ii} = a_{ix_1}.$$

The matrix B_t is given by

$$(B_t)_{ij} = b_{ij} a_{jx_t}, \quad (3)$$

and

$${}^t v = (1, 1, 1, \dots, 1),$$

where ${}^t v$ is a transposed vector of v .

3 Bayes Learning

In the present paper, we investigate the ergodic generalization error under the condition that the hidden sequence of the Markov random process y_t converges

to the ergodic probability distribution $p^\infty(j|w)$ on Y . For example, if $b_{ij} > 0$ for all (i, j) , then such a condition is satisfied. The output distribution converges to

$$p^\infty(k|w) = \sum_{j \in Y} p^\infty(j|w) a_{jk} \quad (k \in X).$$

Let w^* be the parameter that represents the true unknown information source. In other words, we assume that the time sequence x used in training is randomly taken from the probability distribution $p(x|w^*)$. If an *a priori* distribution is denoted by $\varphi(w)$, then the *a posteriori* distribution is equal to

$$p(w|x) = \frac{1}{Z} p(x|w) \varphi(w),$$

where Z is a normalizing constant. The predictive distribution of Bayes learning is defined by

$$p(x_{T+1}|x) = \int p(x_{T+1}|w)p(w|x)dw.$$

The predictive distribution converges to

$$p^\infty(k|x) = \int p^\infty(k|w)p(w|x)dw.$$

We propose that the following ergodic generalization error is appropriate for time-dependent learning theory.

Definition. The ergodic generalization error is defined by

$$G_T = E_x \left[\sum_{k \in X} p^\infty(k|w^*) \log \frac{p^\infty(k|w^*)}{p^\infty(k|x)} \right], \quad (4)$$

where $E_x[\cdot]$ shows the expectation value over all training sequences x . The ergodic learning coefficient $\lambda(w^*)$ is defined by

$$\lambda(w^*) = \lim_{T \rightarrow \infty} T G_T. \quad (5)$$

In other words,

$$G_T \cong \frac{\lambda(w^*)}{T} + o\left(\frac{1}{T}\right).$$

In general, it is difficult to study the ergodic generalization error theoretically for general time-dependent training samples. Hence, in the present paper we experimentally analyze its behavior.

4 Experiments

4.1 Settings

In the experiments, the sets of hidden states and alphabets were $Y, X = \{1, 2\}$. The parameters of the true distribution were set symmetrically as Fig. 2(a).

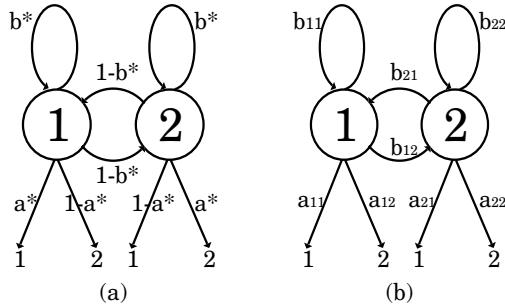


Fig. 2. (a) True distribution. (b) Learning machine.

The transition probability of the true distribution from the hidden state $i \in Y$ to itself was denoted by b^* and the other was denoted by $1 - b^*$. When the hidden state was $j \in Y$, the output probability of the alphabet $j \in X$ was denoted by a^* and the other $1 - a^*$. The parameter of the learning machine w was $(\{b_{ij}\}, \{a_{jk}\})$, $k \in X$. The initial probability distributions of both the true parameter π^* and the learning machine π were set as $\pi^* = \pi = (1.0, 0.0)$.

In order to obtain the numerical value of the average over the *a posteriori* distribution, we used the Metropolis algorithm. Two thousand samples were generated from the *a posteriori* distribution. The average ergodic generalization error G_T , shown in Eq.(4), was experimentally calculated using the average over 100 trials. To simplify the experiments, the length T of time sequences used in training was set as $5n + 1$, $n = 1, 2, \dots, 59$.

By definition, the value G_T approaches the function $f(T) = \lambda(w^*)/T$, where the length T was large. Therefore, in experiments, the ergodic learning coefficient $\lambda(w^*)$ in Eq.(5) was determined by the value that minimized the following sum-of-squares error $E(\lambda)$, as follows:

$$\lambda = \underset{\lambda}{\operatorname{argmin}} E(\lambda) , \quad (6)$$

$$E(\lambda) = \frac{1}{2} \sum_{n=20}^{59} (f(n, \lambda) - G_{5n+1})^2 ,$$

where

$$f(n, \lambda) = \frac{\lambda}{5n + 1} .$$

4.2 Results

In order to investigate the behavior of ergodic generalization error where the length T is large, we first conducted an experiment using one parameter at $a^* = b^* = 0.1$, as shown in Fig. 3. The horizontal axis indicates the length of sequences T , and the vertical axis indicates the ergodic generalization error G_T . Both axes are on a log scale. The solid line denotes the ergodic learning curve,

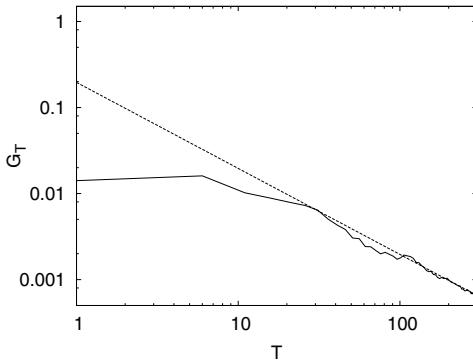


Fig. 3. The solid curve denotes the results of the ergodic generalization error G_T with respect to the length of time sequences T on a log scale. The dotted line is the value λ/T against T , for the case in which the true parameter is $b^* = a^* = 0.1$.

and the dotted straight line denotes the function $f(T) = \lambda/T$. The ergodic learning coefficient $\lambda(w^*)$ was approximately equal to 0.196. When the length T is large, we see that the learning curve approaches the function $f(T)$. For the case in which the true parameter took another value, the learning curve also approached the function $f(T)$. However, the ergodic learning coefficient λ was different in each case.

In order to investigate the relation between the learning coefficient and the true parameter, we conducted experiments in which the parameters a^* and b^* were in the set $\{0.1, 0.2, \dots, 0.9\}$. The obtained results are shown in Fig. 4. The horizontal axes denote the true parameters a^*, b^* , and the vertical axis denotes the learning coefficient λ . We can see from Fig. 4 that the learning coefficient takes different values depending on the true parameter. Then, we investigated

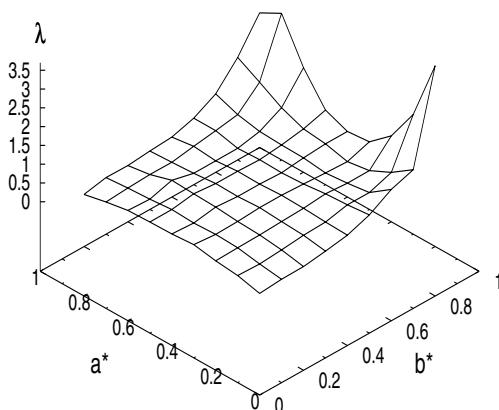


Fig. 4. Three-dimensional-plot of the true parameters a^*, b^* and λ

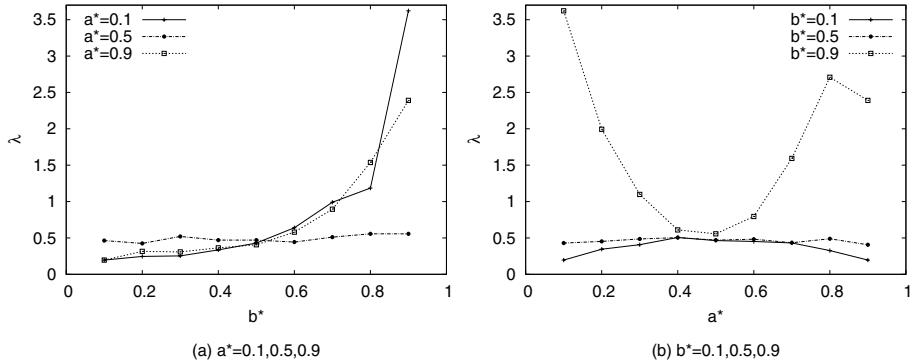


Fig. 5. (a) $\lambda(w^*)$ plotted with respect to the true parameter b^* when $a^* = 0.1$ (solid line), 0.5 (dashed line), and 0.9 (dotted line). (b) λ plotted with respect to the parameter a^* when $b^* = 0.1$ (solid line), 0.5 (dashed line), and 0.9 (dotted line).

the relation between one parameter and λ . The dashed lines in Fig. 5(a) and Fig. 5(b) show that the values λ are approximately equal to 0.5. The other cases in Fig. 5(a) show that the value b^* is made large, the coefficient λ also becomes large. For the case in which the value b^* is constant in Fig. 5(b), if the value $b^* = 0.1$ and the value a^* diverge greatly around 0.5, the ergodic learning coefficient λ becomes lower. However, if $b^* = 0.9$, then as a^* dig from 0.5, the learning coefficient λ becomes larger, except for the case in which the parameter a^* is 0.9.

5 Discussion

When the true parameters were in the set $\{a^* \cup b^* = 0.5\}$, the ergodic learning coefficients were approximately equal to 0.5. In such cases, the second eigenvalues of matrices B_t in Eq.(3) are zero, and there are no correlations in the sequences. These cases are equivalent to the learning of sequences from a Bernoulli distribution, where the output probabilities are 0.5.

In other cases, the value of the coefficient behaved differently. If the value a^* was constant and only the value b^* was made large, the learning coefficient was also large. If $b^* > 0.5$ and a^* was made to be far from 0.5, the ergodic learning coefficient λ became larger. In this case, the transition probability from one state to another is very low. Therefore, the sequences have little information about transition probability. This is one reason that the learning is difficult. In contrast, for the cases in which $b^* < 0.5$ and a^* is a constant, the farther a^* is from 0.5, the smaller the learning coefficient λ . These results will be studied precisely by detailed experiments.

Due to the setting whereby the output probabilities in the true distribution have symmetry, we conjecture that the value λ also has symmetry with respect to the plane $a^* = 0.5$. In this experiment, however, it is difficult to determine the

symmetry accurately. One reason for this is that the *a posteriori* distribution may have separate peaks and the Metropolis algorithm did not work well. The length of training sequences may not be sufficient. Another reason is that the number of averages was insufficient to approximate the calculation of the expected value.

In the future, we hope to clarify the case in which the true parameter is asymmetrical and the number of states is greater than 2.

6 Conclusion

First, we defined the ergodic generalization error and the learning coefficient in order to investigate machine learning for the case in which the data was time-dependent.

Second, we carried out the experiments on hidden Markov models, which are widely used in the analysis of time sequences.

Finally, we obtained the following results:

1. The learning coefficient for one time sequence depended strongly on the true parameter.
2. If the time-dependency was made to converge to zero, the learning coefficient converged to a constant.
3. If the time-dependency was made to be stronger, the learning coefficient became increasingly different from the case in which the time-dependency was zero.
4. The proposed ergodic generalization error was useful in the investigation of time-dependent learning theory.

Acknowledgment

The present study was supported by the Ministry of Education, Science, Sports, and Culture of Japan through a Grand-in-Aid for Scientific Research, No. 18079007.

References

1. Aoyagi, M., Watanabe, S.: Stochastic complexities of reduced rank regression in Bayesian estimation. *Neural Networks* 18(7), 924–933 (2005)
2. Watanabe, S.: Algebraic analysis for nonidentifiable learning machines. *Neural Computation* 13(4), 899–933 (2001)
3. Watanabe, S.: Algebraic geometrical methods for hierarchical learning machines. *Neural Networks* 14(8), 1049–1060 (2001)
4. Yamazaki, K., Watanabe, S.: Singularities in mixture models and upper bounds of stochastic complexity. *Int. J. Neural Networks* 16, 1029–1038 (2003)
5. Yamazaki, K., Watanabe, S.: Algebraic geometry and stochastic complexity of hidden Markov models. *Neurocomputing* 69(1), 62–84 (2005)

Design of Exchange Monte Carlo Method for Bayesian Learning in Normal Mixture Models

Kenji Nagata¹ and Sumio Watanabe²

¹ P. & I Lab. Tokyo Institute of Technology, MailBox: R2-5, 4259, Nagatsuta-chou, Midori-ku, Yokohama, 226-8503, Japan
`kenji.nagata@cs.pi.titech.ac.jp`

² P. & I Lab. Tokyo Institute of Technology, Japan
`swatanab@pi.titech.ac.jp`

Abstract. The exchange Monte Carlo (EMC) method was proposed as an improved algorithm of Markov chain Monte Carlo method, and its effectiveness has been shown in spin-glass simulation, Bayesian learning and many other applications. In this paper, we propose a new algorithm of EMC method with Gibbs sampler by using the hidden variable representing the component from which the datum is generated, and show its effectiveness by the simulation of Bayesian learning of normal mixture models.

1 Introduction

A normal mixture model is a learning machine which estimates the target probability density by the sum of several normal distributions. In recent studies of machine learning theory, Bayesian learning provides good generalization performance for normal mixture models [4]. However, Bayesian learning needs to estimate an expectation over the Bayesian posterior distribution, which can seldom be performed analytically. Hence, the Markov chain Monte Carlo (MCMC) method is well used as the approximation method for Bayesian learning.

The MCMC methods are computational algorithms for sampling from the probability distribution based on constructing a Markov chain that has the target distribution as its stationary distribution [3]. Recently, the exchange Monte Carlo (EMC) method [1] was proposed as an improved algorithm of the conventional MCMC methods such as Metropolis algorithm and the Gibbs sampler, and its effectiveness has been shown in Bayesian learning for the learning machines with hierarchical structures [2]. This method involves generating the Markov chain from a joint distribution, which consists of some distributions with different temperatures. Its algorithm is based on two steps of MCMC simulations. One step is the conventional update of the MCMC simulation for each distribution. The other is the probabilistic exchange process between two distributions.

In this paper, we propose a new algorithm of EMC method for Bayesian learning of normal mixture models. In previous works, the Gibbs sampler has generally been used for Bayesian learning of normal mixture models by using the hidden variable representing the component from which the datum is generated.

The proposed EMC algorithm is constructed by using the hidden variable in order to apply the Gibbs sampler to sampling from the replicated distribution with different temperatures. Moreover, the effectiveness of proposed algorithm is shown by the simulation for Bayesian learning of normal mixture models.

2 Normal Mixture Model

A normal mixture model is a learning machine which estimates the target probability density by sum of normal distributions. A normal mixture model $p(x|w)$ of an M -dimensional input $x \in R^M$ with a parameter vector w is defined by

$$p(x|w) = \sum_{k=1}^K a_k g(x - \mu_k), \quad g(x) \propto \exp\left(-\frac{\|x\|^2}{2\sigma_k^2}\right),$$

where K is the number of components and $\{a_k | a_k \geq 0, \sum_{k=1}^K a_k = 1\}$ is the set of coefficients. In this paper, we consider the case that the variance σ_k^2 for each component is a constant. Hence, the parameter w of this learning machine is $w = \{a_k, \mu_k\}_{k=1}^K$.

The normal mixture model can be rewritten as follows by using a hidden variable $y = (y_{(1)}, \dots, y_{(K)}) \in \{(1, 0, \dots, 0), (0, 1, \dots, 0), \dots, (0, 0, \dots, 1)\}$,

$$p(x, y|w) = \prod_{k=1}^K [a_k g(x - \mu_k)]^{y_{(k)}}.$$

The hidden variable y is not observed and is representing the component from which the datum x is generated. If the datum x is from the k -th component, then $y_{(k)} = 1$, and if otherwise, $y_{(k)} = 0$. Then, it follows that

$$\sum_y p(x, y|w) = p(x|w), \quad (1)$$

where \sum_y means the sum over all the set of hidden variables y .

The conjugate prior distribution $\varphi(w)$ for the normal mixture model is given by multiple of the following two distributions,

$$\varphi(\mathbf{a}) = \frac{\Gamma(K\phi_0)}{\Gamma(\phi_0)^K} \prod_{k=1}^K a_k^{\phi_0-1}, \quad \varphi(\boldsymbol{\mu}) = \prod_{k=1}^K \sqrt{\frac{\xi_0}{2\pi}}^M \exp\left(-\frac{\xi_0\|\mu_k - \nu_0\|^2}{2}\right),$$

where \mathbf{a} and $\boldsymbol{\mu}$ are sets of the parameter a_k and μ_k . Then, the hyperparameters are $\phi_0 > 0$, $\xi_0 > 0$, and $\nu_0 \in R^M$. By considering the conjugate prior distribution, we can construct the algorithm of the Gibbs sampler [3]. Hence, this paper treats the conjugate prior distribution.

The normal mixture models have hierarchical structure, and hence, Bayesian learning provides better generalization performance than the maximum likelihood estimation or the MAP estimation [4].

3 Framework of Bayesian Learning

Let $X^n = (x_1, x_2, \dots, x_n)$ be n training data independently and identically taken from the true distribution $q(x)$, where $x \in R^N$. In Bayesian learning of a learning machine $p(x|w)$ whose parameter is $w = \{w_{(1)}, \dots, w_{(d)}\} \in R^d$, the prior distribution $\varphi(w)$ of the parameter w needs to be set. Then, the posterior distribution $p(w|X^n)$ is defined by the given dataset X^n and the prior distribution $\varphi(w)$ as follows,

$$p(w|X^n) = \frac{1}{Z(X^n)} \varphi(w) \prod_{i=1}^n p(x_i|w),$$

where $Z(X^n)$ is the normalization constant. In Bayesian learning, the predictive distribution $p(x|X^n)$ is given by averaging the learning machine over the posterior distribution,

$$p(x|X^n) = \int p(x|w)p(w|X^n)dw,$$

which estimates the true density function of x given dataset X^n .

Bayesian learning requires an expectation over the Bayesian posterior distribution, which cannot be performed analytically. The well-used application methods of Bayesian learning include the Markov chain Monte Carlo method. In particular, the Gibbs sampler, one of MCMC algorithms, has generally been used for Bayesian learning of normal mixture models.

4 Gibbs Sampler for Normal Mixture Models

The Gibbs sampler is one algorithm of MCMC methods based on the sampling from the full conditional distribution of the target distribution. In Bayesian learning of normal mixture models, it is difficult to sample from the distribution $p(w|X^n)$ by the Gibbs sampler directly. Instead, we can construct the algorithm of the Gibbs sampler by considering the sampling from the following joint distribution of the set of the hidden variables $Y^n = \{y_1, \dots, y_n\}$ and the parameter w ,

$$p(Y^n, w|X^n) \propto \varphi(w) \prod_{i=1}^n p(x_i, y_i|w).$$

Its algorithm is outlined as follows [3]:

Algorithm of Gibbs Sampler

1. Set the initial values, $\mathbf{a}^{(0)}$ and $\boldsymbol{\mu}^{(0)}$.
2. Sample the set of hidden variables $Y^{n,(t)}$ based on the conditional probability distribution $p(Y^n|w^{(t-1)}, X^n)$. More specifically, randomly sample each hidden variable y_i for the training datum x_i from the following multinomial distribution,

$$\text{Mult}(y_i; 1, \gamma_i) = \prod_{k=1}^K (\gamma_{i,(k)})^{y_{i,(k)}},$$

where $\gamma_i = \{\gamma_{i,(1)}, \dots, \gamma_{i,(K)}\}$ is the occurrence probability as follows,

$$\gamma_{i,(k)} = \frac{a_k g(x_i - \mu_k)}{\sum_{j=1}^K a_j g(x_i - \mu_j)}.$$

3. Sample the parameters $\mathbf{a}^{(t)}$ and $\boldsymbol{\mu}^{(t)}$ based on the following conditional probability distribution $p(\mathbf{a}|Y^{n,(t)}, X^n)$ and $p(\boldsymbol{\mu}|Y^{n,(t)}, X^n)$,

$$p(\mathbf{a}|Y^n, X^n) \propto \prod_{k=1}^K a_k^{\phi_k - 1}, \quad p(\boldsymbol{\mu}|Y^n, X^n) \propto \prod_{k=1}^K \exp\left(-\frac{\|\mu_k - \bar{\mu}_k\|^2}{2\bar{\sigma}_k^2}\right), \quad (2)$$

where

$$\phi_k = n_k + \phi_0, \quad \bar{\sigma}_k^2 = \frac{1}{n_k + \xi_0}, \quad \bar{\mu}_k = \frac{n_k \nu_k + \xi_0 \nu_0}{n_k + \xi_0}, \quad (3)$$

$$n_k = \sum_{i=1}^n y_{i,(k)}, \quad \nu_k = \frac{1}{n_k} \sum_{i=1}^n x_i y_{i,(k)}.$$

From Eqs.(2), The parameters $\mathbf{a}^{(t)}$ and $\boldsymbol{\mu}^{(t)}$ are respectively sampled from the Dirichlet distribution and from the normal distribution.

4. Iterate the update for Step 2 and Step 3 alternately T times.

The sequence $\{w^{(t)}, Y^{n,(t)}\}$ obtained by the above algorithm is subject to the probability distribution $p(Y^n, w|X^n)$. Hence, the sequence $\{w^{(t)}\}$ can be regarded as the samples from the distribution $p(w|X^n)$ because it holds that

$$\sum_{Y^n} p(Y^n, w|X^n) = \sum_{y_1} \cdots \sum_{y_n} p(Y^n, w|X^n) = p(w|X^n)$$

5 Exchange Monte Carlo Method

The EMC method was proposed as an improved algorithm of the MCMC method. In Bayesian learning, the conventional MCMC method is to sample from the target distribution $p(w|X^n)$ directly. On the contrary, the EMC method is to simulate the sampling from the joint distribution of the distribution $p(w|\beta, X^n)$ with different value of inverse temperatures $\{\beta_l : l = 1, 2, \dots, L\}$,

$$p(w_1, \dots, w_L) = \prod_{l=1}^L p(w_l|\beta_l, X^n), \quad p(w|\beta, X^n) \propto \varphi(w) \prod_{i=1}^n \{p(x_i|w)\}^\beta.$$

Its algorithm is to repeat the following two steps alternately:

1. Update each samples $\{w_l\}$ for a few steps by the conventional MCMC method.

2. Exchange the position between two chains, that is, $\{w_l, w_{l+1}\} \rightarrow \{w_{l+1}, w_l\}$, with the following probability u ,

$$u = \min(1, r), \quad r = \frac{p(w_l | \beta_{l+1}, X^n) p(w_{l+1} | \beta_l, X^n)}{p(w_l | \beta_l, X^n) p(w_{l+1} | \beta_{l+1}, X^n)}.$$

The sequence $\{w_1^{(t)}, \dots, w_L^{(t)}\}$ obtained by the EMC method is subject to the joint distribution $p(w_1, \dots, w_L)$, and each sequence $\{w_l^{(t)}\}$ can be regarded as the samples from the replicated distribution $p(w | \beta_l, X^n)$. Consequently, we can generate the sample sequence $\{w^{(t)}\}$ of parameters from the target distribution $p(w | X^n)$ by setting one of the inverse temperatures as $\beta = 1.0$.

The advantage of the EMC method is that it results in accelerated convergence of the Markov chain compared with the conventional MCMC method. The conventional MCMC method of generating a sample sequence from the target distribution is very computationally expensive because of the slow convergence of the Markov chain. The EMC method ensures the efficient sampling by preparing a simple distribution such as a normal distribution, which makes convergence of the sample sequence straightforward.

6 Proposal – EMC Method for Normal Mixture Models

In this paper, we propose a new algorithm of EMC method for Bayesian learning of normal mixture models. This algorithm is constructed by using the hidden variables Y^n in order to apply the Gibbs sampler to sampling from the replicated distributions. Firstly, we consider the sampling from the following joint distribution $p(Y^n, w | \beta, X^n)$ of the set of hidden variables Y^n and the parameter w with the inverse temperature β ,

$$p(Y^n, w | \beta, X^n) \propto \varphi(w) \left\{ \prod_{i=1}^n p(x_i, y_i | w) \right\}^\beta.$$

We can generate the sample sequence $\{Y^n, w\}$ from this distribution by the Gibbs sampler similar to the algorithm shown in Section 4. However, the sequence $\{w\}$ cannot be regarded as the samples from the replicated distribution $p(w | \beta, X^n)$ because the marginal distribution of $p(Y^n, w | \beta, X^n)$ over the hidden variables Y^n does not correspond to the distribution $p(w | \beta, X^n)$ as follows,

$$\begin{aligned} \sum_{Y^n} p(Y^n, w | \beta, X^n) &= \varphi(w) \prod_{i=1}^n \sum_{k=1}^K \{a_k g(x_i - \mu_k)\}^\beta \\ &\neq \varphi(w) \prod_{i=1}^n \left\{ \sum_{k=1}^K a_k g(x_i - \mu_k) \right\}^\beta = p(w | \beta, X^n) \end{aligned}$$

Hence, we cannot generate the samples from the replicated distribution $p(w | \beta, X^n)$ by the Gibbs sampler. In other words, we can construct a new algorithm

of EMC method with Gibbs sampler by redefining the replicated distribution $p'(w|\beta, X^n)$ as follows,

$$p'(w|\beta, X^n) \propto \varphi(w) \prod_{i=1}^n p(x_i|w, \beta), \quad p(x|w, \beta) \propto \sum_{k=1}^K \{a_k g(x - \mu_k)\}^\beta.$$

Consequently, the proposed algorithm of EMC method is to iterate the following two steps in alternate shifts:

1. Update each Markov chain $\{w_l\}$ for a few steps by the Gibbs sampler. Its algorithm is similar to the algorithm shown in Section 4 except two points: to redefine the occurrence probability $\gamma'_{i,(k)}$ for Step 2 as follows,

$$\gamma'_{i,(k)} = \frac{\{a_k g(x - \mu_k)\}^\beta}{\sum_{k=1}^K \{a_k g(x - \mu_k)\}^\beta},$$

and to change the parameter n_k to $n'_k = \beta n_k$ in Eqs. (3).

2. Exchange the position between two chains, $\{w_l, w_{l+1}\} \rightarrow \{w_{l+1}, w_l\}$, with the following probability u' ,

$$u' = \min(1, r'), \quad r' = \frac{p'(w_l|\beta_{l+1}, X^n)p'(w_{l+1}|\beta_l, X^n)}{p'(w_l|\beta_l, X^n)p'(w_{l+1}|\beta_{l+1}, X^n)}.$$

7 Experimental Results

In order to clarify the sampling accuracy of the proposed method, we simulate the experiments for Bayesian learning in normal mixture models. These experiments compared the following three algorithms of the MCMC method.

1. [Gibbs]. In this algorithm, the Gibbs sampler was carried out for the Bayesian posterior distribution.
2. [EMC1]. In this algorithm, the exchange Monte Carlo method was carried out for the Bayesian posterior distribution. For the temperature setting, the total number L of temperatures was set 50, and each value was configured by

$$\beta_l = \begin{cases} 0 & (\text{if } l = 1) \\ (1.25)^{-L+l} & (\text{otherwise}). \end{cases}$$

In this case, each temperature is contained the range $[0, 1]$.

3. [EMC2]. In this algorithm, the exchange Monte Carlo method was carried out for the Bayesian posterior distribution. For the temperature setting, the total number L of temperatures was set 20, and each value was configured by $\beta_l = (1.25)^{-L+l+10}$. In this case, each temperature is contained the range $[0.1, 10.0]$.

For these algorithms, the Monte Carlo Steps (MCS), the criterions for iteration of MCMC method, were defined as follows.

1. [Gibbs]. In this algorithm, one MCS means carrying out 100 steps of the Gibbs sampler and picking up the last sample.

2.[EMC1]. In this algorithm, one MCS means carrying out 2 steps of the Gibbs sampler for 50 probability distribution $p(w_l|\beta_l, X^n)$ and picking up the last sample w_l from the probability distribution $p(w_l|\beta_l = 1.0, X^n)$.

3.[EMC2]. In this algorithm, one MCS means carrying out 5 steps of the Gibbs sampler for 20 probability distribution $p(w_l|\beta_l, X^n)$ and picking up the last sample w_l from the probability distribution $p(w_l|\beta_l = 1.0, X^n)$.

Under these setting, the computational costs for these three algorithms are almost equal.

7.1 Behavior of Markov Chain

In this experiment, we compared the behavior of the Markov chain subject to the Bayesian posterior distribution. Let x be 1-dimensional datum from the following true distribution $q(x)$ with one component, $q(x) = g(x - \mu^*)$. For the input data $X^n = \{x_1, \dots, x_n\}$, we simulated learning of the following normal mixture model $p(x|w)$ with two components, $p(x|w) = ag(x - \mu_1) + (1 - a)g(x - \mu_2)$. Then, the parameter w of the learning machine $p(x|w)$ is $\{a, \mu_1, \mu_2\}$. In this case, the true parameter, on which $p(x|w) = q(x)$, is on the following subsets of the parameters,

$$\{a = 1, \mu_1 = \mu^*\} \cup \{a = 0, \mu_2 = \mu^*\} \cup \{\mu_1 = \mu_2 = \mu^*\}.$$

These subsets can be classified into two classes of parameters. One is the set of parameters on which the learning machine reduces the redundant component, which is $\{a = 1, \mu_1 = \mu^*\} \cup \{a = 0, \mu_2 = \mu^*\}$. The other is the set of parameters on which the learning machine uses all the components including the redundant component, which is $\{\mu_1 = \mu_2 = \mu^*\}$.

The number n of the training data was set 500, and the hyperparameters of the prior distribution were respectively set $\phi_0 = 1.0$, $\nu_0 = 0.0$, and $\xi_0 = 1.0$. For the parameter μ^* of the true distribution, two types were considered; $\mu^* = 0.0$, and $\mu^* = 5.0$. For the iteration of the MCMC methods, we simulated 200 MCS for the burn-in period, and 2000 MCS for sampling from the target distribution.

Fig. 1 shows their experimental results. The upper figures indicate the results in the case of $\mu^* = 0.0$ and the lower ones $\mu^* = 5.0$. The left, middle, and right figures respectively show the results of the algorithm “Gibbs”, “EMC1”, and “EMC2”. The solid lines in these figures show the sets of true parameters. Hence, the samples of the Markov chain are around the whole region of the true parameters if the accurate sampling of the MCMC method is executed. According to these results, the algorithm “Gibbs” can accurately sample parameters w from the target distribution in the case $\mu^* = 0.0$, while this algorithm samples locally around the parameter set $\{a = 1, \mu_1 = \mu^*\}$ in the case $\mu^* = 5.0$. This reason is that the Bayesian posterior distribution $p(w|X^n)$ has some regions with high density which are separated from each other in the case $\mu^* = 5.0$. On the contrary, while the algorithm “EMC1” samples locally around the parameter set $\{\mu_1 = \mu_2 = \mu^*\}$ in each case, the algorithm “EMC2” can sample from the target distribution in each case.

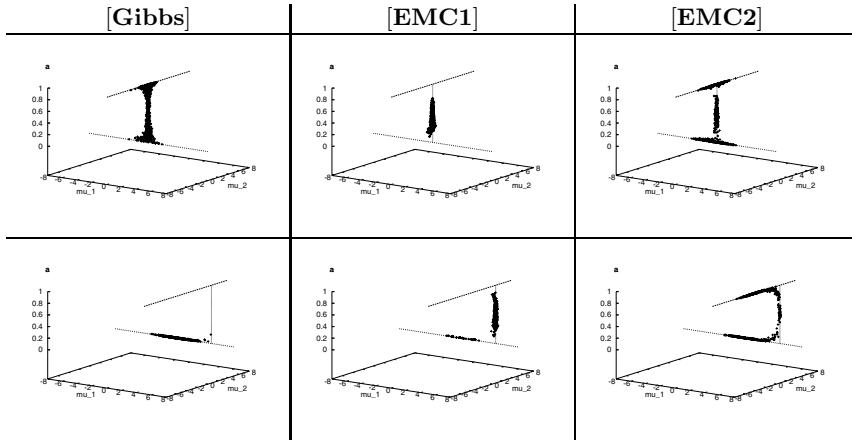


Fig. 1. Behavior of sample sequence from the Bayesian posterior distribution $p(w|X^n)$. The upper figures show the result for $\mu^* = 0.0$ and the lower ones for $\mu^* = 5.0$.

7.2 Generalization Performance

This experiment is to study the generalization performance of Bayesian learning by the three algorithms of the MCMC methods. The dimension N of data x is set 3, and the number of component of the true distribution K_0 and of the learning machine K are respectively set 4 and 10. The true parameter \mathbf{a}^* and $\boldsymbol{\mu}^*$ are respectively set as $\mathbf{a}^* = \{0.33, 0.26, 0.18, 0.23\}^T$, $\mu_1^* = \{-3.39, 1.22, -0.14\}^T$, $\mu_2^* = \{2.79, 3.32, -2.15\}^T$, $\mu_3^* = \{1.40, -0.05, -2.09\}^T$, and $\mu_4^* = \{1.22, 3.05, 1.99\}^T$, where T indicates the transpose of a vector. The number n of the training data was set 1000, and the hyperparameters of the prior distribution were respectively set $\phi_0 = 1.0$, $\nu_0 = \{0.0, 0.0, 0.0\}^T$, and $\xi_0 = 0.1$. For the setting for MCS, we simulated 500 MCS for the burn-in period, and 5000 MCS for sampling from the target distribution. As a function for verifying the generalization performance, the generalization error was approximated by

$$G(X^n) = \frac{1}{n'} \sum_{i=1}^{n'} \log \frac{q(x'_i)}{p(x'_i|X^n)}$$

with $n' = 5000$ test data $\{x'_i\}_{i=1}^{n'}$ from true distribution. Table II shows the average of the generalization error over 100 draws of the training data sets. The results in this table indicate “average \pm standard deviation”. In this case of learning, the upper bound of the generalization error is theoretically clarified [5], which is also shown in Table II. According to these results, the algorithm “Gibbs” and “EMC1” provide the average of generalization error larger than its theoretical upper bound, while the algorithm “EMC2” provide better generalization performance than the other algorithms.

Table 1. The result of generalization error, which indicates “average \pm standard deviation” over 100 sets of training data X^n with $n = 1000$

Algorithm	Generalization Error
Gibbs	0.011139 ± 0.003253
EMC1	0.011268 ± 0.003251
EMC2	0.009881 ± 0.002983
Theoretical upper bound	0.010500

8 Discussion and Conclusion

In this paper, we proposed a new algorithm of EMC method with the Gibbs sampler for Bayesian learning of normal mixture models. Also, we showed its effectiveness by the simulation of Bayesian learning. As a result, we clarified that the proposed algorithm can approximate the Bayesian posterior distribution more accurately than the Gibbs sampler. Moreover, the accurate sampling is executed by the optimal temperature setting for the EMC method.

In this section, we discuss the behavior of the Markov chain for the Gibbs sampler. In the first experimental result, the Gibbs sampler can approximate the Bayesian posterior distribution in the case $\mu^* = 0.0$, while it generates the Markov chain localized around a set of true parameter in the case $\mu^* = 5.0$. This is because the support of the prior distribution $\varphi(\mu_k)$ does not include the true parameter μ^* . In order to avoid this problem, we need to set the prior distribution so that its support includes most of the training data. However, this setting is generally more difficult as the dimension of the data x or the number of component K increases. In this way, it is difficult for the Gibbs sampler to approximate the Bayesian posterior distribution in normal mixture models accurately.

On the other hand, for the EMC method, the algorithm “EMC2” is found to approximate the Bayesian posterior distribution accurately, while the algorithm “EMC1” generates the samples around the set of parameters on which all the components are employed. This is caused by the behavior of the Gibbs sampler for the replicated distribution $p'(w|\beta, X^n)$. Fig. 2 shows the behavior

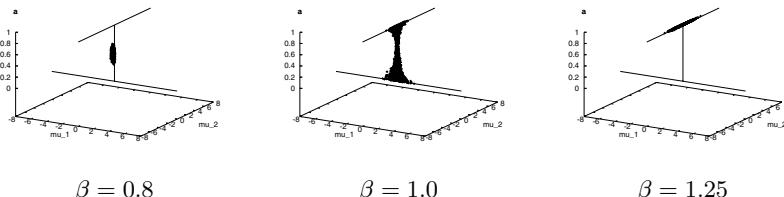


Fig. 2. The behavior of Markov chains for the Gibbs sampler from the replicated distribution $p'(w|\beta, X^n)$ with different temperatures

of the Markov chains for the Gibbs sampler from the replicated distributions $p'(w|\beta, X^n)$ with the different temperatures $\beta = 0.8, 1.0, 1.25$ in the case $\mu^* = 0.0$ of the first experiment. In this experiment, if $\beta < 1.0$, the samples of the Markov chain are around the parameters w on which all the components are employed, and if $\beta > 1.0$, the samples are around the parameters on which the redundant component is reduced without exception. Because of these properties, the algorithm “EMC1”, in which the values of temperatures are within the range [0.0, 1.0], generates the parameter w on which all the components are employed. Also, the algorithm “EMC2”, in which the values of temperatures are within the range [0.1, 10.0], approximates the Bayesian posterior distribution more accurately than the other algorithms.

Finally, we discuss the improvement of the proposed algorithm of EMC method. Our proposed algorithm is based on the Gibbs sampler for Bayesian learning of normal mixture model. In general, the Gibbs sampler is applicable to Bayesian learning of mixture of exponential families as well as normal mixture models. Hence, the proposed algorithm of EMC method can be applied to Bayesian learning of mixture of exponential families such as binomial distributions and Gamma distributions. Moreover, we can consider the improvement for the replicated distribution $p'(w|\beta, X^n)$. In this setting, the inverse temperature β influences the mixing ratio a_k and the variance of each component for the distribution $p(x|w, \beta)$. As an improvement, the replicated distribution is constructed in order to distinguish the above influences of the inverse temperature β . That is to say, the replicated distribution is redefined as follows,

$$p'(w|\beta_1, \beta_2, X^n) \propto \varphi(w) \prod_{i=1}^n \sum_{k=1}^K \{a_k\}^{\beta_1} g(x - \mu_k)^{\beta_2},$$

which can be also sampled by the Gibbs sampler. This setting includes the proposed algorithm in this paper. Hence, this improved algorithm can be better for sampling accuracy and for generalization performance than the proposed algorithm. Also, for another improvement, the hyperparameters ϕ_0, ν_0 and ξ_0 of the prior distribution $\varphi(w)$ can be regarded as the inverse temperature for EMC method. That is, the replicated distribution is considered as follows,

$$p(w|\phi_0, \nu_0, \xi_0, X^n) \propto \varphi(w|\phi_0, \nu_0, \xi_0) \prod_{i=1}^n p(x_i|w)$$

This distribution can be regarded as the Bayesian posterior distribution with the different prior distribution. In this way, considering the improvements of the EMC method is very important for Bayesian learning. Hence, to clarify the sampling accuracy and the generalization performance for these improved algorithm of EMC method should be addressed as our future works.

Acknowledgement. This work was supported by the Ministry of Education, Science, Sports and Culture, a Grant-in-Aid for JSPS Research Fellows 18-5809 and for Scientific Research 18079007.

References

1. Hukushima, K., Nemoto, K.: Exchange Monte Carlo Method and Application to Spin Glass Simulation. *Journal of the Physical Society of Japan* 65(6), 1604–1608 (1996)
2. Nagata, K., Watanabe, S.: Exchange Monte Carlo Sampling from Bayesian Posterior for Singular Learning Machines. *IEEE Transactions on Neural Networks* 19(7), 1253–1266 (2008)
3. Robert, C.P., Casella, G.: *Monte Carlo Statistical Methods*, 2nd edn. Springer, New York (2004)
4. Watanabe, S.: Algebraic analysis for nonidentifiable learning machines. *Neural Computation* 13(4), 899–933 (2001)
5. Yamazaki, K., Watanabe, S.: Singularities in mixture models and upper bounds of stochastic complexity. *Neural Networks* 16(7), 1029–1038 (2003)

Image Filling-In: A Gestalt Approach

Jun Ma

School of Information Security Engineering,
Shanghai Jiao Tong University
mjnicky@gmail.com

Abstract. In this paper, we proposed a bottom-up computational model of visual filling-in to recover not only the texture but also the structure pattern in the unknown area of the images. Different from previous works of image inpainting and texture synthesis, our approach in the first step recovers the structure information of the missing part of an image; and then in the second step, each missing region with homogeneous composition is recovered independently. The structure recovery strategy is based on Gestalt laws of human visual perception, especially the good continuation law that predict the curvilinear continuity in contour completion of human behavior. In the experiment section, we provide the comparative results of our model and other proposed methods. Our model can achieve better performance in recovering images, especially when the scene contains rich structural information.

1 Introduction

Filling-in has been an active research topic in many areas. In computer science, filling-in is motivated by the increasing need of image processing applications such as object removal and image recovery. The problem is mainly addressed by two classes of algorithms: image inpainting and texture synthesis. The former works well on recovering small gaps and simple structures of the image, while the latter can reproduce rich texture based on statistical features. Both approaches have some encouraging results on image textures, however, the recovery of structure remains a challenge in both methods.

Texture synthesis, as way of generating statistically similar contents of an image from a given texture-rich sample, has been widely discussed in [1, 2, 3]. Texture synthesis algorithm can be directly applied to filling-in, in which we use the original image as the sample image to generate texture in the missing area. However, these approaches are based on the assumption that the image has a statistically homogeneous composition, which is often not the case when various components are presented in the image.

Image inpainting techniques [4] are inspired by the work of professional restorers, who are skilled in filling the gaps in deteriorated images with the pigment similar to the gap surroundings. Inpainting techniques fill the missing area by propagating pixels via diffusion. The algorithm works well on filling small gaps and text overlays. And it can also preserve some simple (linear) structures in

the image. However, when applied to the recovery of large missing areas, the diffusion process inevitably introduces blur.

Different approaches have been proposed to take advantage of both texture synthesis and image inpainting methods [5, 6, 7]. Criminisi's method [6] used an example based inpainting approach, adapting texture synthesis method of [2] to inpainting techniques of [4]. Structures are preserved in texture synthesis by guiding the patch synthesis order similar to the order of pixel diffusion in image inpainting. However, the method only preserves limited continuation of image structures based on local correlation of patches. As a result, the method cannot be applied to rich structure images.

Other approaches have been proposed to improve structure recovery, such as interactive methods which allow human to sketch the key boundary structures in the missing area before filling-in [7]. Also, machine learning techniques are introduced to overcome the limitation of local correlation, such as learning from global image statistics [8], and learning from large image databases [9].

In neuroscience, the phenomenon of visual filling-in has been widely discussed [10], and some computation models of visual filling-in have been proposed. In [11] the author implemented a neural network model with both top-down and bottom-up paths in which the missing pattern are reconstructed by feedback signals from the highest stage of the network. However, the pattern reconstruction is based on top-down path so that the performance of their method is affected by the pattern memory issues.

2 Recovering Image Structures – A Gestalt Perspective

The goal of our approach is to find an efficient way to recover the structure information in the missing area of the image by a bottom-up path. Unlike the recovery of textures, which is mainly based on local statistical information of the image, structure recovery is more difficult because of the complexity to extract and represent structures in the image. Our work is inspired by the visual grouping ability of human beings, which makes human beings naturally qualify for the filling-in job, although the underlying mechanisms remain unclear [12]. This ability has been widely discussed as Gestalt psychology and has been concluded into a series of rules, called Gestalt laws.

The concept of Gestalt psychology starts with the assumption of active grouping laws in visual perception, the theory governs how detailed component parts are organized and perceived into overall patterns in visual perception [13, 14]. Gestalt theory implies that we tend to order our perception in a manner that is regular, orderly, symmetric, and simple [14, 15]. Specifically, the perceptual organization is determined by Gestalt laws including Closure, Similarity, Proximity, Symmetry and Continuity. Given a partly corrupted input image, these laws can predict how human eyes will behavior to recover the structure in the missing area. We believe that to implement a computational model of these laws can simply simulate the behavior of human perception in a bottom-up manner.

It has been illustrated in [7] that object boundaries are crucial in structure recovery. In our approach, we achieve the goal of automatically structure recovery

by completing missing texture boundaries in the image. The guiding rule is Gestalt laws, especially the Law of Good Continuity: "Human tends to continue contours whenever the elements of the pattern establish an implied direction". To be simplified, people tend to draw a good continuous contour in the missing area.

3 Structure Preserving Image Filling-In

3.1 Structure Extraction and Representation

In the first step of our approach, we conduct a series of computing stages to extract and encode the structure information in an image. As stated in Gestalt psychology, the fundamental mechanism of human perception is grouping. Grouping mechanism helps human beings to extract and represent structures from visual input by combining small components of texture into a larger pattern or "object". In our approach, we conduct an image segmentation algorithm, which works in a similar way as visual grouping, to extract and encode the structures in images.

Image segmentation is a method to classify the image pixels into different clusters by their local and global features, such as color and position. It groups pixels that are similar in color and close in position into a homogeneous class, these cluster classes represents the objects and textures in the image. We implement a color image segmentation algorithm similar to the widely used JSEG [16]. First, we conduct a k-means cluster algorithm to group the pixels into several clusters by their color similarity. Each pixel is classified into the cluster most close to it by color distance, and the color of a cluster is defined by the average color of all the cluster's pixels. Second, we segment the clusters by considering spatial position information. Here we use the method discussed in [16]: the J-Image to detect real spatial texture boundaries based on the result of color cluster. The result of image segmentation is the class-map. Each pixel in the class-map is labeled by a class type, which denotes its texture. We can easily get texture contours based on class-map, which denotes the texture and object patterns and they are the input of later contour completion stage. The procedure of image segmentation is illustrated in Fig 1.

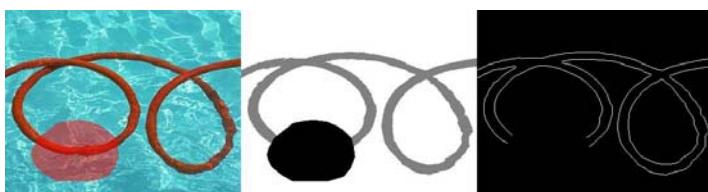


Fig. 1. Structure extraction from partially corrupted images. The transparent red area in the left image denotes the missing area, the white and gray color in the class-map (middle) represent two texture types in the image, and the incomplete texture boundaries (right) denote object pattern.

To better encode the structure of the image, we also construct piecewise linear approximation of the texture contour obtained by the class-map. Piecewise linear representation of curve contour has been widely used in many applications, especially some contour completion systems [17]. In the construction of piecewise linear representation, we split each continuous contour into several lines by a threshold angle θ , which is illustrated in Fig 2. By using piecewise linear representation, contour information in the image can be transformed into line structures. These piecewise contours are more informative than discrete pixels, thus help us to reconstruct missing structures in later contour completion stage.



Fig. 2. The angle θ determines whether to split a contour into several lines

3.2 Curvilinear Continuity in Contour Completion

As concluded in Gestalt laws, human perception tends to draw continuous contours in the missing area. To implement such continuation in our contour completion algorithm, we propose two computation stages: statistic learning stage and recursion completion stage.

In the statistic learning stage, we compute some statistics of our piecewise lines, mainly two types of statistics: average piecewise line length and average angle between two connecting piecewise lines. Both of them reflect the continuation level of the original image's contours. For example, smaller average line length means the contour of the image is more continuous.

In the second stage, we complete the missing contour by drawing connected short lines in the missing area using the statistic parameters obtained before. According to the Law of Closure in Gestalt psychology, each texture region should be surrounded by a closed contour, so we assume that every missing contour connects two lines outside the missing area, we think that complicated missing patterns such as T-structure are handled by top-down path of human visual system and needs a system with rich pattern database, which is not the topic of this paper here. There are two constraints guiding the contour completion. (1) Each line in the missing area is a line of specific length: Average Line Length. (2) The angle between two connecting lines is smaller than Average Connecting Angle. Then we use a recursion function to examine all the possible line groups connecting two outside lines, and extract the most continuous one as the result of contour completion. The most continuous line group is defined as follow: suppose we have several line groups in the missing area, $\phi = \{\psi_1, \psi_2, \dots, \psi_{n_1}\}$. Each line group consists of a series of connecting lines that connects two contour lines outside the missing area: $\psi = \{L_1, L_2, \dots, L_{n_2}\}$. Suppose k_i is the slope of line

L_i in a group ψ , and A_ψ is defined as the maximum angle difference between two connecting lines in a specific group:

$$A_\psi = \max_{0 < i, i+1 < n_2} |\arctan(k_i) - \arctan(k_{i+1})| \quad (1)$$

The most continuous group of lines is the group with the smallest A_ψ :

$$\psi_p = \arg \min_{\psi_q \in \phi} A_{\psi_q} \quad (2)$$

This procedure is illustrated in Fig 3.



Fig. 3. Corrupted contours(left) and all possible line groups connecting two outside contours(middle), the most continuous line groups among all line groups(red in right)

3.3 Regional Textural Filling-In

After the completion of missing contours, we conduct a class propagation method to grow the class label (texture type) into the missing area. The propagation is constrained by our filled contours: class labels cannot propagate over contours, since contours are texture boundaries. The pixels in the missing area will be synthesized with the homogeneous texture type based on the class label of it in later stages. The class propagation procedure is illustrated in Fig 4.

Once all the class labels of the pixels in the missing area have been generated, we use texture synthesis algorithm to reproduce the texture in the missing area. Texture synthesis algorithm draws pixels based on patches. A patch is a small regional area centered at the pixel, in our approach we use the default window

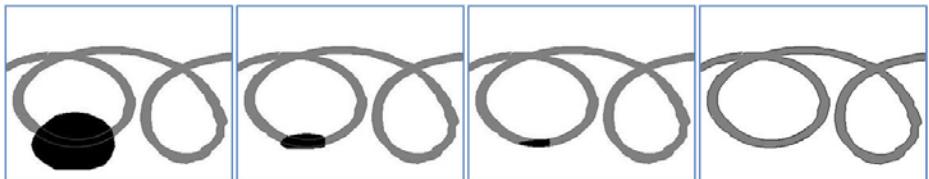


Fig. 4. The class propagation procedure in class-map (from left to right). White color and gray color represents two texture types in the image. They cannot propagate across the fitted boundaries in the missing area.

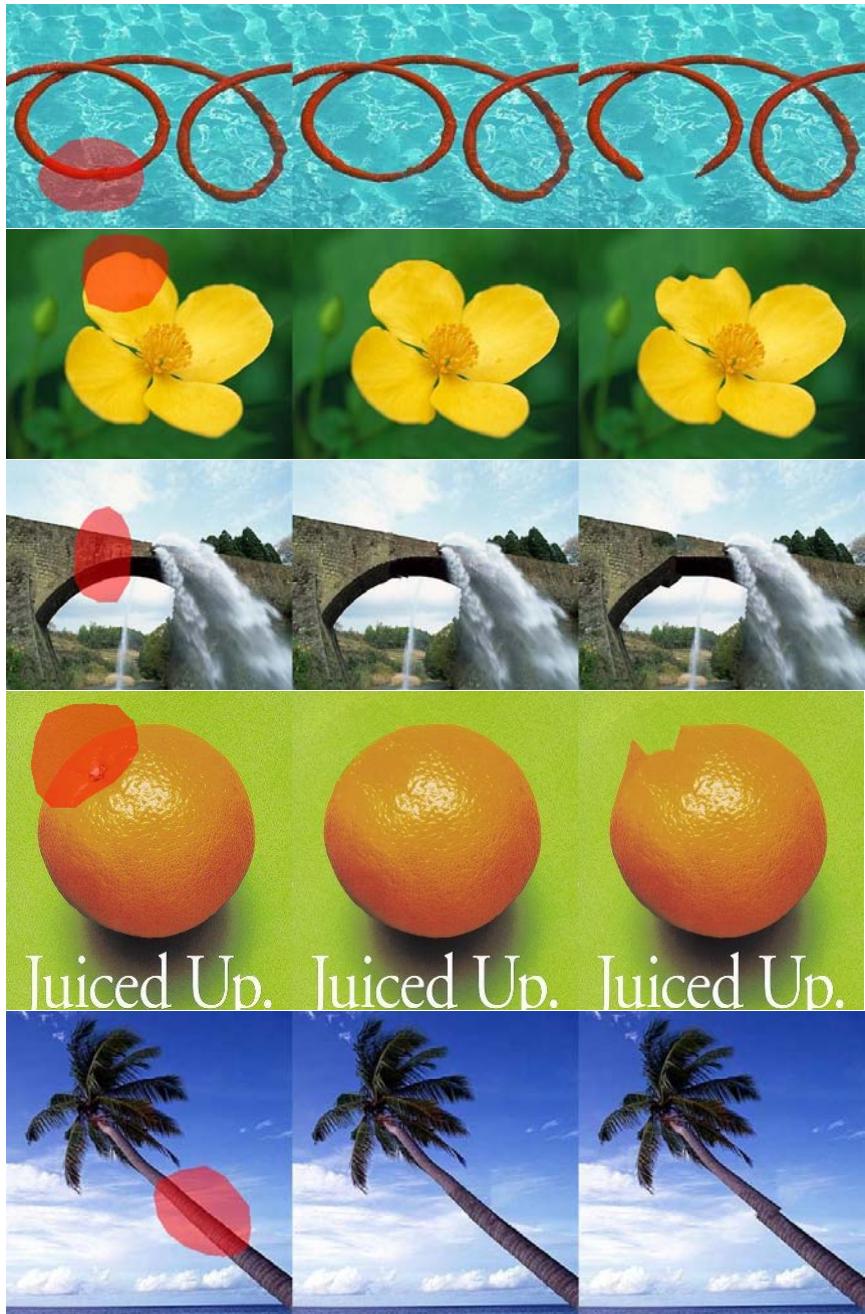


Fig. 5. Comparisons of Different Filling-in Algorithm. Left column: original image. Middle column: result of our approach. Right column: result of normal texture synthesis approach.

size of 9×9 pixels. To each pixel in the missing area, the method find the most similar patch in the image by comparing the pixel's patch with the image patches which has the same texture label, then draw the pixel in the missing area by the value of its most similar patch. The similarity between two patches is defined by the distance between them, which has many different implements. Here we use the most common implementation of sum of square (SSD) to value the similarity between two patches. And the most similar patch P_q is the patch with the smallest distance to the current patch P_c among all patches in the image that have the same texture label: $Q = \{P_1, P_2, \dots, P_{n_3}\}$:

$$P_q = \arg \min_{P_q \in Q} d(P_c, P_q) \quad (3)$$

Similar to the idea in [6], we also define the priority of pixels in the missing area: the priority of a missing pixel is the number of existing pixels in its patch. With this set of priority, the algorithm ensures the pixels on the boundaries of the missing area will be filled at first. Fig 5 illustrates the result of texture synthesis in this stage and its comparison to normal concentric texture synthesis approaches.

4 Discussions

In this paper, we have proposed a framework to recover the missing structures of the image. As we have stated before, the structure of the image is represented by the class map, the result of image segmentation. The criterion for recovering the texture boundaries in the missing area is continuity by the Gestalt laws. The results are satisfactory compared to normal approaches of texture synthesis.

However, there are still some limitations in our algorithm. The result of the algorithm is greatly affected by the quality of the image segmentation step. If the image segmentation step cannot effectively recognize the textures and object in the original image, our model of class map can not accurately represent the texture structure in the image, and later steps such as boundary fitting will be meaningless. Moreover, the bottom-up approach cannot be applied to all types of images. It is applicable in most natural images, where the law of continuity widely occurs [18]. However, in many mankind images, there are many sharp angle structures that do not abide Gestalt rules. To efficiently recover these structures, we should conduct top-down learning procedures like the one in [9] to better recover the visual patterns, and this is what we will emphasize in our future work.

References

1. Ashikhmin, M.: Synthesizing natural textures. In: Proc. ACM Symp. on Interactive 3D Graphics, Research Triangle Park, NC, March 2001, pp. 217–226 (2001)
2. Efros, Leung, T.: Texture synthesis by non-parametric sampling. In: Proc. ICCV, Kerkyra, Greece, September 1999, pp. 1033–1038 (1999)

3. Efros, Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of SIGGRAPH 2001, August 2001, pp. 341–346 (2001)
4. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proc. ACM Conf. Comp. Graphics (SIGGRAPH), New Orleans, LU, July 2000, pp. 417–424 (2000)
5. Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. IEEE Transactions on Image Processing 12(8) (August 2003)
6. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: CVPR 2002, p. 721 (2002)
7. Sun, J., Yuan, L., Jia, J., Shum, H.-Y.: Image Completion with Structure Propagation. In: SIGGRAPH 2005, vol. 24, pp. 861–868 (2005)
8. Levin, A., Zomet, A., Weiss, Y.: Learning How to Inpaint from Global Image Statistics. In: ICCV 2003 (2003)
9. Hays, J., Efros, A.A.: Scene Completion Using Millions of Photographs. ACM Transactions on Graphics (SIGGRAPH 2007) 26(3) (August 2007)
10. Komatsu, H.: The neural mechanisms of perceptual filling-in. Nature Reviews Neuroscience 7 (March 2006)
11. Fukushima, K.: Restoring Partly Occluded Patterns: A Neural Network Model with Backward Paths. In: Kaynak, O., Alpaydin, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 393–400. Springer, Heidelberg (2003)
12. Pessoa, L., Thompson, E., Noe, A.: Finding out about filling in: a guide to perceptual completion for visual science and the philosophy of perception. Behavioral and Brain Sciences 21(6), 723–802 (1998)
13. Wertheimer, M.: Laws of organization in perceptual forms (partial translation). In: Ellis, W. (ed.) A sourcebook of Gestalt Psychology, pp. 71–88. Harcourt Brace and Company (1938)
14. Koffka, K.: Principles of Gestalt psychology. Hartcourt, Brace and World, New York (1935) (1967)
15. Desolneux, A., Moisan, L., Morel, J.-M.: Gestalt Theory and Computer Vision. Theory and Decision Library 38
16. Deng, Y., Manjunath, B.S.: Unsupervised segmentation of color-texture regions in images and video. IEEE Trans. Pattern Anal. Mach. Learn. (PAMI) 23(8), 800–810 (2001)
17. Ren, X., Fowlkes, C.C., Malik, J.: Scale-Invariant Contour Completion using Conditional Random Fields. In: ICCV 2005, Beijing, vol. 2, pp. 1214–1221 (2005)
18. Geisler, W.S., Perry, J.S., Super, B.J., Gallogly, D.P.: Edge co-occurrence in natural images predicts contour grouping performance. Vision Research 41, 711–724 (2001)

Sports Video Segmentation Using a Hierarchical Hidden CRF

Hirotaka Tamada and Akira Hayashi

Graduate School of Information Sciences, Hiroshima City University,
3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima 731-3194, Japan
{tamada,akira}@robotics.im.hiroshima-cu.ac.jp

Abstract. Hidden Markov Models (HMMs) are very popular generative models for sequence data. Recent research has, however, shown that Conditional Random Fields (CRFs), a type of discriminative model, outperform HMMs in many tasks. We have previously proposed Hierarchical Hidden Conditional Random Fields (HHCRFs), a discriminative model corresponding to hierarchical HMMs (HHMMs). Given observations, HHCRFs model the conditional probability of the states at the upper levels. States at the lower levels are hidden and marginalized in the model definition. In addition, we have developed a parameter learning algorithm that requires only the states at the upper levels in the training data. Previously we applied HHCRFs to the segmentation of electroencephalographic (EEG) data for a Brain-Computer Interface, and showed that HHCRFs outperform HHMMs. In this paper, we apply HHCRFs to labeling artificial data and sports video segmentation.

1 Introduction

Hidden Markov Models (HMMs) are very popular generative models for sequence data. Recent research has, however, shown that Conditional Random Fields (CRFs), a type of discriminative model, perform better than HMMs in many tasks [1].

There are several differences between CRFs and HMMs. (1) HMMs are generative models and thus model the joint probability of observation and label sequences, whereas CRFs are discriminative models that model the conditional probability of a label sequence given an observation. (2) HMMs make independence assumptions on observations given states, whereas CRFs do not. (3) For model parameter estimation, HMMs do not need the states, whereas CRFs do.

A hierarchical HMM (HHMM) is a generalization of a HMM with a hierarchical structure [2]. Murphy [3] has shown that a HHMM is a special kind of Dynamic Bayesian Network (DBN), and has derived an efficient inference algorithm [3].

Previously, we proposed Hierarchical Hidden CRFs (HHCRFs), a discriminative model corresponding to the generative model of the HHMM [4]. In hierarchical models, we are mainly interested in the states at the upper levels and hence, HHCRFs directly model the conditional probability of the states at the upper levels given observations. The states at the lower levels are hidden and marginalized in the model definition.

In addition, we have developed a parameter learning algorithm that requires only the states at the upper levels in the training data. Note that all the states are required to train standard CRF models.

In an earlier paper [4], we applied HHCRFs to the segmentation of electroencephalographic (EEG) data for a Brain-Computer Interface, and showed that HHCRFs outperform HHMMs. In this paper, we confirm that HHCRFs outperform HHMMs through two additional experiments, namely labeling artificial data and sports video segmentation.

2 Related Work

HHMMs were originally defined by Fine et al. [2]. Later, Murphy and Paskin [3] devised a DBN representation for HHMMs and developed a linear time inference algorithm based on the DBN representation. We have developed a semi-supervised learning algorithm for HHMMs [5]. Applications of HHMMs include handwritten character recognition, information extraction, video structure discovery, and topic transition detection.

CRFs were originally proposed by Lafferty et al. [1]. Since then, CRFs have successfully been applied to many problems including parsing, named entity recognition, object recognition, and activity recognition.

Sutton et al. [6] proposed Dynamic CRFs (DCRFs), an extension of CRFs corresponding to factorial HMMs [7], while Liao et al. [8] proposed hierarchical CRFs, a further extension corresponding to HHMMs. Because hierarchical CRFs do not have hidden states, all the states must be labeled for model parameter estimation.

For the purpose of phone classification, Gunawardana et al. [9] proposed Hidden CRFs (HCRFs), which have two advantages over the previous CRF extensions. First, they have output feature functions that can express continuous Gaussian outputs. This is in contrast to the binary valued output feature functions in most other CRFs. The second advantage is that HCRFs have hidden states. HCRFs have 2 levels. The states at the bottom level are hidden and thus do not need to be labeled for training. However, the problem with HCRFs is that the states at the top level do not change with time. Therefore, whilst HCRFs can be applied to sequence classification, they cannot be applied to sequence segmentation or sequence labeling.

Influenced by the above-mentioned approaches, our HHCRFs share many characteristics with them. In HHCRFs, however, not only are the states at the lower levels hidden, but the states at the upper levels also change with time. This makes it possible to apply HHCRFs to sequence segmentation and sequence labeling without labeling the states at the lower levels for training.

3 HHMMs

A hierarchical HMM (HHMM) is a generalization of a HMM with a hierarchical structure [2] and has three kinds of states: internal, production, and end states.

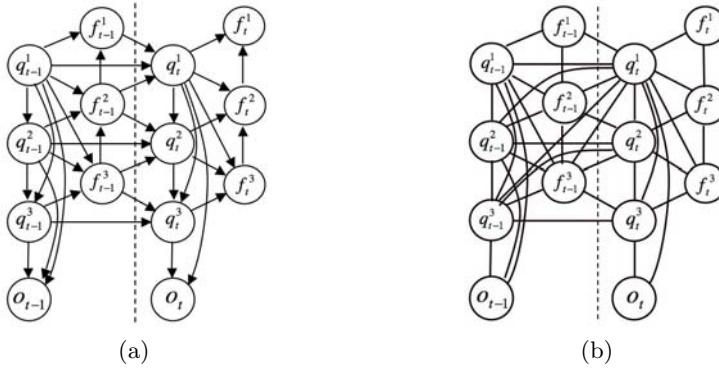


Fig. 1. (a) An HHMM represented as a DBN. (b) An HHCRF represented as an undirected graph. Both figures describe only the part of the model from $t - 1$ to t .

There are also three kinds of transitions: vertical, horizontal, and forced transitions. Murphy [3] has shown that a HHMM is a special kind of DBN, and has derived an efficient inference algorithm [3].

We can represent a HHMM as a DBN as shown in Fig. 1(a). (We assume for simplicity that all production states are at the bottom of the hierarchy.) A state of the HHMM is denoted by q_t^d ($d \in \{1, \dots, D\}$), where d is the hierarchy index: at the top level $d = 1$, while at the bottom level $d = D$.

f_t^d is an indicator variable that is equal to 1 if q_t^d has transited to its end state, and 0 otherwise. Note that if $f_t^d = 1$, $f_t^{d'} = 1$ for all $d' > d$, and hence the number of indicator variables that are equal to 0 denotes the level of the hierarchy we are currently on. The indicator variables play an important role in representing the HHMM as a DBN.

Defined below are the transition and output probability distributions. These complete the definition of the model. Once q_t^d has transited to its end state, $f_t^d = 1$ and this is the signal that the states at the upper levels can be changed. Furthermore, it is a signal that the next value of q_{t+1}^d should be determined by a vertical transition, instead of a horizontal transition. Formally, we denote this as follows:

$$\begin{aligned}
 p(q_t^d = j' | q_{t-1}^d = j, f_{t-1}^{d+1} = b, f_{t-1}^d = f, q_t^{1:d-1} = i) &= \begin{cases} \delta(j, j') & \text{if } b = 0 \\ A_i^d(j, j') & \text{if } b = 1 \wedge f = 0 \\ \pi_i^d(j') & \text{if } b = 1 \wedge f = 1 \end{cases} \\
 p(f_t^d = 1 | q_t^d = j, q_t^{1:d-1} = i, f_t^{d+1} = b) &= \begin{cases} 0 & \text{if } b = 0 \\ Ae^d(i, j) & \text{if } b = 1 \end{cases} \quad (1)
 \end{aligned}$$

$$\begin{aligned}
 E[o_t | q_t^{1:D} = i] &= \mu(i) \\
 \text{Cov}[o_t | q_t^{1:D} = i] &= \sigma^2(i)
 \end{aligned}$$

where the state vector $q_t^{1:d} = \{q_t^1, \dots, q_t^d\}_{d \in \{1, \dots, D\}}$ is represented by an integer i (i.e., i is the index for "mega state"). In Eq. (1), we assume a dummy state $q_t^0 = 0$ (i.e., the root state) for notational convenience. We also assume dummy

indicator variables $f_0^{2:D} = 1$ and $f_t^{D+1} = 1$ for the first slice and for the bottom level, respectively.

$\delta(j, j')$ is Kronecker's delta. $A_i^d(j, j')$ is the horizontal transition probability into the j' th state (except into an end state) from the j th state at level d . $\pi_i^d(j')$ is the vertical transition probability into the j' th state from the i th state at level d . $Ae^d(i, j)$ is the horizontal transition probability into an end state from the j th state at level d .

μ and σ^2 are, respectively, the mean vector and covariance of the Gaussian distribution of the observations emitted at each time. Note that for simplicity, we use scalar observations and a single Gaussian density output. We could, of course, use vector valued observations and a Gaussian mixture density output.

4 HHCRFs

A HHCRF is an undirected graphical model (as shown in Fig. 1(b)) that encodes the conditional probability distribution:

$$p(Q^1|O; \Lambda) = \frac{1}{Z(O; \Lambda)} \sum_{Q^{2:D}} \sum_{F^{1:D}} \exp \left(\sum_{k=1}^K \lambda_k \Phi_k(Q^{1:D}, F^{1:D}, O) \right) \quad (2)$$

where $Q^1 = \{q_1^1, \dots, q_T^1\}$ is the state sequence at the top level¹, $O = \{o_1, \dots, o_T\}$ is the sequence data (observations) and $\Lambda = \{\lambda_1, \dots, \lambda_K\}$ is the model parameter. We represent the state sequence $Q^{1:D} = \{Q^1, \dots, Q^D\}$ and the indicator variable sequence $F^{1:D} = \{F^1, \dots, F^D\}$. $Z(O; \Lambda)$ is the partition function ensuring that $p(Q^1|O; \Lambda)$ is properly normalized.

$$Z(O; \Lambda) = \sum_{Q^1} \sum_{Q^{2:D}} \sum_{F^{1:D}} \exp \left(\sum_{k=1}^K \lambda_k \Phi_k(Q^{1:D}, F^{1:D}, O) \right) \quad (3)$$

where $\Phi_k(Q^{1:D}, F^{1:D}, O)$ is a feature function that is selected arbitrarily.

To compare the performance of HHCRFs and HHMMs that have a Markov structure in the state sequence, we restrict the feature function, $\Phi_k(Q^{1:D}, F^{1:D}, O) = \sum_{t=1}^T \phi_k(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t)$, and make the model structure equivalent to that of the HHMMs. The various feature functions $\phi_k(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t)$ are given below.

$$\begin{aligned} \phi_{j,j',i,d}^{(Hor)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) = & \left(\delta(q_{t-1}^d = j) \cdot \delta(q_t^d = j') \cdot \delta(q_t^{1:d-1} = i) \right. \\ & \left. \cdot \delta(f_{t-1}^{d+1} = 1) \cdot \delta(f_{t-1}^d = 0) \right) \quad \forall_j, \forall_{j'}, \forall_i, \forall_d \end{aligned}$$

¹ For simplicity we assume that only the state sequence at the top level is not hidden. We could, of course, assume that the state sequences at multiple upper levels are not hidden.

$$\begin{aligned}
\phi_{i,j',d}^{(Ver)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \left(\delta(q_t^{d-1} = i) \cdot \delta(q_t^d = j') \right. \\
&\quad \left. \cdot \delta(f_{t-1}^{d+1} = 1) \cdot \delta(f_{t-1}^d = 1) \right) \quad \forall_i, \forall_{j'}, \forall_d \\
\phi_{i,j,d}^{(End)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \left(\delta(q_t^{1:d-1} = i) \cdot \delta(q_t^d = j) \right. \\
&\quad \left. \cdot \delta(f_t^{d+1} = 1) \cdot \delta(f_t^d = 1) \right) \quad \forall_i, \forall_j, \forall_d \\
\phi_i^{(Occ)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \delta(q_t^{1:D} = i) \quad \forall_i \\
\phi_i^{(M1)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \delta(q_t^{1:D} = i) \cdot o_t \quad \forall_i \\
\phi_i^{(M2)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \delta(q_t^{1:D} = i) \cdot o_t^2 \quad \forall_i
\end{aligned}$$

where $\delta(q = q')$ is equal to 1 when $q = q'$ and 0 otherwise. The first three feature functions are transition features. $\phi_{j,j',i,d}^{(Hor)}$ counts the horizontal transition into the j' th state (except into an end state) from the j th state at level d . $\phi_{i,j',i,d}^{(Ver)}$ counts the vertical transition into the j' th state from the i th state at level d . $\phi_{i,j,d}^{(End)}$ counts the horizontal transition into an end state from the j th state at level d . $\phi_i^{(Occ)}, \phi_i^{(M1)}, \phi_i^{(M2)}$ are output features that are necessary to represent the Gaussian density output \mathbf{q} .

It can be shown that setting the parameter Λ (i.e., the weight of the feature functions) as given below produces the conditional probability distribution induced by HHMMs with the transition probability distributions and output probability distributions defined in Eq. (II):

$$\begin{aligned}
\lambda_{j,j',i,d}^{(Hor)} &= \log A_i^d(j, j') & \lambda_i^{(Occ)} &= -\frac{1}{2} \left(\log 2\pi\sigma^2(i) + \frac{\mu^2(i)}{\sigma^2(i)} \right) \\
\lambda_{i,j',d}^{(Ver)} &= \log \pi_i^d(j') & \lambda_i^{(M1)} &= \frac{\mu(i)}{\sigma^2(i)} \\
\lambda_{i,j,d}^{(End)} &= \log A e^d(i, j) & \lambda_i^{(M2)} &= -\frac{1}{2\sigma^2(i)}
\end{aligned} \tag{4}$$

As in HHMMs, parameter estimation for HHCRFs is based on the maximum likelihood principle given a training set $\mathcal{D} = \{O^{(n)}, Q^{1(n)}\}_{n=1}^N$. The difference is that we maximize the conditional probability distribution $p(Q^1|O; \Lambda)$ for HHCRFs, whereas we maximize the joint probability distribution $p(Q^1, O; \Lambda_1)$ for HHMMs. Here, Λ_1 is the parameter for HHMMs.

5 Experiment

5.1 Labeling Artificial Data

We first compare the performance of HHCRFs and HHMMs using artificial data. This experiment is motivated by the comparison of logistic regression (one of

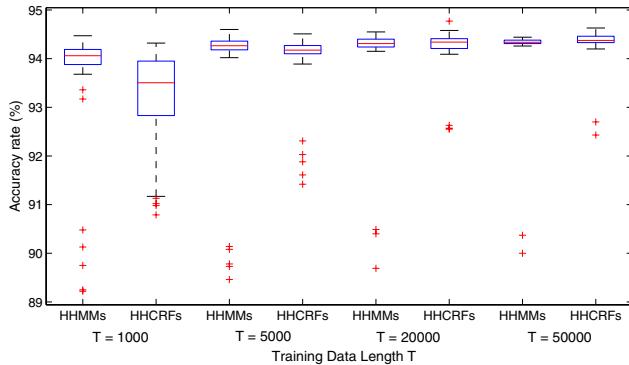


Fig. 2. Accuracy rate for labeling artificial data, given training data of varying lengths. Boxplot of 50 runs.

the discriminative models) and naive Bayes (a generative model) presented by Ng and Jordan [10]. They concluded that given a sufficient number of training examples, logistic regression outperforms the naive Bayes. We in turn compare HHCRFs and HHMMs, given training data of varying lengths.

Artificial data is generated from a 2-level HHMM for training and testing. The length of the test data is 10000. For state sequence estimation, we use the forward-backward algorithm. Figure 2 shows the accuracy rates for labeling artificial data for each execution of the experiment using trained models. When the length of training data is small, HHMMs outperform HHCRFs. As the length of data increases, the performance of both models improves. Once the training data reaches a certain length, HHCRFs outperform HHMMs.

5.2 Sports Video Segmentation

According to the literature, generative models, such as HHMMs [11,12] and layered HMMs [13,14], have been used to model the hierarchical temporal structure of sports videos. In this section, we compare the performance of HHCRFs and HHMMs in segmenting sports videos. In particular, we use volleyball videos recorded from a TV broadcast. The state sequence data in the video clearly has a hierarchical structure.

The HHMMs and HHCRFs each have 3 levels. The top level has 2 states corresponding to "play" and "time-out", while the second level has 4 states defined as follows. When the top level state is "play", each second level state corresponds to either (1) a shot showing a player attempting to serve, (2) a shot of an overall view of the court during a rally, (3) a close-up shot of the players after a point has been scored, or (4) an instant replay, where a shot is a series of frames taken with a camera. The bottom level should describe the states of the video frames within a shot, but these are hidden.

We extract the following features from frames: the averages and variances of RGB values (6D) and optical flows (4D). We then reduce the data dimensions from

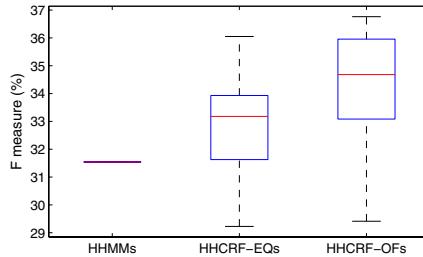


Fig. 3. Segmenting the video data according to the play units. Boxplot of 20 runs.

10 to 3 using principal component analysis (PCA). The training data comprises 47208 frames containing 109 segments, while the test data comprises 24579 frames containing 54 segments.

HHMMs have a single Gaussian density output. There are two kinds of HHCRFs, namely HHCRF-EQs and HHCRF-OFs. In HHCRF-EQs, the transition and observation feature functions are restricted to make the model structure equivalent to that of HHMMs. In HHCRF-OFs, the observation feature functions use the so-called *overlapping features*: at frame t , the observational feature functions take $o(t)$ and $o(t+1)$.

The parameters for the HHCRFs are initialized using the corresponding trained HHMM parameters. For the hidden top and second level state inference, which is necessary for segmentation, we use the forward-backward algorithm.

Figure 3 shows the performance of the three models in segmenting the video data according to the play units. A transition from a second level state to an end state corresponds to the end of a play unit. Therefore, the segmentation is done by detecting a transition of a second level state to the end state (i.e., detecting when f_t^2 becomes 1). We consider the transition to be correctly detected, if the inferred time for the transition is within a second of the actual transition time. We can see from the boxplot that almost all the accuracy rates of both kinds of HHCRFs are higher than those of the HHMMs. These results show that HHCRFs perform better than HHMMs in segmenting video data. In addition, it is clear that HHCRF-OFs outperform HHCRF-EQs. We conjecture that the performance gain comes from the overlapping feature of the observation. Furthermore, judging from the results of the first experiment, more training data will improve the performance of both kinds of HHCRFs.

6 Conclusion

In this paper, we have described HHCRFs, a discriminative model corresponding to HHMMs. HHCRFs directly model the conditional probability of the states at the upper levels given observations. The states at the lower levels are hidden, and marginalized in the model definition.

In applications of HHCRFs to labeling artificial data and sports video segmentation, we have shown that HHCRFs outperform HHMMs.

References

1. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning, pp. 633–723.
2. Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden Markov model: Analysis and applications. *Machine Learning* 32(1), 41–62 (1998)
3. Murphy, K., Paskin, M.: Linear time inference in hierarchical HMMs. In: Advances in Neural Information Processing Systems, vol. 14, pp. 833–840 (2001)
4. Sugiura, T., Gotou, N., Hayashi, A.: A discriminative model corresponding to hierarchical HMMS. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 375–384. Springer, Heidelberg (2007)
5. Gotou, N., Hayashi, A., Suematsu, N.: Learning with segment boundaries for hierarchical HMMs. In: Proceedings of the 3rd International Conference on Advances in Pattern Recognition, pp. 538–543 (2005)
6. Sutton, C., McCallum, A., Rohanimanesh, K.: Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *The Journal of Machine Learning Research* 8, 693–723 (2007)
7. Ghahramani, Z., Jordan, M.I.: Factorial hidden Markov models. In: Advances in Neural Information Processing Systems, vol. 8, pp. 472–478 (1995)
8. Liao, L., Fox, D., Kautz, H.: Hierarchical conditional random fields for GPS-based activity recognition. In: Proceedings of the 12th International Symposium on Robotics Research (2005)
9. Gunawardana, A., Mahajan, M., Acero, A., Platt, J.C.: Hidden conditional random fields for phone classification. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 1117–1120 (2005)
10. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifier: A comparison of logistic regression and naive bayes. In: Advances in Neural Information Proceeding Systems, vol. 16, pp. 841–848 (2002)
11. Xie, L., Chang, S.F., Divakaran, A., Sun, H.: Unsupervised discovery of multilevel statistical video structures using hierarchical hidden Markov models. In: Proceedings of the 2003 International Conference on Multimedia and Expo., vol. 3, pp. III– 29–III– 32(2003)
12. Wang, F., Ma, Y.F., Zhang, H.J., Li, J.T.: A generic framework for semantic sports video analysis using dynamic bayesian networks. In: Proceedings of the 11th International Multimedia Modelling Conference (MMM 2005), pp. 115–122 (2005)
13. Barnard, M., Odobezi, J.M.: Sports event recognition using layered hmms. In: IEEE International Conference on Multimedia and Expo., pp. 1150–1153 (2005)
14. Kijak, E., Oisel, L., Gros, P.: Hierarchical structure analysis of sport videos using hmms. In: Proceedings of the 2003 International Conference on Image Processing, vol. 3, pp. II– 1025–II– 1028(2003)

Learning Manifolds for Bankruptcy Analysis

Bernardete Ribeiro¹, Armando Vieira², João Duarte², Catarina Silva¹,
João Carvalho das Neves³, Qingzhong Liu⁴, and Andrew H. Sung⁴

¹ CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
bribeiro@dei.uc.pt

² Physics Department, Polytechnic Institute of Porto, Portugal
asv@isep.pt

³ ISEG - School of Economics, Technical University of Lisbon, Portugal
jcneves@iseg.utl.pt

⁴ Computer Science Department, University of New Mexico Tech, USA
sung@cs.nmt.edu

Abstract. We apply manifold learning to a real data set of distressed and healthy companies for proper geometric tuning of similarity data points and visualization. While Isomap algorithm is often used in unsupervised learning our approach combines this algorithm with information of class labels for bankruptcy prediction. We compare prediction results with classifiers such as Support Vector Machines (SVM), Relevance Vector Machines (RVM) and the simple k -Nearest Neighbor (KNN) in the same data set and we show comparable accuracy of the proposed approach.

1 Introduction

Manifold learning is an approach to nonlinear dimension reduction. In many real applications observational high-dimensional data can be cast into low-dimensional manifolds with few degrees of freedom and embedded intrinsic geometry. Instead of working with points in a high-dimensional space as a consequence of the high number of features of the real problem, a function with few parameters is learned during the reduction procedure. A direct consequence of nonlinear dimension reduction is the visualization of data which can help to reveal the data structures. Manifold methods include a number of nonlinear approaches to data analysis that exploit the geometric properties of the manifold on which the data is supposed to lie. These include algorithms like Isomap [1], LLE (Local Linear Embedding) [2], Laplacian Eigenmaps [3] and their variants.

Predicting bankruptcy has been a very important topic in accounting and finance attracting considerable research both from academic and business areas [4][5]. The question of how to determine the credit-worthiness of a customer or how safe is to grant credit remains a main concern for banks and investors, particularly, with the recent financial crisis. The problem of bankruptcy prediction can be addressed as follows: given a set of financial ratios describing the situation of a company over a given period, predict the probability that this company may become bankrupted in a near future, normally during the following year.

In this work, we present an approach for data visualization and bankruptcy analysis in a financial setting which is a less studied problem than prediction models. In [6] a self-organizing map (SOM) is computed allowing to visualize the data space in a topology-preserving manner. However, the method involves the estimation of the conditional probability which is computationally expensive for the numerical computations of the Fisher information matrix. Our approach incorporates class label information for embedding the intrinsic similarity of financial data in low dimension spaces by means of the Isomap [11] algorithm. We use a large database of three thousands distressed French companies (and sixty thousand healthy ones) containing information on a wide set of financial ratios spanning over a period of five years. In this study we analyze Isomap visualization power and show it only works well when data are comprised of one, well sampled cluster problem. However, in the financial data comprising 30 financial ratios the mappings get significantly worst due to its multi-cluster nature (Healthy and Distressed firms). We propose to overcome this limitation by presenting a supervised approach to the Isomap algorithm where the prior knowledge of a variable (indicating bankruptcy risk) is incorporated into a dissimilarity matrix. Once the low-dimension manifold is estimated, the embedded mapping is learned using a generalized regression neural network [7]. Finally a classifier in this reduced space is designed for testing new points. Several techniques have been compared: i) k -nearest neighbor (KNN) ii) Fisher Linear Discriminant (FLD) iii) Support Vector Machines (SVM).

The combined algorithm is compared to the state-of-the-art kernel learning machines (SVM) [8] and Relevance Vector Machines (RVM) [9]. The results show comparable accuracy in the financial setting despite the reduced space i.e., fewer dimensions than those used by the kernel approaches. In the remainder of the paper we will consider the following notation for the problem of dimensionality reduction.

- The n high-dimensional input points will be referred to as x_1, x_2, \dots, x_n . D is the dimensionality of the input (i.e. $x_i \in \mathbb{R}^D$).
- The low-dimensional output representations will be referred to as y_1, y_2, \dots, y_n . d is the dimensionality of the manifold that the input is assumed to lie on and, accordingly, the dimensionality of the output (i.e. $y_i \in \mathbb{R}^d$).
- k is the number of nearest neighbors used by a particular algorithm.

The paper is organized as follows. In section 2 manifold learning and Isomap are briefly described. In section 3 the approach is introduced as well as related work on supervised dimension reduction. In section 4 data sets for the bankruptcy analysis and prediction problem are described. In section 5 performance results are presented and discussed. We conclude in section 6 and give directions for future research.

2 Manifold Learning

Given data points $x_1, x_2, \dots, x_n \in \mathbb{R}^D$ we assume that data lie on a d -dimensional manifold embedded into \mathbb{R}^D , where $d < D$. Moreover, we assume the manifold M can be described by a single coordinate chart $f : M \longrightarrow \mathbb{R}^d$. The manifold learning consists of finding $y_1, \dots, y_n \in \mathbb{R}^d$, where $y_i = f(x_i)$. Isomap is an algorithm [11] for non linear dimension reduction which can be viewed as an extension of the Multidimensional Scaling (MDS) [10]. Isomap consists of three main steps:

1. Estimates which points are neighbors on the manifold M , based on the distances $d_X(i, j)$ between pairs of points i, j in the input space X by computing the weighted graph G of neighborhood relations given by the edges of weight $d_X(i, j)$.
2. Estimates the geodesic distances between all pairs of data points in the manifold M by computing the shortest path distance on the k 's nearest neighbor graph built on the data set.
3. Applies classical MDS to the matrix of graph distances $D_G = \{d_G(i, j)\}$, constructing an embedding of the data in a d -dimensional Euclidean space Y that best preserves the manifold's estimated intrinsic geometry

Isomap assumes that there is an isometric chart that preserves distances between points. If x_i and x_j are two points in the manifold M embedded into \mathbb{R}^D and the geodesic distance between them is $d_G(x_i, x_j)$, then there is a chart $f : M \longrightarrow \mathbb{R}^d$ such that $\|f(x_i) - f(x_j)\| = d_G(x_i, x_j)$

It is assumed that for nearby points in the high-dimensional space the Euclidean distance is a good approximation of the geodesic distance whereas for distant points this is not true. Therefore, a weighted graph with k 's nearest neighbors is built where its edges are weighted by the Euclidean distances between nearby data points. Then a shortest path computation algorithm such as, Dijkstra's or Floyd's, will complete the calculus of the remainder geodesic distances. MDS is then used to estimate the points whose Euclidean distance equals the geodesic distances. Given a matrix $D \in \mathbb{R}^{n \times n}$ of dissimilarities, MDS constructs a set of points whose interpoint Euclidean distances closely match those in D .

3 Supervised Nonlinear Dimensionality Reduction

Recent research on supervised nonlinear dimensionality reduction includes the approach in [11][12] in which S-Isomap (Semi-Supervised Isomap) is used as a preprocessing step before classification can take place. The rationale here is to map the high-dimensional

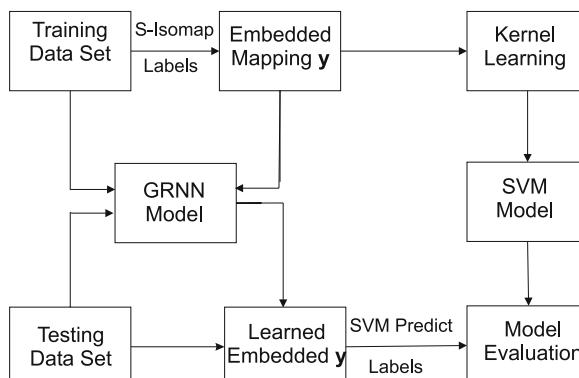


Fig. 1. S-Isomap Semi-Supervised approach

data space into a lower dimensional space where classification methods do not suffer from the curse of dimensionality. As the explicit mapping is not found by the algorithm some learning methodology must be used. Our approach, depicted in Figure 1, uses the training labels in the data set to provide a better construction of features. We further apply the dissimilarity measure (1) [12]

$$D(x_i, x_j) = \begin{cases} ((a - 1)/a)^{1/2} & \text{if } c_i = c_j \\ a^{1/2} - d_0 & \text{if } c_i \neq c_j \end{cases} \quad (1)$$

where $a = 1/e^{-d_{ij}^2/\sigma}$ with d_{ij} set to one of the distance measures described above, σ is a smoothing parameter (set according to the data ‘density’), d_0 is a constant ($0 \leq d_0 \leq 1$) and c_i, c_j are the data class labels. When a reduced space is reached, our aim is to learn a kernel-based model that can be applied in new cases of failed and non-failed firms. As we can not generate the test set directly, since we would need the labels, we use a generalized regression neural network (GRNN) [7] to learn the mapping, before the SVM prediction phase.

4 Experimental Setup

We used a sample obtained from Diane, a database containing financial statements of French companies. The initial sample consisted of financial ratios of about 60 000 industrial French companies, for the years of 2002 to 2006, with at least 10 employees. From these companies, about 3000 were declared bankrupted in 2007 or presented a restructuring plan (“Plan de redressement”) to the court for approval by the creditors. We decided not to distinguish these two categories as both signals companies in financial distress. The dataset includes information about 30 financial ratios produced by COFACE¹ described in Table 1 which allow the description of firms in terms of the financial strength, liquidity, solvability, productivity of labor and capital, margins, net profitability and return on investment.

4.1 Preprocessing

The database contains many cases with missing values, especially for defaults companies (distressed or bankruptcy). For this reason we sorted out the default cases by the number of missing values and selected the examples with 10 missing values at most. A final set of 600 default examples was obtained. In order to obtain a balanced dataset we selected randomly 600 non-default examples resulting in a set of 1200 examples. The remaining missing data was treated as follows. For the ratios of the years 2003 and 2006 each missing value was replaced by the closest available year value; for 2004 and 2005, if values of the next and previous years were available, each missing value was replaced by their mean, otherwise it was replaced by the remaining value. In some cases there was no data available for a ratio in any of the years. In this very few cases the missing data was replaced by the median value of the ratio in each year. Finally, all ratios were logarithmized and then standardized to zero mean and unity variance.

¹ Coface is a credit risk provider in France that defines a bankruptcy score based on discriminant analysis.

Table 1. DIANA Data Base

FINANCIAL RATIOS	
DIANA Data Base	1. Number of employees
	2. Financial Debt/Capital Employed %
	3. Capital Employed / Fixed Assets
	4. Depreciation of Tangible Assets (%)
	5. Working capital / current assets
	6. Current ratio
	7. Liquidity ratio
	8. Stock Turnover days
	9. Collection period
	10. Credit Period
	11. Turnover per Employee (thousands euros)
	12. Interest / Turnover
	13. Debt Period days
	14. Financial Debt / Equity (%)
	15. Financial Debt / Cashflow
	16. Cashflow / Turnover (%)
	17. Working Capital / Turnover (days)
	18. Net Current Assets/Turnover (days)
	19. Working Capital Needs / Turnover (%)
	20. Export (%)
	21. Value added per employee
	22. Total Assets / Turnover
	23. Operating Profit Margin (%)
	24. Net Profit Margin (%)
	25. Added Value Margin (%)
	26. Part of Employees (%)
	27. Return on Capital Employed (%)
	28. Return on Total Assets (%)
	29. EBIT Margin (%)
	30. EBITDA Margin (%)

4.2 Incorporating Historical Data

Companies are often subjected to fluctuation of the market, economy cycles and unavoidable contingencies related to its business activity. Therefore, we need to incorporate historical information in our dataset. In fact, yearly variations of important financial ratios reflecting the balance sheet, sometimes quite relevant, are common particularly for small companies. We included information from the past 3 years preceding the default. The number of inputs is therefore increased from 30 to 90 ratios. Moreover, more relevant than the ratios themselves, are the variations that occur over the period range of the analysis. Therefore we built four data sets for corporate prediction in 2007:

- 2006: previous year only (30 financial ratios)
- 2005: two years before bankruptcy (30 financial ratios)
- 2006-2005: 3 years, 2006, 2005 and 2004
- 2005-2004: 3 years, 2005, 2004 and 2003

5 Results and Discussion

For visualization purposes we first estimate the trustworthiness of the nonlinear manifold feature space. The trustworthiness measure quantifies how trustworthy is a projection of a high-dimensional data set onto a low-dimensional space. A projection is trustworthy if the set of the k nearest neighbors of each data point in the low-dimensional space are also close-by in the original space. More formally, the trustworthiness measure T is defined as [13]:

$$M(k) = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^N \sum_{j \in U_k(i)} (r(i, j) - k) \quad (2)$$

where $r(i, j)$ is the rank of the data point j in the ordering according to the distance from i in the original data space, and $U_k(i)$ denotes the set of those data points that are among the k -nearest neighbors of the data point i in the low-dimensional space but not in the original space. The maximal value of trustworthiness is equal to one. The closer $M(k)$ is to one, the better the low-dimensional space describes the original data.

Other measures have been defined based on the usual contingency matrix for binary classification such as Recall ($\frac{tp}{tp+fn}$) and Precision ($\frac{tp}{tp+fp}$). Also important is Error type I ($\frac{fp}{fp+tn}$) which represents the number of companies classified as bankrupt when in reality they are healthy, divided by the total number of healthy companies; and Error type II ($\frac{fn}{fn+tp}$) which is the number of samples classified as healthy when they are observed to be bankrupt, divided by the number of bankrupt companies. Error Rate is defined as ($\frac{fp+fn}{tp+fp+fn+tn}$). Figure 2 shows the visible separation of patterns (from healthy to bankrupt firms) with prior knowledge of the bankruptcy risk variable incorporated into the dissimilarity matrix. It was obtained for low nonlinear dimension reduction ($nldr=3$). In the right picture, trustworthiness measures are also shown averaged over 10 experiments of 10 cross-validation for the data set incorporating historical data (2006-2005). The data set with 88 features was reduced to $nldr = 3$, $nldr = 5$ and $nldr = 10$. As expected the trustworthiness is higher for $nldr=10$ taking the value $T = 0.9$ for $k = 3$. It is observed that fewer nearest neighbors, the better the projection and thus the trustworthiness.

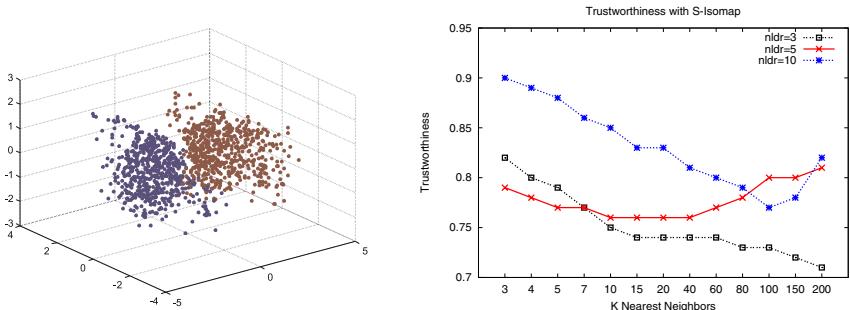


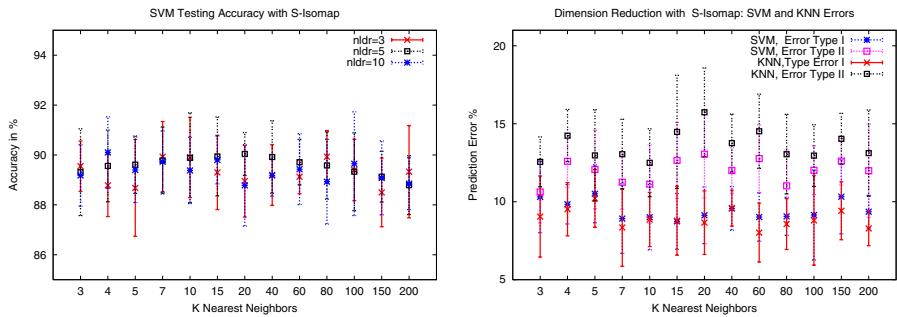
Fig. 2. Data visualization with historical DataSet by S-Isomap

Table 2 illustrates the performance results for all considered data sets with S-Isomap and with the learning algorithms KNN, SVM and RVM. In case of S-Isomap, results are indicated for 3-dimension reduced space. S-Isomap presents better results in testing accuracy than single KNN and RVM by 2% and 10%, respectively; and comparable one with SVM (slightly better by 2.5% with the latter), however, with much reduced embedded space ($nldr=3$) whereas SVM algorithm is used with all financial ratios.

Figure 3 shows the results for the average testing accuracy, using several k , with the historical data set 2006-2005 where SVM is used as a classifier in the final stage. Standard deviations are also indicated. The right picture allows to compare performances using SVM and KNN. As observed, the error of type II, corresponding to a failure of the correct prediction of bankruptcy is lower for the SVM. The same happens with a

Table 2. Performance Measures on Diana Financial DataSet

S-Isomap	Train	Test	recall	precision	errorTypeI	errorTypeII
2006	91.85 ± 0.54	87.73 ± 1.54	86.79 ± 2.62	87.94 ± 1.96	11.30 ± 1.91	13.21 ± 2.62
2005	78.70 ± 0.91	77.08 ± 2.02	77.13 ± 2.66	76.64 ± 3.62	22.87 ± 3.37	22.87 ± 2.66
2006-2005	94.26 ± 0.41	89.55 ± 1.01	89.38 ± 1.98	89.72 ± 1.94	10.31 ± 2.30	10.62 ± 1.98
2005-2004	96.74 ± 0.27	79.65 ± 1.42	77.61 ± 2.71	80.61 ± 2.12	18.38 ± 2.79	22.39 ± 2.71
KNN	Train	Test	recall	precision	errorTypeI	errorTypeII
2006	90.92 ± 0.76	85.77 ± 1.68	77.95 ± 3.29	92.51 ± 2.00	6.32 ± 1.69	22.05 ± 3.29
2005	84.78 ± 0.76	76.86 ± 1.71	73.22 ± 3.33	79.02 ± 1.98	19.46 ± 1.66	26.78 ± 3.33
2006-2005	91.18 ± 1.00	86.09 ± 1.88	76.99 ± 3.87	94.22 ± 3.03	4.74 ± 2.81	23.01 ± 3.87
2005-2004	84.39 ± 0.81	75.60 ± 1.79	64.80 ± 3.50	82.72 ± 1.65	13.58 ± 1.38	35.20 ± 3.50
SVM	Train	Test	recall	precision	errorTypeI	errorTypeII
2006	95.09 ± 0.42	90.54 ± 1.28	89.33 ± 2.24	91.73 ± 1.76	8.19 ± 1.90	10.67 ± 2.24
2005	86.06 ± 0.76	81.63 ± 1.76	81.01 ± 3.81	82.42 ± 2.84	17.64 ± 2.92	18.99 ± 3.81
2006-2005	95.85 ± 0.55	91.18 ± 1.28	92.10 ± 1.93	90.56 ± 1.69	9.74 ± 1.72	7.90 ± 1.93
2005-2004	89.93 ± 0.66	80.29 ± 1.54	81.04 ± 2.34	79.81 ± 2.58	20.42 ± 2.53	18.96 ± 2.34
RVM	Train	Test	recall	precision	errorTypeI	errorTypeII
2006	97.88 ± 0.63	81.25 ± 1.78	67.35 ± 2.98	92.31 ± 1.98	5.39 ± 2.01	32.65 ± 1.45
2005	93.25 ± 0.54	76.75 ± 1.25	72.64 ± 2.19	79.35 ± 2.34	19.09 ± 1.78	27.36 ± 2.03
2006-2005	99.68 ± 0.35	80.71 ± 2.11	72.47 ± 6.08	89.47 ± 2.55	8.71 ± 2.56	27.53 ± 6.08
2005-2004	100.00 ± 0.0	70.75 ± 1.74	65.36 ± 2.29	73.68 ± 1.53	23.46 ± 1.03	34.64 ± 2.29

**Fig. 3.** S-Isomap with Euclidean distance combined with KNN and SVM

false alarm, i.e., indicating a bankruptcy for a healthy firm, which corresponds to the error of type I. The fact that firms clump nicely in the reduced space not only enhances financial data visualization but also improves prediction results as compared with the kernel machines.

6 Conclusions

In this paper we proposed an approach for bankruptcy analysis and prediction based on a supervised Isomap algorithm where class label information is incorporated. Assuming that corporate financial statuses lie in a manifold we attempt to uncover this embedded

structure using manifold learning. Isomap acts as a preprocessing stage allowing financial data visualization. Despite the final generalization error has to take into account two factors: the first brought by the method itself and the second brought by the interpolation method, results have shown that comparable testing accuracy can be obtained even using a 3-dimensional reduced space. Although the results in the finance setting seem promising, further work is foreseen in the improvement of a method for avoiding the interpolation error resulting from the mapping learning stage.

Acknowledgments. Financial support from “Fundação da Ciência e Tecnologia” under the project PTDC/GES/70168/2006 is gratefully acknowledged.

References

1. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500), 2319–2323 (2000)
2. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500), 2323–2326 (2000)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, pp. 585–591. MIT Press, Cambridge (2002)
4. Altman, E.I.: *Corporate Financial Distress and Bankruptcy: A Complete Guide to Predicting and Avoiding Distress and Profiting from Bankruptcy*, 2nd edn. John Wiley & Sons, New York (1993)
5. Atiya, A.F.: Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Trans. Neural. Net.* 12(4) (2001)
6. Kaski, S., Sinkkonen, J., Peltonen, J.: Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks* 12(4), 936–947 (2001)
7. Specht, D.: A general regression neural network. *IEEE Transactions on Neural Networks* 2(6), 568–576 (1991)
8. Vapnik, V.N.: *The nature of statistical learning theory*. Springer, New York (1995)
9. Tipping, M.E.: Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* 1, 211–244 (2001)
10. Cox, T.F., Cox, M.A.A.: *Multidimensional Scaling*, 2nd edn. Chapman and Hall/CRC, Boca Raton (2001)
11. Vlachos, M., Domeniconi, C., Gunopoulos, D., Kollios, G., Koudas, N.: Non-linear dimensionality reduction techniques for classification and visualization. In: *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 69, pp. 645–651. ACM, New York (2002)
12. Geng, X., Zhan, D.G., Zhou, Z.H.: Supervised nonlinear dimensionality reduction in visualization and classification. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics* 35(6), 1098–1107 (2005)
13. Venna, J., Kaski, S.: Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In: *Workshop of Self-Organizing Maps*, pp. 695–702 (2005)

Information Geometry of Interspike Intervals in Spiking Neurons with Refractories

Daisuke Komazawa¹, Kazushi Ikeda², and Hiroyuki Funaya²

¹ Graduate School of Informatics, Kyoto University, Kyoto 606-8501 Japan
komazawa@sys.i.kyoto-u.ac.jp

² Graduate School of Information Science, Nara Institute of Science and Technology,
Ikoma 630-0192 Japan
kazushi@is.naist.jp, hiroyuki-fn@is.naist.jp

Abstract. An information geometrical method is developed for characterizing or classifying neurons in cortical areas, whose spike rates fluctuate in time. When the interspike intervals of a spike sequence of a neuron obey a gamma process with a time-variant spike rate and a fixed shape parameter, the information geometry for semiparametric estimation has given the optimal method from the statistical viewpoint. Recently a more suitable statistical model for interspike intervals is proposed, which have an absolute refractory period. This work extends the information geometrical method and derives the optimal method for the new model.

1 Introduction

Recently, the characteristics of neurons in cortical areas have been discussed based on the statistical properties of the interspike intervals (ISIs) of a spike sequence, such as the coefficient of variation, C_V , the skewness coefficient, S_K , the correlation coefficient of consecutive intervals, C_{OR} , and the local variation, L_V [1,2,3,4,5]. In particular, the local variation, L_V , has attracted much attention since L_V is robust against changes of the spike rate as well as it provided an interesting problem from the statistical viewpoint.

In the theoretical analyses, ISIs can be modeled as a gamma process with a variable rate but a fixed shape parameter [5,6]. Since gamma distributions form a two-dimensional e -flat manifold S from the information-geometrical viewpoint [7,8,9], the problem of estimating the shape parameter results in a semiparametric estimation [10,11].

Shinomoto has recently claimed that the local variation with a refractory period, L_{VR} , is more suitable to classify neurons from their interspike intervals [12]. In this study, we give a theoretical support for L_{VR} using the information geometrical analysis for the semiparametric estimation. More concretely, we extend the analysis in [11] by adding the refractory parameter R to the statistical model, where R is also a parameter to be estimated from the interspike intervals.

In the following, we introduce some statistical measures of a spike sequence and their properties in Section 2. Section 3 formulates the problem of ISIs from the statistical viewpoint and the answer is given in Section 4. Section 5 shows our conclusions.

2 Statistical Measures for ISIs

When a spike sequence is given and its N ISIs are written as T_1, T_2, \dots, T_N , the C_V and S_K measures are defined as the standard deviation of ISIs divided by the mean of ISIs and the skewness of ISI distribution, respectively. That is,

$$C_V = \frac{1}{\bar{T}} \sqrt{\frac{1}{N-1} \sum_{n=1}^N (T_n - \bar{T})^2}, \quad (1)$$

$$S_K = \frac{1}{N-1} \sum_{n=1}^N (T_n - \bar{T})^3 / \left(\frac{1}{N-1} \sum_{n=1}^N (T_n - \bar{T})^2 \right)^{3/2}, \quad (2)$$

where

$$\bar{T} = \frac{1}{N} \sum_{n=1}^N T_n. \quad (3)$$

The C_V measure expresses the regularity: It takes a low value for a regular spike sequence, one for a sequence of an infinite length generated by a fixed Poisson process, and a large value when the process is time-dependent. The S_K measure shows the asymmetry of a sequence: It can be either positive or negative, but takes two for a sequence of an infinite length generated by a stationary Poisson process. Since they are based on the mean \bar{T} of ISIs, C_V or S_K will take a large value when the spike rate is globally modulated even though the spike sequence is locally quasi-regular [11, 13]. Due to this property, they are not suitable for classifying such neurons in cortical areas that change their spike rate drastically in a waiting period task, for example.

To overcome the problem, [5] proposed the L_V measure, defined as

$$L_V = \frac{1}{N-1} \sum_{n=1}^{N-1} \frac{3(T_n - T_{n+1})^2}{(T_n + T_{n+1})^2}, \quad (4)$$

where the factor 3 is taken so that the expectation of L_V becomes one when the sequence obeys a stationary Poisson process. Since the L_V measure reflects the stepwise variability of ISIs and does not compare ISIs with different spike rates, L_V can take a small value even for a sequence with a time-variant spike rate. They confirmed that C_V undergoes a large change but L_V does not for a sequence generated by a time-dependent Poisson process [13].

Recently, Shinomoto extended L_V so that it explicitly includes an absolute refractory period R to each neuron, where the value is common to all neurons. That is,

$$L_{VR} = \frac{1}{N-1} \sum_{n=1}^{N-1} \frac{3(T_n - T_{n+1})^2}{(T_n + T_{n+1} - 2R)^2}. \quad (5)$$

3 Statistical Model of Interspike Intervals

Suppose that an interspike interval T independently obeys a gamma distribution with parameters $\xi^{(l)}$, κ and R ,

$$q(T; \xi^{(l)}, \kappa, R) = \frac{(\xi^{(l)}\kappa)^{\kappa}}{\Gamma(\kappa)} (T - R)^{\kappa-1} \exp\left[-\xi^{(l)}\kappa(T - R)\right], \quad (6)$$

where l runs from 1 to N . This means that each neuron has an absolute refractory period R before firing according to a gamma distribution, which results in the existing models if R constantly takes zero.

When each T has a different $\xi^{(l)}$, we cannot estimate κ at all even if $R = 0$. So, we devide N ISIs into N/m sets, as is done in [10, 11], assuming that m observations, $\{T^{(l)}\} \equiv \{T_1^{(l)}, \dots, T_m^{(l)}\}$, are given for each $\xi^{(l)}$, where $\xi^{(l)}$ is generated from an unknown probability density $k(\xi)$. That is, the ISIs in the l th set is described as

$$p(\{T^{(l)}\}; \xi^{(l)}, \kappa, R) = \prod_{i=1}^m q(T_i^{(l)}; \xi^{(l)}, \kappa, R). \quad (7)$$

We would like to ascertain only κ and R and take no interest in $\xi^{(l)}$ or $k(\xi)$. Such a problem is called semiparametric statistical estimation and is known to be solved by

$$\sum_{l=1}^{N/m} \sum_{i=1}^m f(T_i^{(l)}, \kappa, R) = 0, \quad (8)$$

where $f(T; \kappa, R)$ is called an estimating function [14, 15], which satisfies

$$E_{\kappa, R, k}[f(T; \kappa, R)] = 0 \quad (9)$$

for any κ , R and k , where $E_{\kappa, R, k}$ denotes the expectation with respect to the distribution

$$p(T; \kappa, R, k) = \int p(T; \xi, \kappa, R) k(\xi) d\xi. \quad (10)$$

Information geometry of estimating functions [16] gave how to derive the optimal estimating function, which was also utilized in analyzing ISIs [11]. In the sequel, we extend the analysis in [11] to the case where the refractory period R is common and constant.

4 Estimation by Estimating Functions

See section 3 of [11] for a summary of the theory of estimating functions. We employ the same notation as [11] and omit the superscripts (l) for $\xi^{(l)}$ and $T_i^{(l)}$ in the following.

Since we assume m observations for each ξ , the probability model is written as

$$p(\{T\}; \kappa, R, k(\xi)) = \int \prod_{i=1}^m q(T_i; \xi, \kappa, R) k(\xi) d\xi, \quad (11)$$

where

$$\prod_{i=1}^m q(T_i; \xi, \kappa, R) = \exp [\xi \cdot s(\{T\}, \kappa) + r(\{T\}, \kappa, R) - \psi(\kappa, R, \xi)], \quad (12)$$

$$s(\{T\}, \kappa) \equiv -\kappa \sum_{i=1}^m T_i, \quad (13)$$

$$r(\{T\}, \kappa, R) \equiv (\kappa - 1) \sum_{i=1}^m \log(T_i - R), \quad (14)$$

$$\psi(\kappa, R, \xi) \equiv -m\kappa \log(\xi\kappa) + m \log \Gamma(\kappa) - m\xi\kappa R. \quad (15)$$

Then, the optimal estimating functions for κ and R are given by

$$u_\kappa^I(\{T\}, \kappa, R) = u_\kappa - E[u_\kappa | s] \quad (16)$$

$$= \sum_{i=1}^m \log(T_i - R) - mE[\log(T_1 - R) | s], \quad (17)$$

$$u_R^I(\{T\}, \kappa, R) = u_R - E[u_R | s] \quad (18)$$

$$= (1 - \kappa) \sum_{i=1}^m \frac{1}{T_i - R} - m(1 - \kappa) E \left[\frac{1}{T_1 - R} | s \right], \quad (19)$$

where u_κ^I and u_R^I are defined so that they are orthogonal to any function of s . The marginal distribution of s and the conditional expectation in the last term of (17) are given in Appendix of [11] by

$$p(s) = \int \delta \left[s + \kappa \sum_{i=1}^m T_i \right] \prod_{i=1}^m q(T_i; \xi, \kappa, R) dT_i k(\xi) d\xi \quad (20)$$

$$= \prod_{i=1}^{m-1} B(i\kappa, \kappa) \left(-\frac{s}{\kappa} - mR \right)^{m\kappa-1} \frac{(\xi\kappa)^{m\kappa}}{\Gamma(\kappa)^m} \exp[\xi s + \xi\kappa mR] \frac{k(\xi)}{\kappa} d\xi, \quad (21)$$

$$E[\log(T_1 - R) | s] = \int \log(T_1 - R) \delta \left[s + \kappa \sum_{i=1}^m T_i \right] \prod_{i=1}^m q(T_i; \xi, \kappa, R) dT_i k(\xi) d\xi \frac{1}{p(s)} \quad (22)$$

$$= \log \left[-\frac{s}{\kappa} - mR \right] - \phi(m\kappa) + \phi(\kappa), \quad (23)$$

where δ is the Dirac delta function and the digamma function $\phi(\kappa)$ is defined as

$$\phi(\kappa) = \frac{\Gamma'(\kappa)}{\Gamma(\kappa)}. \quad (24)$$

Similarly, we can derive the conditional expectation in the last term of (19) as

$$\mathbb{E} \left[\frac{1}{T_1 - R} | s \right] = \int \frac{1}{T_1 - R} \delta \left[s + \kappa \sum_{i=1}^m T_i \right] \prod_{i=1}^m q(T_i; \xi, \kappa, R) dT_i k(\xi) d\xi \frac{1}{p(s)} \quad (25)$$

$$= \frac{1}{-s/\kappa - mR} \left(1 - \frac{m\kappa - \kappa}{\kappa - 1} \right). \quad (26)$$

In total, the optimal estimating functions are written as

$$u_\kappa^I(\{T\}, \kappa, R) = \sum_{i=1}^m \log(T_i - R) - m \log \sum_{i=1}^m (T_i - R) + m\phi(m\kappa) - m\phi(\kappa), \quad (27)$$

$$u_R^I(\{T\}, \kappa, R) = (1 - \kappa) \sum_{i=1}^m \frac{1}{T_i - R} - \frac{m(1 - \kappa)}{\sum_{i=1}^m 1/(T_i - R)} \left(1 - \frac{m\kappa - \kappa}{\kappa - 1} \right). \quad (28)$$

From [16], the estimators of κ and R are given by the solution of the equations,

$$\sum_{l=1}^N u_\kappa^I(\{T^{(l)}\}, \kappa, R) = 0, \quad \sum_{l=1}^N u_R^I(\{T^{(l)}\}, \kappa, R) = 0. \quad (29)$$

We can confirm that the estimator does not depend on ξ as follows. Suppose that $T - R$ obeys a gamma distribution with parameters ξ and κ . Then, $t - R \equiv (T - R)/\xi$ obeys the standard gamma distribution and its optimal estimating functions, v_κ^I and v_R^I , are written as

$$v_\kappa^I(\{t\}, \kappa, R) = \sum_{i=1}^m \log(t_i - R) - m \log \sum_{i=1}^m (t_i - R) + m\phi(m\kappa) - m\phi(\kappa) \quad (30)$$

$$= u_\kappa^I(\{T\}, \kappa, R), \quad (31)$$

$$v_R^I(\{t\}, \kappa, R) = (1 - \kappa) \sum_{i=1}^m \frac{1}{t_i - R} - \frac{m(1 - \kappa)}{\sum_{i=1}^m 1/(t_i - R)} \left(1 - \frac{m\kappa - \kappa}{\kappa - 1} \right) \quad (32)$$

$$= u_R^I(\{T\}, \kappa, R)/\xi. \quad (33)$$

Clearly, these have the same solution as (29) no matter what positive value ξ takes.

5 Conclusions

In this paper, we extended the optimal method for characterizing interspike intervals of a spike sequence from the statistical or the information geometrical

viewpoints to a more general model which has an absolute refractory period, assuming that the interspike intervals of a spike sequence of a neuron obey a gamma process with a time-variant spike rate and a fixed shape parameter. Since the shape parameter has recently been elucidated to have a relationship with the function of each neuron, statistical methods for analyzing the characteristic has been getting important. Our future work includes applying our new method to practical physiological data.

Acknowledgments. This study is supported in part by a Grant-in-Aid for Scientific Research (18300078) from MEXT, Japan.

References

1. Holt, G.R., Softky, W.R., Koch, C., Douglas, R.J.: Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons. *Journal of Neurophysiology* 75, 1806–1814 (1996)
2. Shinomoto, S., Sakai, Y., Funahashi, S.: The ornstein-uhlenbeck process does not reproduce spiking statistics of neurons in prefrontal cortex. *Neural Computation* 11, 935–951 (1999)
3. Sakai, Y., Funahashi, S., Shinomoto, S.: Temporally correlated inputs to leaky integrate-and-fire models can reproduce spiking statistics of cortical neurons. *Neural Networks* 12, 1181–1190 (1999)
4. Shinomoto, S., Shima, K., Tanji, J.: New classification scheme of cortical sites with the neuronal spiking characteristics. *Neural Networks* 15(10), 1165–1169 (2002)
5. Shinomoto, S., Shima, K., Tanji, J.: Differences in spiking patterns among cortical neurons. *Neural Computation* 15(12), 2823–2842 (2003)
6. Tiesinga, P.H.E., Fellous, J.M., Sejnowski, T.J.: Attractor reliability reveals deterministic structure in neuronal spike trains. *Neural Computation* 14, 1629–1650 (2002)
7. Amari, S.I.: Differential-Geometrical Methods in Statistics. Lecture Notes in Statistics, vol. 28. Springer, Heidelberg (1985)
8. Amari, S.I., Nagaoka, H.: Methods of Information Geometry. Translations of Mathematical Monographs, vol. 191. AMS and Oxford Univ. Press, Oxford (2000)
9. Miura, K., Shinomoto, S., Okada, M.: Search for optimal measure to discriminate random and regular spike trains. Technical Report NC2004-52, IEICE (2004)
10. Ikeda, K.: Information geometry of interspike intervals in spiking neurons. *Neural Computation* 17(12), 2719–2735 (2005)
11. Miura, K., Okada, M., Amari, S.I.: Estimating spiking irregularities under changing environments. *Neural Computation* 18(10), 2359–2386 (2006)
12. Shinomoto, S.: Private communication
13. Shinomoto, S., Tsubo, Y.: Modeling spiking behavior of neurons with time-dependent poisson processes. *Physical Review E* 64, 41910 (2001)
14. Godambe, V.P.: Conditional likelihood and unconditional optimum estimating equations. *Biometrika* 63, 277–284 (1976)
15. Godambe, V.P. (ed.): *Estimating Functions*. Oxford Univ. Press, Oxford (1991)
16. Amari, S.I., Kawanabe, M.: Information geometry of estimating functions in semi-parametric statistical models. *Bernoulli* 2(3) (1996)

Convulsive Blind Speech Separation by Decorrelation*

Fuxiang Wang and Jun Zhang

School of Electronics and Information Engineering, Beihang University,
Beijing 100083, China
wfux@ee.buaa.edu.cn, buaazhangjun@vip.sina.com.cn

Abstract. This paper proposes a new method for convulsive blind speech separation by decorrelation. First, by the design of a FIR separating filter and a whitening process of the observed data, the separation of convulsive speech mixtures are transformed to find a semi-unitary separating matrix. Then we estimate the separating semi-unitary matrix by the semi-unitary joint diagonalization for a set of correlation matrices. And a numerical algorithm for semi-unitary joint diagonalization is proposed. Simulation results of speech separation demonstrate the effectiveness of the new approach.

1 Introduction

In recent years, many studies have been devoted to convulsive blind speech separation (BSS) because of its application in multi-channel speech enhancement systems.

Generally, the problem of covolutive BSS can be modeled as follows

$$\mathbf{x}(t) = \sum_{l=0}^{P-1} \mathbf{A}(l) \mathbf{s}(t-l) \quad (1)$$

where $\mathbf{s}(t) \in C^{N \times 1}$ and $\mathbf{x}(t) \in C^{M \times 1}$ are the speech sources and observed convulsive speech mixtures respectively. $\mathbf{A}(l) \in C^{M \times N}$ is the mixing matrix at time lag l . The sources are assumed to be zero-mean and mutually independent. The objective is to extract the sources using only the information within the observed convulsive speeches.

A few methods have been proposed as introduced as follows:

- Time-domain methods [1-3]. A FIR inverse filter \mathbf{H} of length Q is formulated to estimate the source speeches

$$\mathbf{y}(t) = \sum_{l=0}^{Q-1} \mathbf{H}(l) \mathbf{x}(t-l) \quad (2)$$

* This work is funded by the Chinese National Science Foundation for Outstanding Scholarship (Grant No: 60625102), the Key Project of National Natural Sciences Foundation of China (Grant No: 60532030).

where $\mathbf{y}(t) = (y_1(t), y_2(t), \dots, y_N(t))^T$, $\mathbf{H}(l) \in C^{N \times M}$. The purpose of demixing process is to make the output signals $\mathbf{y}(t)$ be as mutually independent as possible.

- Frequency-domain methods [4-6]. Equivalently, (1) can be characterized in frequency domain by writing

$$\mathbf{x}(w) = \mathbf{H}(w)\mathbf{s}(w) \quad (3)$$

Then the frequency domain is divided into many frequency bin. In each frequency bin, the problem is regarded as an instantaneous BSS problem and any complex-valued instantaneous methods for blind source separation can be employed to deal with it.

In this paper, a new approach for convolutive BSS in time-domain is presented by decorrelation. After a whitening process of the observed data, the problem of convolutive blind speech separation is reduced to find a semi-unitary separating matrix. Then we estimate the separating parauitary matrix by semi-unitary joint diagonalization for a set of correlation matrices. And a numerical algorithm for semi-unitary joint diagonalzaiton is proposed by the Jacobi technique.

The structure of the paper is organized as follows. In section II, the application of our new method is discussed. In section III, a numerical algorithm for the problem of semi-unitary joint diagonalization is proposed. Section IV presents some experimental results to show the effectiveness of our method. We summarize our conclusion in section V.

2 Convolutive BSS Method by Decorrelation

2.1 Whitening

The first step of our procedure is a whitening process shown in the following.
(2) can be reformulated as

$$\mathbf{y}(t) = \sum_{l=0}^{Q-1} \mathbf{H}(l) \mathbf{x}(t-l) = \mathbf{H}\mathbf{r}(t) \quad (4)$$

where

$$\mathbf{H} = (\mathbf{H}(0), \mathbf{H}(1), \dots, \mathbf{H}(Q-1)) \in C^{N \times MQ} \quad (5)$$

and

$$\mathbf{r}(t) = (\mathbf{x}^T(t), \mathbf{x}^T(t-1), \dots, \mathbf{x}^T(t-Q+1))^T \in C^{MQ \times 1} \quad (6)$$

The whitening process is achieved by applying a whitening matrix $\mathbf{W} \in C^{L \times MQ}$ to the data $\mathbf{r}(t)$ such that

$$E\{\mathbf{z}(t)\mathbf{z}(t)^T\} = E(\mathbf{W}\mathbf{r}(t)\mathbf{r}^T(t)\mathbf{W}^T) = \mathbf{W}E(\mathbf{r}(t)\mathbf{r}^T(t))\mathbf{W}^T = \mathbf{I}_{L \times L} \quad (7)$$

where $\mathbf{z}(t) = \mathbf{W}\mathbf{r}(t) \in C^{L \times 1}$, $\mathbf{I}_{L \times L}$ is a $L \times L$ identity matrix. The whitening matrix \mathbf{W} can be computed by the eigen-decomposition of the correlation matrix $\mathbf{R}_{rr} = E\{\mathbf{r}(t)\mathbf{r}(t)^H\}$ and L is equal to the number of non-zero eigenvalue of \mathbf{R}_{rr} .

The following process is based on the whitened data $\mathbf{z}(t) = \mathbf{W}\mathbf{r}(t)$. Consequently, the model (4) becomes

$$\mathbf{y}(t) = \mathbf{V}\mathbf{z}(t) \quad (8)$$

where $\mathbf{V} \in C^{N \times L}$ and $\mathbf{H} = \mathbf{V}\mathbf{W}$.

The correlation matrix of the output data in (8) is

$$\mathbf{R}_{yy} = E\{\mathbf{y}(t)\mathbf{y}^T(t)\} = E(\mathbf{V}\mathbf{z}(t)\mathbf{z}^T(t)\mathbf{V}^T) = \mathbf{V}\mathbf{V}^T \quad (9)$$

We assume that the source correlation matrix $\mathbf{R}_{ss} = \mathbf{I}_{N \times N}$, i.e., the sources have unit covariance matrix. Then we can require that the matrix $\mathbf{V} \in C^{N \times L}$ satisfies

$$\mathbf{V}\mathbf{V}^T = \mathbf{I}_{N \times N} \quad (10)$$

That implies that the $N \times L$ matrix \mathbf{V} is a semi-unitary matrix.

2.2 The Separation Criterion by Decorrelation

By utilizing the non-stationary of speech signals, some methods by decorrelation of the outputs have been proposed. The correlation matrices in different time lags τ_k can be expressed by

$$\mathbf{R}_{yy}(\tau_k) = E\{\mathbf{y}(t)\mathbf{y}(t+\tau_k)\} = \mathbf{V}\mathbf{E}\{\mathbf{z}(t)\mathbf{z}(t+\tau_k)\}\mathbf{V}^T = \mathbf{V}\mathbf{G}_k\mathbf{V}^T \quad (11)$$

where $\mathbf{G}_k = \mathbf{E}\{\mathbf{z}(t)\mathbf{z}(t+\tau_k)\}$.

Our aim is to find a semi-unitary matrix $\mathbf{V} \in C^{N \times L}$ that approximately diagonalizes the correlation matrices $\mathbf{G}_k \in C^{L \times L}$ in K different time lags τ_1, \dots, τ_K . As in [7,8], a least square criterion is defined to evaluate the diagonality of the transformed matrix $\mathbf{V}\mathbf{G}_k\mathbf{V}^T$

$$C_1 = \sum_{i=1}^K \text{off}(\mathbf{V}\mathbf{G}_k\mathbf{V}^H) \quad (12)$$

where $\text{off}(\bullet)$ represents the square sum of all non-diagonal elements of a matrix.

3 Algorithm for Semi-unitary Joint Diaognalziation

Since two arbitrary rows of semi-unitary matrix \mathbf{V} is mutually orthogonal, the matrix \mathbf{V} can be expressed by the leading N rows of an unitary matrix $\mathbf{U} \in C^{L \times L}$, that is

$$\mathbf{V} = \mathbf{E}\mathbf{U} \quad (13)$$

where $\mathbf{E} \in C^{N \times L}$ is the leading rows of the $L \times L$ identity matrix

$$\mathbf{E} = \begin{pmatrix} \mathbf{I}_{N \times N} & \mathbf{O}_{N \times (L-N)} \end{pmatrix} \quad (14)$$

Using (13), minimizing the LS cost function (12) becomes

$$\min \left\{ C_2 = \sum_{k=1}^K \text{off}(\mathbf{E}\mathbf{U}\mathbf{G}_k \mathbf{U}^T \mathbf{E}^T) \right\} \quad (15)$$

The ultimate diagonalizing matrix \mathbf{V} is given by $\mathbf{V} = \mathbf{E}\mathbf{U}^*$, where \mathbf{U}^* is the optimum of (15).

B. Jacobi Rotation optimization

A Jacobi rotation transform $\mathbf{J}(\theta, p, q) \in C^{L \times L}$ with four parameters θ, p, q is defined as

$$\mathbf{J}(\theta, p, q) = \begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cos \theta & \cdots & -\sin \theta & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \sin \theta & \cdots & \cos \theta & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \quad (16)$$

where p and q ($p < q$) stand for indices of rows and columns respectively.

The unitary matrix \mathbf{U} is initialized by the identity matrix. Then \mathbf{U} is updated as follows,

$$\mathbf{U} := \mathbf{J}(\theta, p, q)\mathbf{U} \quad (17)$$

Simultaneously, the target matrices are updated by

$$\mathbf{G}_k := \mathbf{J}(\theta, p, q)\mathbf{G}_k \mathbf{J}^T(\theta, p, q), \quad k = 1, \dots, K. \quad (18)$$

Note that \mathbf{U} and $\mathbf{J}(\theta, p, q)$ are unitary matrices, thus the updated matrix \mathbf{U} in (16) is an unitary matrix.

For fixed indices p and q , the parameters θ (and thus the transform matrix $\mathbf{J}(\theta, p, q)$) are determined by solve the following optimization problem

$$\min \left\{ C_3 = \sum_{k=1}^K \text{off}(\mathbf{E}\mathbf{J}(\theta, p, q)\mathbf{G}_k \mathbf{J}^T(\theta, p, q)\mathbf{E}^T) \right\} \quad (19)$$

In the following, we discuss the optimization problem (19) in three cases:

◆ Case 1: $p < q \leq N$

We rewrite the $L \times L$ matrix \mathbf{G}_k as

$$\mathbf{G}_k = \begin{pmatrix} \mathbf{G}_{11}^{(k)} & \mathbf{G}_{12}^{(k)} \\ \mathbf{G}_{21}^{(k)} & \mathbf{G}_{22}^{(k)} \end{pmatrix} \quad (20)$$

where $\mathbf{G}_{11}^{(k)} \in C^{N \times N}$, $\mathbf{G}_{12}^{(k)} \in C^{N \times (L-N)}$, $\mathbf{G}_{21}^{(k)} \in C^{(L-N) \times N}$ and $\mathbf{G}_{22}^{(k)} \in C^{(L-N) \times (L-N)}$.

The matrix $\mathbf{J}(\theta, p, q)$ in (16) is accordingly rewritten as

$$\mathbf{J}(\theta, \psi, p, q) = \begin{pmatrix} \mathbf{J}_{N \times N}(\theta, p, q) & \mathbf{O}_{N \times (L-N)} \\ \mathbf{O}_{(L-N) \times N} & \mathbf{I}_{(L-N) \times (L-N)} \end{pmatrix} \quad (21)$$

where $\mathbf{J}_{N \times N}(\theta, p, q)$ represents a $N \times N$ Jabobi matrix similar as in (16).

Using the notations in (14), (20) and (21), we have

$$\mathbf{E}\mathbf{J}(\theta, p, q)\mathbf{G}_k\mathbf{J}^H(\theta, p, q)\mathbf{E}^T = \mathbf{J}_{N \times N}(\theta, p, q)\mathbf{G}_{11}^{(k)}\left(\mathbf{J}_{N \times N}(\theta, p, q)\right)^T \quad (22)$$

The cost function in (19) becomes

$$C_3 = \sum_{k=1}^K off\left(\mathbf{J}_{N \times N}(\theta, p, q)\mathbf{G}_{11}^{(k)}\left(\mathbf{J}_{N \times N}(\theta, p, q)\right)^T\right) \quad (23)$$

Using the extend Jacobi algorithm [7], the optimization problem (23) is solved and $\mathbf{J}_{N \times N}(\theta, p, q)$ is obtained.

◆ Case 2: $p \leq N < q \leq M$

Let $\mathbf{J}(\theta, p, q) = (\mathbf{e}_1^T \ \cdots \ \mathbf{u}_p^T \ \cdots \ \mathbf{u}_q^T \ \cdots \ \mathbf{e}_M^T)^T$, where \mathbf{u}_p and \mathbf{u}_q denote the p -th and q -th row of $\mathbf{J}(\theta, p, q)$. From the definition of \mathbf{E} in (4), then $\mathbf{E}\mathbf{J}(\theta, p, q) = (\mathbf{e}_1^T \ \cdots \ \mathbf{u}_p^T \ \cdots \ \mathbf{e}_N^T)^T$. Hence, we have

$$\mathbf{E}\mathbf{J}(\theta, p, q)\mathbf{G}_k\mathbf{J}^H(\theta, p, q)\mathbf{E}^H = \begin{pmatrix} \mathbf{G}_k(1,1) & \mathbf{G}_k(1,2) & \cdots & \mathbf{e}_1\mathbf{G}_k\mathbf{u}_p^T & \cdots & \mathbf{G}_k(1,N) \\ \mathbf{G}_k(2,1) & \mathbf{G}_k(2,2) & \cdots & \mathbf{e}_2\mathbf{G}_k\mathbf{u}_p^T & \cdots & \mathbf{G}_k(2,N) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{u}_p\mathbf{G}_k\mathbf{e}_1^T & \mathbf{u}_p\mathbf{G}_k\mathbf{e}_2^T & \cdots & \mathbf{u}_p\mathbf{G}_k\mathbf{u}_p^T & \cdots & \mathbf{u}_p\mathbf{G}_k\mathbf{e}_N^T \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{G}_k(N,1) & \mathbf{G}_k(N,2) & \cdots & \mathbf{e}_N\mathbf{G}_k\mathbf{u}_p^T & \cdots & \mathbf{G}_k(N,N) \end{pmatrix} \quad (24)$$

where $\mathbf{G}_k(i, j)$ denotes the element in row i and column j of \mathbf{G}_k . It follows from (24) that the elements of $\mathbf{EJ}(\theta, p, q)\mathbf{G}_k\mathbf{J}^H(\theta, p, q)\mathbf{E}^H$ are constants with respect to Θ and Ψ except those elements in the p th row and the p th column. The optimization of (19) is equivalent to minimize the following square sum of all non-diagonal elements in the p th row and p th column with respect to the parameters Θ and Ψ ,

$$C_4 = \sum_{k=1}^K \left[\sum_{l=1, l \neq p}^N |\mathbf{u}_p \mathbf{G}_k \mathbf{e}_l^T|^2 + \sum_{l=1, l \neq p}^N |\mathbf{e}_l \mathbf{G}_k \mathbf{u}_p^T|^2 \right] = \mathbf{u}_p \mathbf{R} \mathbf{u}_p^T, \quad (25)$$

where the symmetric matrix $\mathbf{R} \in \mathbb{C}^{M \times M}$ is defined as

$$\mathbf{R} = \sum_{k=1}^K \mathbf{G}_k \left(\sum_{l=1, l \neq p}^N \mathbf{e}_l^T \mathbf{e}_l \right) \mathbf{G}_k^T + \sum_{k=1}^K \mathbf{G}_k^T \left(\sum_{l=1, l \neq p}^N \mathbf{e}_l^T \mathbf{e}_l \right) \mathbf{G}_k = (r_{i,j})_{L \times L} \quad (26)$$

Since \mathbf{u}_p is the p -th row of $\mathbf{J}(\theta, p, q)$ in (16) and has only two nonzero elements, the cost function (25) can be simplified as

$$C_4 = \frac{r_{pp} + r_{qq}}{2} + \frac{(r_{pp} - r_{qq})}{2} \cos 2\theta - r_{pq} \sin 2\theta \quad (27)$$

By letting $\mathbf{G}_k = (g_{i,j}^{(k)})_{L \times L}$, r_{pq} , r_{qp} , r_{pp} , and r_{qq} in (27) are calculated by

$$r_{i,j} = \sum_{k=1}^K \sum_{l=1, l \neq i}^n \left[g_{i,l}^{(k)} g_{j,l}^{(k)} + g_{i,l}^{(k)} g_{j,l}^{(k)} \right] \quad (28)$$

where $i, j = p, q$.

We now discuss the minimization of the cost function in (27) with respect to Θ and Ψ . There are two sub-cases:

(a) $r_{pp} - r_{qq} = 0$ and $r_{pq} = 0$

In this sub-case, (27) becomes

$$C_4 = \frac{r_{pp} + r_{qq}}{2}$$

since C_4 is not related to Θ , we can arbitrarily set Θ , e.g. $\Theta^* = 0$.

(b) $r_{pp} - r_{qq} \neq 0$ or $r_{pq} \neq 0$

(27) can be rewritten as

$$C_4 = \frac{r_{pp} + r_{qq}}{2} + \frac{1}{2} \sqrt{(r_{pp} - r_{qq})^2 + 4(r_{pq})^2} \sin(\xi - 2\theta) \quad (29)$$

where ξ is determined by

$$\begin{cases} \sin \xi = \frac{r_{pp} - r_{qq}}{\sqrt{(r_{pp} - r_{qq})^2 + 4(r_{pq})^2}} \\ \cos \xi = \frac{2r_{pq}}{\sqrt{(r_{pp} - r_{qq})^2 + 4(r_{pq})^2}} \end{cases} \quad (30)$$

We can minimize (29) by finding θ^* such that

$$\theta^* = \frac{\pi}{4} + \frac{\xi}{2} \quad (31)$$

◆ Case 3: $N < p < q \leq M$

Let

$$\mathbf{J}(\theta, p, q) = \begin{pmatrix} \mathbf{I}_{N \times N} & \mathbf{O}_{N \times (L-N)} \\ \mathbf{O}_{(L-N) \times N} & \mathbf{J}_{(L-N) \times (L-N)}(\theta, \psi, p, q) \end{pmatrix} \quad (32)$$

From (20), we have

$$\mathbf{E}\mathbf{J}(\theta, p, q)\mathbf{G}_k\mathbf{J}^T(\theta, p, q)\mathbf{E}^T = \mathbf{G}_{11}^{(k)} \quad (33)$$

In this case, C_3 is not related to $\mathbf{J}(\theta, p, q)$. Thus the update in (17) and (18) are not necessary.

From the above analysis, we need to determine the transformation $\mathbf{J}(\theta, p, q)$ for

$$p = 1, 2, \dots, N, \quad q = p + 1, \dots, L \quad (34)$$

Our algorithm is an iterative one of which each iteration contains the updates in (17) and (18) for all pairs (p, q) in (34).

4 Simulations

In this example, convulsive blind speech separation based on our method is considered. Furthermore, we compare our algorithm with Para's algorithm [4], which also based on the decorrelation of the output signals in frequency-domain.

In a normal office room, two Speakers saying the numbers from one to 10 in English and Spanish have been recorded speaking simultaneously. The two speech mixtures are available in <http://newton.bme.columbia.edu/~lparra/publish/bss/>. For our method, we set the length of the inverse filter $Q = 50$ and the number of correlation matrices $K = 1000$.

The performance of convulsive BSS is evaluated by the correlation between the output signals. Under the assumption that all estimated signals are zero-mean and unit variance, the following evaluation function is used

$$\varepsilon = \sum_{i=1}^K off(\mathbf{R}_{yy}(\tau_k)) \quad (35)$$

where $\mathbf{R}_{yy}(\tau_k)$ represents the correlation matrix of the output signals in time lag τ_k .

The values of the performance index in (37) for our algorithm and Parra's algorithm are 5.18×10^{-8} and 20.56 respectively.

5 Conclusion

In this paper, we discussed a new method for convolutive blind speech separation. By a whitening process, we used semi-unitary joint diagonalization of a set of the correlation matrices of whitened data lags to recover the speech sources. A novel algorithm for semi-unitary joint diagonalization is proposed by the Jacobi techniques. The performance of our method is demonstrated by a real speech simulation.

References

- [1] Yellin, D., Weinstein, E.: Criteria for multichannel signal separation. *IEEE Trans. Signal Process.* 42(8), 2156–2168 (1994)
- [2] Castella, M., Rhioui, S., Moreau, E., Pesquet, J.-C.: Quadratic Higher Order Criteria for Iterative Blind Separation of a MIMO Convulsive Mixture of Sources. *IEEE Transactions on Signal Processing* 55(1), 218–232 (2007)
- [3] Buchner, H., Aichner, R., Kellermann, W.: A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics. *IEEE Trans. Speech Audio Process.* 13(1), 120–134 (2005)
- [4] Parra, L., Spence, C.: Convulsive blind separation of nonstationary sources. *IEEE Trans. Speech Audio Process.* 8(3), 320–327 (2000)
- [5] Wang, W., Sanei, S., Chambers, J.A.: Penalty function-based joint diagonalization approach for convulsive blind separation of nonstationary sources. *IEEE Transactions on Signal Processing* 53(5), 1654–1669 (2005)
- [6] Rahbar, K., Reilly, J.P.: A Frequency Domain Method for Blind Source Separation of Convulsive Audio Mixtures. *IEEE trans. On Speech and Audio Processing* 5(5), Part 2, 832–844 (2005)
- [7] Cardoso, J.-F., Souloumiac, A.: Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.* 17, 161–164 (1996)
- [8] Vollgraf, R., Obermayer, K.: Quadratic optimization for simultaneous matrix diagonalization. *IEEE Trans. Signal Processing* 54(9), 3270–3278 (2006)

Part VII

**Special Session: Recent Advances in
Brain-Inspired Technologies for Robotics**

Cognitive Representation and Bayesian Model of Spatial Object Contexts for Robot Localization

Chuho Yi, Il Hong Suh, Gi Hyun Lim, Seungdo Jeong, and Byung-Uk Choi

Department of Electronics and Computer Engineering, Hanyang University

17 Haengdang-dong, Sungdong-gu, Seoul, 133-791 Korea

{d1uck, ihsuh, hmetal, sdjeong, buchoi}@hanyang.ac.kr

Abstract. This paper proposes a cognitive representation and Bayesian model for spatial relations among objects that can be constructed with perception data acquired by a single consumer-grade camera. We first suggest a cognitive representation to be shared by humans and robots consisting of perceived objects and their spatial relations. We then develop Bayesian models to support our cognitive representation with which the location of a robot can be estimated sufficiently well to allow the robot to navigate in an indoor environment. Based on extensive localization experiments in an indoor environment, we show that our cognitive representation is valid in the sense that the localization accuracy improves whenever new objects and their spatial relations are detected and instantiated.

1 Introduction

Service robots are required to perform complex tasks with frequent human interaction. Suppose you are visiting a university and ask a staff member how to find Professor Suh's office. The person might say, "Go in the front door of the Information Technology Building and follow the first corridor on your right. You will see the washrooms on your left and classrooms on your right. When you reach the end of the corridor, turn left. At that point, you will see his office on your right." This description would enable you to find the correct office without any difficulty.

Humans do not necessarily require precise quantitative information to perceive space in their current location or to move to another location. Instead, they remember a few landmarks, such as specific structures or distinct objects that delimit the space. They then restructure their knowledge based on spatial contexts and apply that knowledge to the current situation [1]. This method may not metrically show the exact location, but as many pieces of spatial context are accumulated, it enables efficient high-level space recognition and localization.

Robot localization has required building an accurate metric map as well as a semantic map for symbolic inference. Moreover, it has required symbol matching between the metric data and the cognitive representation. An accurate sensor is essential in this complicated process. In the cognitive map proposed by Kuipers et al. [2], the semantic structure of space is inferred using a model based on a

hierarchy of successive environmental information elements obtained while the robot moves around. A global map is derived by integrating global topology information with local metric data.

Many robot localization methods have been developed over the last decade. These can be classified depending on the type of maps they use: grid-based maps [3], feature-based maps [4], topological maps [5], semantic maps [2], and adaptive selection from multiple types of maps [6]. However, these methods of localization and map representation may not be directly applicable to semantic localization.

Humans find locations easily using cognitive information without any metric data. Cognitive representation is also necessary for service robots that interact with humans to take orders and complete tasks. Until now, research has concentrated on robot-centered knowledge that enables a human to interact with robots [7] [8].

Most research works centered on high-level knowledge have generally required identifying objects in a camera image and estimating the distance from the camera to objects to establish relationships among objects [7]. However, these estimates of distance may be inaccurate due to lens distortion and incorrect feature detection and matching, resulting in errors when used in real environments.

To cope with those issues, we suggest a cognitive representation and a Bayesian model for spatial relationships between objects. We show that our representation can be useful in cognitive localization by a mobile robot. The spatial context used in the proposed cognitive representation includes observed objects, a distance context that represents the distance from the robot to a certain object, and a spatial context that describes the relationship among objects. We propose a probabilistic Bayesian model by which localization accuracy can be improved as more elements of the spatial context are accumulated. Finally, we verify the practicality of the proposed methods through a localization experiment in an indoor environment.

2 Spatial Object Relationship

2.1 Object Recognition and Sensing of Object-Related Metrics

Object recognition is a fundamental factor in cognitive representation. In general, an object can be recognized visually by measuring the similarity between its features and those of the corresponding object model. In this section, we use scale-invariant feature transform features that are known to be invariant to image scale and rotation [9] [10].

The metric distance from the robot to an observed object is estimated with a single camera to derive a piece of spatial context. After an object is recognized, its height in an image space is measured using a set of corresponding features. Then, the metric distance is estimated using

$$r_e = \left(\frac{1}{n_r} \right) \sum_{i=1}^{n_r} \left[\left(\frac{h'}{h_i} \right) \cdot c_{r_i} \right] \quad (1)$$

where n_r is the number of spatial contexts for the distance, h' is the height estimated from the corresponding features, and h_i and c_{r_i} are the height and

distance of the spatial context, respectively, which are pre-computed and saved in the object model. Here, i indicates the corresponding context index. The variables h' and h_i are quantities represented by the number of pixels.

Figure 1 illustrates the ratio relationship between a recognized object and its corresponding object model. Since it can be exactly calculated only for an object directly in front of the camera, we assume that the metric distance is given as a Gaussian random variable corresponding to the imperfect alignment of object and camera. This random variable is normally distributed with mean r^e and variance $(\frac{r^e}{c_{qr}})$.

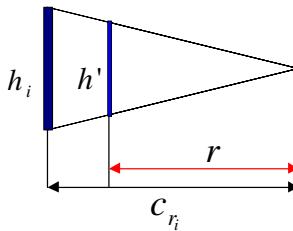


Fig. 1. Relationship of the ratio between the recognized object and corresponding object model

The metric bearing between objects is estimated using the horizontal position of each object and the optical angle of the camera. Previous work [1] has shown that this is simple but inaccurate. To account for this uncertainty, we handle the bearing with a probabilistic model as we did for distance. It is also a Gaussian random variable with mean ω^e and variance $(\frac{\omega^e}{c_{q\omega}})$.

2.2 Cognitive Representation

Figure 2(a) shows a change of metric relations between the robot and observed objects according to the robot's displacement from previous state (location) s_0 to a state in the real world s_1 . The r and ω denote the metric distance of object relative to the robot and the bearing between objects in the robot coordinates, respectively. The subscripts indicate the indexes of observed objects. In general, the metric data quantities are inaccurate. Thus, for cognitive representation, the metric data are linked to the symbol of the spatial relationship appropriate to the given condition.

Figure 2(b) illustrates the cognitive representation consisting of observed objects and their respective spatial symbols. Here, the spatial context includes the distance context and the bearing context. The distance context denoted by c_r is a distance level of object from the robot. Each distance context is represented with one of a set of distance symbols, i.e., $c_r = \{\text{nearby}, \text{near}, \text{far}\}$. The bearing context denoted by c_ω is a bearing of one object relative to another.

Table 1 shows the cognitive representation including all the spatial relationships for the example with symbols. The robot localization scheme we developed

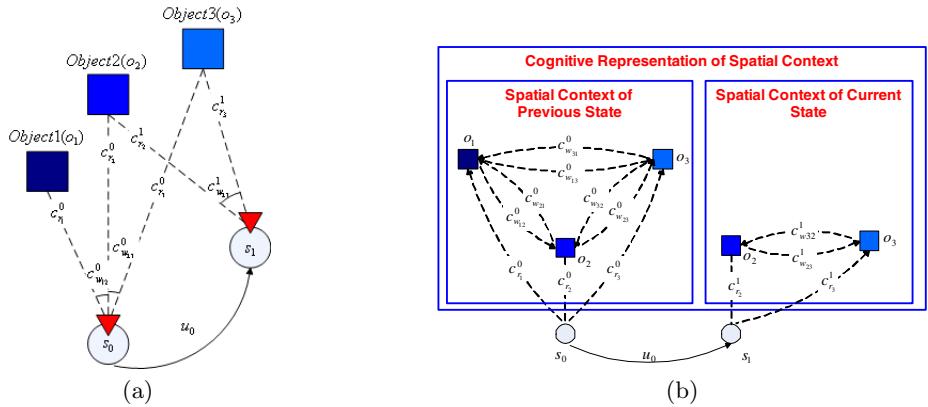


Fig. 2. (a) Spatial relationship between robot and objects in the real world and (b) cognitive representation comprising a set of respective spatial contexts among observed objects and the robot

Table 1. Cognitive representation including all the spatial relationships in Fig. 2

frame	cognitive representation
previous frame	distance(o_1 , nearby), bearing(o_1, o_2 , right near), bearing(o_1, o_3 , right far) distance(o_2 , near), bearing(o_2, o_1 , left near), bearing(o_2, o_3 , right near) distance(o_3 , far), bearing(o_3, o_1 , left near), bearing(o_3, o_2 , left near)
current frame	distance(o_2 , far), bearing(o_2, o_3 , right near) distance(o_3 , near), bearing(o_3, o_2 , left near)

finds the position of the robot using only these kinds of cognitive representation without any metric data.

For symbolic representation, the reference distance may vary according to the performance of the camera, the efficiency of the vision algorithms, and the size of the robot. Due to such distance variations, the symbolic representation should be easily adaptable to diverse applications. That is, symbolic cognitive representation provides us with high-level knowledge to be shared among a number of robot applications [7].

3 Mobile Robot Localization

3.1 Probabilistic Localization Model

We use a Bayesian approach, which partially brings out the probabilistic Bayesian model that was proposed by Ranganathan et al. [2], to estimate the robot location with a cognitive representation for spatial object contexts. The following notation is used to explain the probabilistic model. The robot location and object are denoted by $S = [x \ y \ \theta]^T$ and O , respectively. $C_r = \{c_{r_{1:N}}\}$ denotes the spatial

context of distance, which is an $N \times 1$ matrix. $C_{\omega_{1:N}} = \{c_{r_{\omega_{1:N},1:N}}\}$ is the spatial context of bearing, which is an $N \times N$ matrix. Thus, a set of cognitive representations for spatial object contexts containing distance and bearing is denoted by $C = \{C_r, C_{\omega_{1:N}}\}$, where N is the number of observed objects in the corresponding state. We represent the features extracted from the image with $Z = \{z_{1:n}\}$, where n is the number of features. A set of object model features is denoted by $A = \{a_{1:m}\}$, where m is the number of object features. The numbers of spatial contexts of distance and bearing are n_r and n_q , respectively.

The localization model is described using the cognitive representation. This is the principal focus of this work. Spatial contexts are uncertain data, so they should be approximated with stochastic distributions. The focuses of this section are the location model and the object contexts. We assume that sensors are uncertain, so less context results in a wider distribution. The location posterior is calculated using the cognitive representation described in section 2.

In our research, inaccurate metric data caused a large error in estimating the robot location. We model the localization with a probabilistic approach to overcome this problem.

To estimate the robot location, the localization denoted by $p(S|O, C, Z, A)$ can be factored as

$$p(S|O, C, Z, A) \propto p(O|S, Z, A)p(C|S, O, Z, A)p(S) \quad (2)$$

where $p(O|S, Z, A)$ is the object likelihood that there is a similarity between observed objects in the current state and the ones in the previous state. This is formulated as

$$p(O|S, Z, A) = \exp(-\|O - O_S\|^2) \quad (3)$$

where O and O_S represent observed object in the current state and in the previous state, respectively. The $p(C|S, O, Z, A)$ is the spatial object context likelihood. The $p(s)$ is a prior location that is initialized with a uniform distribution at initial states, and then estimated with the location distribution of the previous state.

The spatial object context likelihood is computed as

$$p(C|S, O, Z, A) = \prod_{j=0}^N \prod_{i=0}^N \left[\exp\left(\frac{-(c_{r_i} - r_i^e)^2}{2\left(\frac{r_i^e}{c_{qr}}\right)^2} \right) \cdot \exp\left(\frac{-(c_{\omega_{ij}} - \omega_{ij}^e)^2}{2\left(\frac{\omega_{ij}^e}{c_{q\omega}}\right)^2} \right) \right] \quad (4)$$

where c_{qr} and $c_{q\omega}$ are the range of the spatial context of distance and bearing, respectively. The denominator terms $\frac{r_i^e}{c_{qr}}$ and $\frac{\omega_{ij}^e}{c_{q\omega}}$ in Eq. (4) are variances used to reflect uncertainty, where r_i^e and ω_{ij}^e are the estimated metric distance and bearing. The further object is from the robot or other objects, the more inaccurate the metric distance and bearing will be. Dividing the spatial object context more finely will improve the localization performance.

3.2 Recursive Bayesian Model

The localization posterior can be calculated with Eq. (2). This is inaccurate because it uses the spatial context containing a high uncertainty. To improve that, we modified the probabilistic Bayesian model to a recursive Bayesian model. However, the modification cannot be applied directly to our Bayesian model because Eq. (4) is not a predictable term. To address this, we modified the location model in Eq. (2) to

$$p(s_k|O_k, C_k, Z_k, A_k) \propto p(O_k|s_k, Z_k, A_k)p(C_k|s_k, O_k, Z_k, A_k) \sum_i \phi_{k-1}^i p(s_k|s_{k-1}^i) \quad (5)$$

where the last term on the right side is an update term. $p(s_k|s_{k-1})$ and ϕ_{k-1} are the state transition and weighted particles, respectively. Spatial objects contexts only vary when motion occurs. Thus, in our research, the state transition takes place in increments based on index k .

Finally, the localization posterior was estimated with the multiplication recursive Bayesian model and object context in Eq. (5). In this work, we used a particle filter that is a kind of recursive Bayesian estimation [13] [14] to manage the complicated computation.

4 Experimental Results

To evaluate the performance of the proposed localization process, we used a Pioneer 3 AT robot carrying a single consumer-grade camera in a 7×6 m indoor environment. The camera captured 252 images as the robot traveled about the test area. Some distinctive objects such as a toy box, a table, a monitor, a drawer and a toy robot were selected for object recognition.

The experimental localization errors are shown in Fig. 3(a). After a quantitative analysis of the results, we learned even though the robot location was initially estimated accurately, errors can increase due to the lack of context obtained during subsequent movement. If the robot observes objects that are far away, the localization error and deviation are relatively high; however, the distribution of robot locations shrinks during subsequent movements as observed objects become closer. If a previously observed object disappears due to robot rotation, the robot deviation increases. In this experiment, minimum and maximum errors were 26.37 and 68.04 centimeters, respectively. The minimum and maximum deviation values were 2.26 and 4.21, respectively.

In Frame A of Fig 3(a), the distribution of robot locations decreased because observed objects were close to the robot. In Frame B, object disappeared due to robot rotation; this caused the distribution to increase. In the case of Frame C, observed objects were far away from the robot. This caused high uncertainty in the spatial context and thus the error of the robot location as well as the distribution increased. However, as shown in Frame D, the error of the robot location and its corresponding distribution decreased again as the robot moved closer to objects.

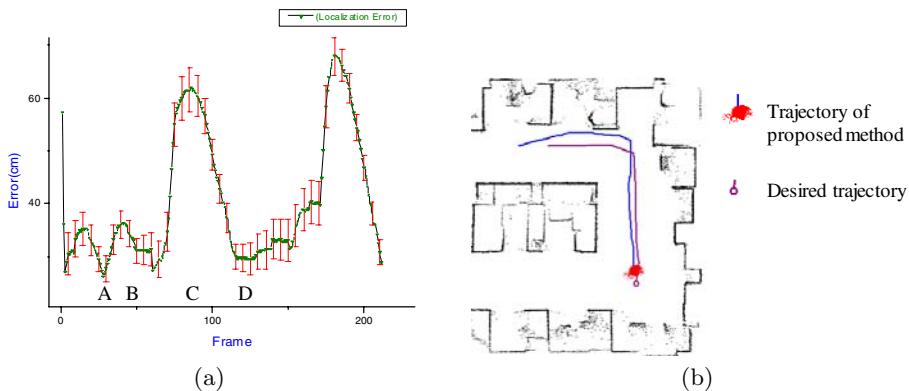


Fig. 3. (a) Localization errors when estimated by the spatial object contexts. (b) Localization results show the trajectory of the robot estimated by the proposed spatial object contexts.

In Fig. 3(b), the circle with line (violet line) indicates the robot trajectory measured using a precise laser sensor. The particles (red dots) represent the posterior distribution of the robot localization.

5 Conclusions

In this paper, we propose that a cognitive representation and Bayesian model of observed objects and their spatial contexts be described by symbols. Our proposed method enabled robots to be localized using spatial object contexts and their probabilistic models. Experimental results from tests of the proposed cognitive robot localization method in an indoor environment can be concluded that as the number of contextual clues is increased, the location accuracy is improved in spite of using inaccurate sensors such as a consumer-grade camera.

References

1. Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavrakij, L.E., Thrun, S.: *Principles of Robot Motion - Theory, Algorithms, and Implementations*. MIT-Press, Cambridge (2005)
2. Kuipers, B.: The Spatial Semantic Hierarchy. *Artificial Intelligence*, 191–233 (2000)
3. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. *IEEE Computer* 22, 44–57 (1989)
4. Miro, V., Zhou, J., Dissanayake, W.: Towards vision based navigation in large indoor environments. In: Proceeding of IEEE the IROS, pp. 2096–2102 (2006)
5. Beeson, P., Kuipers, B.: Towards Autonomous Topological Place Detection Using the Extended Voronoi Graph. In: Proceeding of IEEE ICRA, pp. 4384–4390 (2005)
6. Tully, S., Moon, H., Morales, D., Kantor, G., Choset, H.: Hybrid Localization using the Hierarchical Atlas. In: Proceeding of IEEE the IROS, pp. 2857–2864 (2007)

7. Suh, I.H., Lim, G.H., Hwang, W., Suh, H., Choi, J.H., Park, Y.T.: Ontology-based Multi-layered Robot Knowledge Framework (OMRKF) for Robot Intelligence. In: Proceeding of the IEEE the IROS (2007)
8. Lim, G.H., Suh, I.H.: Weighted Action-coupled Semantic Network (wASN) for Robot Intelligence. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2008)
9. Lowe, D.G.: Distinctive image features from scale invariant keypoints. Int'l. Journal of Computer Vision 60(2), 91–110 (2004)
10. Ke, Y., Sukthankar, R.: PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In: Proceedings of Computer Vision and Pattern Recognition (2004)
11. Yim, B.D., Hwang, S.Y., Song, J.B.: Mobile Robot Localization based on Fusion of Vision and Range Information. In: CASS 2006 (2006)
12. Ranganathan, A., Dellaert, F.: Semantic Modeling of Places using Object. In: Robotics: Science and Systems (RSS) (2007)
13. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
14. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte Carlo Localization for Mobile Robot. In: Proceeding of the ICRA (1999)

Learning of Action Generation from Raw Camera Images in a Real-World-Like Environment by Simple Coupling of Reinforcement Learning and a Neural Network

Katsunari Shibata and Tomohiko Kawano

Oita University, 700 Dannoharu, Oita, Japan

shibata@cc.oita-u.ac.jp

<http://shws.cc.oita-u.ac.jp/~shibata/home.html>

Abstract. For the development of human-like intelligent robots, we have asserted the significance to introduce a general and autonomous learning system in which one neural network simply connects from sensors to actuators, and which is trained by reinforcement learning. However, it has not been believed yet that such a simple learning system actually works in the real world. In this paper, we show that without giving any prior knowledge about image processing or task, a robot could learn to approach and kiss another robot appropriately from the inputs of 6240 color visual signals in a real-world-like environment where light conditions, backgrounds, and the orientations of and distances to the target robot varied. Hidden representations that seem useful to detect the target were found. We position this work as the first step towards taking applications of the simple learning system away from “toy problems”.

1 Introduction

In order to develop human-like intelligent robots, researchers have been trying to introduce sophisticated human functions into them. They have modularized the whole process into some functional modules such as recognition, action planning, and control at first, then developed each module individually, and finally connected them sequentially. However, the brain is massively parallel and cohesively flexible, and much of the brain functions seem to be performed subconsciously. Therefore, we think there is a limitation for humans to know exactly how the brain really works by the guess of the functions through the sequential consciousness. For the optimization and consistency of the entire system, we think that even though the understandability for humans is sacrificed, we should try not to interfere to the robot, but to leave everything to the robot’s optimization through autonomous learning. From these discussions, we have suggested a general and autonomous learning system in which a neural network (NN) simply connects from sensors to actuators in a robot and is trained by reinforcement learning (RL) as shown in Fig. 1 [1, 2].

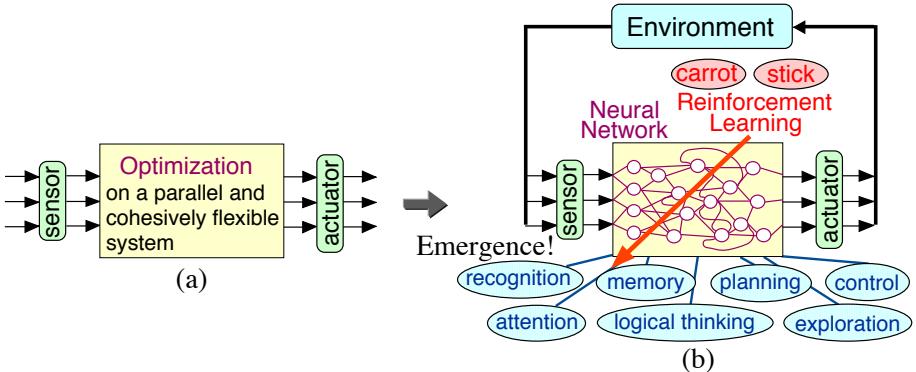


Fig. 1. (a) The objective in robots. (b) Parallel and cohesively flexible learning of the entire process from sensors to actuators by coupling of RL and NN.

In our previous work [3], acquisition of flexible image recognition by coupling of RL and a NN was shown, but the robot was static at the same place and could move its head to one of only nine discrete states. This means that the target recognition could be achieved easily by a “template matching” technique, and actually, in the hidden layers, many neurons seemed to work as a “template”. In this experiment, since one robot was placed at a different initial location at each trial (episode) and walked, the number of possible views of the other robot could not be counted. The robot has to find the other robot in the camera images under various conditions as shown in Fig. 4, and has to approach and kiss it. One can understand that it is not so easy to develop appropriate programs for this task in the real-world-like environment. To learn a task with such a huge state space, the point is whether effective abstraction could be obtained autonomously through learning. We position this work as the first step towards taking applications of the simple learning system away from “toy problems”.

2 Experiment

Here an experiment using two AIBO robots, a white one and a black one, is introduced. The two AIBOs are placed in a $1.5m \times 1.6m$ field. The mission of the black AIBO is to approach the white AIBO and kiss it as shown in Fig. 2. Before each episode, the black AIBO is located manually within the range of about ± 45 degrees from the front line of the white AIBO and on the condition that the white AIBO is caught in the camera of the black AIBO. The black AIBO captures a color image with its camera at its nose after moving every step. The number of pixels is around 350,000, but when the AIBO sends the image to the computer, it is reduced to $52 \times 40 = 2080$ by pixel skipping. The white AIBO sits on a transparent plastic bottle, and is fixed at the same place during one episode. However, the location of the white AIBO was sometimes changed between episodes.

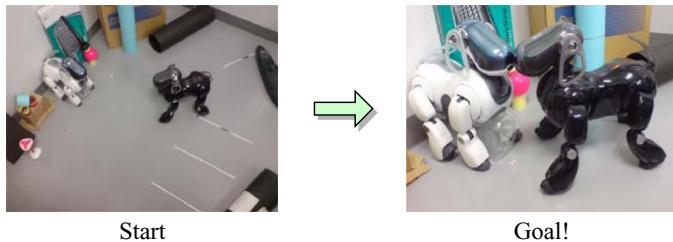


Fig. 2. The mission of the black AIBO is to walk and kiss the white AIBO

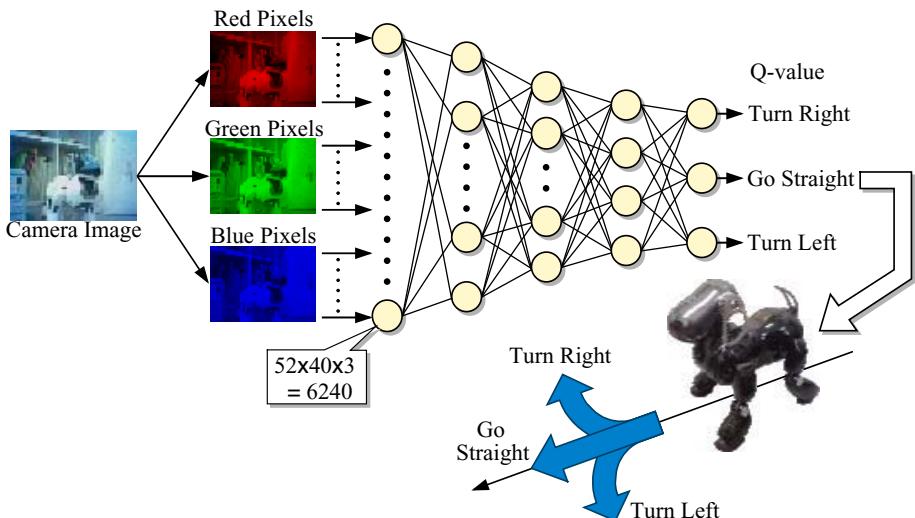


Fig. 3. The processing in the black AIBO. The visual signals are inputted to the NN directly and the black AIBO chooses its action according to the outputs from three possible actions, “turn right”, “turn left” and “go forward”.

The processing in the black AIBO is shown in Fig. 3. The 52×40 image is inputted into the 5-layer NN after inverting and normalizing each pixel value between 0.0 and 1.0. Since each pixel is represented by the gradation of RGB colors, the number of inputs to the NN is 6240 in total. The number of the output neurons is three in accordance with the number of actions that the black AIBO can choose. The three actions are “turn right”, “turn left” and “go forward”. The output is dealt with as the Q -value for the corresponding action, and ϵ -greedy ($\epsilon = 0.13$ here) is employed as the action selection [4]. The actions are based on the software provided by Ito and Kakiuchi [5]. When the robot chooses the action “turn”, the turning radius and angle are 22.5cm and 9.5° respectively. When choosing the action “go forward”, it advances by 6cm. However, the angles and distances actually vary at each time, since it is a real robot.

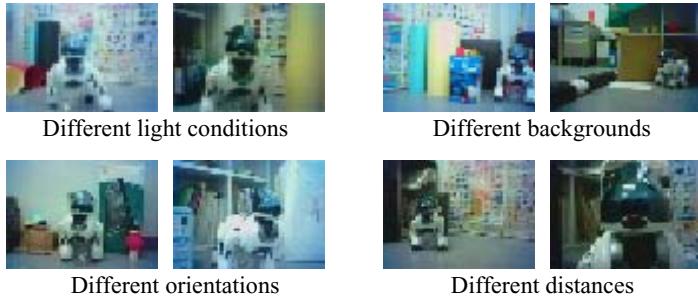


Fig. 4. Some sample images captured under different light conditions with different backgrounds, different orientations of and different distances to the white AIBO. They were used actually in learning.

An experimenter always monitored the situation and the camera image during the experiment. When it judged that the black AIBO kissed the white AIBO, a reward was given to the black AIBO. When it judged that the black AIBO lost sight of the white AIBO, a penalty was given. In both cases, the episode terminated. For acceleration of learning, some small rewards or penalties were imposed depending on the situation without termination of the episode, but they were not imposed after 2000 episodes of learning.

In order to show the variety of the captured images depending on the light condition, background, and orientation of and distance to the white AIBO, Fig. 4 shows some samples. The light conditions change according to the hour when the experiment was done. The location of the white AIBO and also the backgrounds were changed by the experimenter. The black AIBO not only approach from the front of the white AIBO, but also was located diagonally to the front of the white AIBO as an initial state though it was not located at its side or back.

As for the NN, the number of neurons in each layer is 6240-600-150-40-3 from input layer to output layer. The training signal is generated using Q-learning [6] and the NN is trained by Error Back Propagation [7]. When one of the actions is chosen at time t , the robot walked, and forward computation of NN at time $t+1$ is performed with the new camera image S_{t+1} captured after the walk. After that, the previous input S_t are inputted into the NN again, and the training signal is provided only to the output corresponding to the executed action a_t at time t according to

$$T_{a_t, t} = r_{t+1} + \gamma \max_a O_a(S_{t+1}) \quad (1)$$

where γ is a discount factor (here 0.9 is used), $O_a(S_t)$ is the a -th output of the NN when the captured image at time t are entered as sensor signal inputs. Here, the sigmoid function used as the output function of each neuron ranges from -0.5 to 0.5. To adjust the offset between Q -value and NN output, a linear transformation is done. Here, since Q -value 0.0 corresponds to output -0.4 and Q -value 0.8 corresponds to output 0.4, 0.4 is added or subtracted actually for the transformation. When the episode terminates, the second term of right-hand

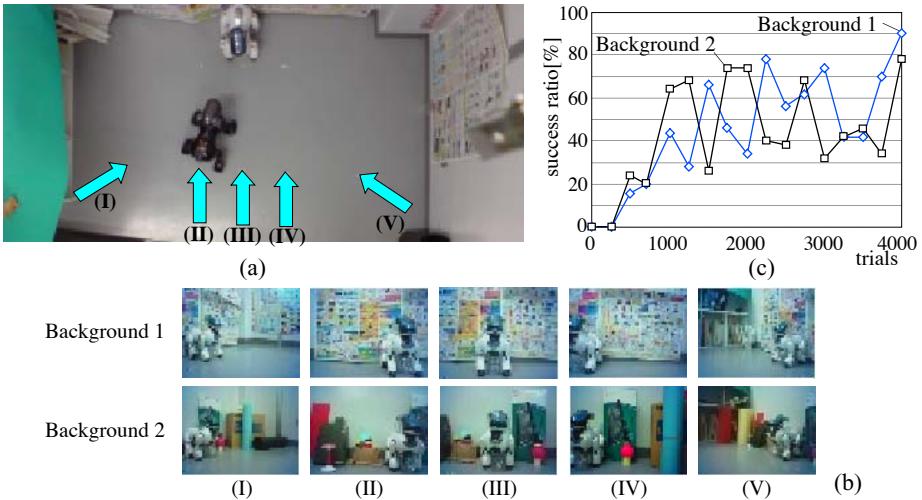


Fig. 5. (a) The environment for the test trials and (b) the camera images at the five initial locations from (I) to (V) for two kinds of backgrounds. (c) The learning curve shows the change of success ratio according to the number of trials.

side of Eq. (II) is set to be 0. The reward for the kiss is 0.8, while the training signal for the missing is equal to the Q -value output minus 0.02. The reward and penalty at non-terminal state is 0.04 and -0.04 respectively.

Since each episode takes a long time, simulation was also run unsupervised. All the visual signals and chosen actions were stored for every episode during the experiment, and in the simulation, the NN was trained by using the stored data up to that time. However, since actions could not be chosen, the same episodes appear. The simulation was run 13 times during the course of 4000 experimental episodes. In each simulation, 10000 episodes of learning were executed.

Fig. 5(c) shows the learning curve. All the connection weights in the NN are stored every 250 episodes of learning. Two backgrounds were prepared for the test, and five initial locations were selected as shown in Fig. 5(a). A sample image from each initial state is also shown in Fig. 5(b). Five trials were performed for each background and initial location set with the greedy action selection, and the success ratio for each background was observed as a learning curve. Using the connection weight set before learning, it could not kiss the white AIBO from any initial positions in either background. The ratio increased gradually at first, and did not increase very much after around 1000 trials. However, eventually, the success ratio reached between 78% to 90% even though the black AIBO was not exposed to the same background in the learning phase.

In order to examine how learning progressed, the time-series of Q -values are observed. Fig. 6 shows the change of Q -values for the two cases of using the connection weight set (a) after 1000 trials and (b)(c) after 4000 trials. If learning converged successfully, it is expected that the maximum Q -values at each step

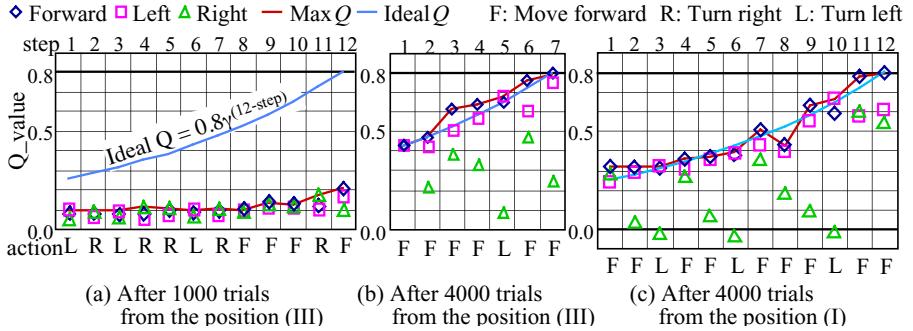


Fig. 6. Some samples of Q -value change in one trial. The weight sets after 1000 trials are used in the case of (a), and that after 4000 trials are used in the case of (b) and (c). The starting location is different between (b) and (c), and refer Fig. 5 for it.

would increase as the ideal Q -value that is the exponential curve decided from the discount factor γ and the final reward 0.8. It can be seen by comparing Fig. 6(a) and (b) that in the case of after 1000 trials, the episode length is larger than the case of after 4000 trials. The series of actions chosen can be seen at the bottom of each figure. The black AIBO chose the “turn left” action even in the case (b) of the front start after 4000 trials. The reason for this is that the black AIBO is likely to turn right slightly even when it chooses the action “go forward”, and so compensated the shift. Furthermore, in the case (a) of after 1000 trials, the Q values are very different from the ideal, but in the cases (b) and (c) of after 4000 trials, the maximum Q -value is very similar to the ideal. From this result, it can be inferred that learning actually progressed even though the success ratio did not seem to change pronouncedly after 1000 trials.

Next, the role of the hidden neuron is guessed from the connection weights. Because the number of connections in each lowest hidden neuron is the same as the number of inputs, the weight pattern after a linear transformation can be observed as a color image whose size is 52×40 . The weight patterns seem random due to the influence of the initial connection weight that were determined randomly. However, revealing patterns could be found when the change of each weight from the initial value was observed. The linear transformation from each weight value to the corresponding pixel value is as

$$\text{pixel}_{\text{color},i,j} = \frac{w_{\text{after,color},i,j} - w_{\text{before,color},i,j}}{\max_{\text{color},i,j} |w_{\text{after,color},i,j} - w_{\text{before,color},i,j}|} \times 127 + 128, \quad (2)$$

where w_{after} , w_{before} indicate the weight after and before learning respectively. The color can be R, G, or B, and i, j indicates the row and column number of a pixel in the image respectively.

Most of the weight images are very vague, but in some of them, the image of the white AIBO is suggested as in Fig. 7(a-1,2). The upper images of Fig. 7(a-1,2) are similar camera images selected from actually captured ones. The representations look different from those observed in our previous experiment [3].

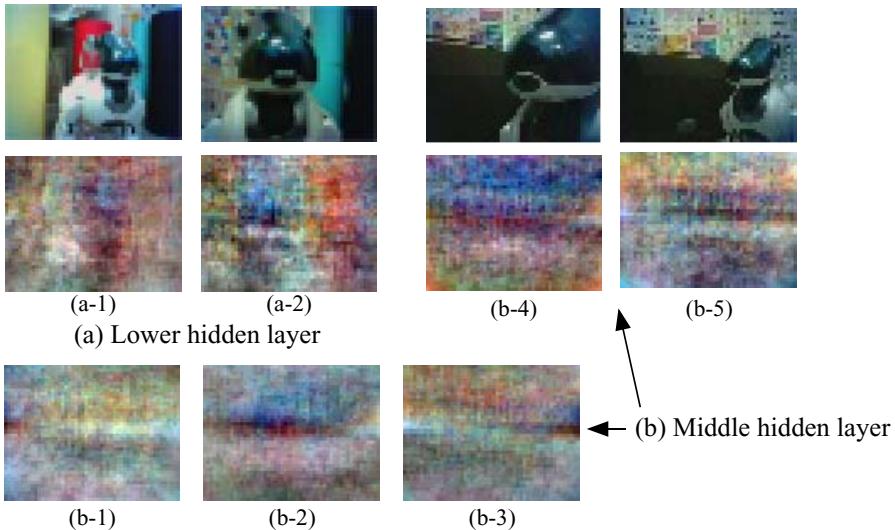


Fig. 7. Visualization of the hidden neurons' role generated from the connection weights change from the initial values. The images for the lowest hidden neurons are generated by Eq. (2), and those for the middle hidden neurons are generated as the weighted sum of the images for the lowest hidden neurons. The four images at the top are actual images that seem relevant to the weight images.

in which a clearer AIBO image could be seen. In the experiment, the black AIBO was fixed and only the head was rotated. Furthermore, the head angle could be one of only 9 states. This means that the number of views of the white AIBO is limited, which was enough for each hidden neuron to represent only one of the views of the target AIBO. However, in this experiment, the number of hidden neurons is not sufficient for each neuron to represent only one view because the number of views is far larger than the number of hidden neurons.

In order to roughly see the role of each neuron in the middle hidden layer, the weighted sum of the weight changes of the lowest hidden neurons by the connection weights between the lowest and middle hidden layers is observed. The word “roughly” is used because the non-linear factor in the lowest hidden layer neurons is neglected. Fig. 7(b-1, . . . , 5) shows the visualized weight image of some neurons in the middle hidden layer after normalization from 0 to 255. In most of the images, the dark or bright thin band-like area spread laterally in the middle section of the image is clearly visible in Fig. 7(b-1, . . . , 5). It can be inferred that neurons such as (b-1, 2, 3) contribute to detect the lateral location of the target AIBO. In most of them, by the lateral extension of the band-like area, the color changes from dark to bright or from bright to dark at the same height. It is inferred that this emphasizes the contrast in the neuron's output according to the lateral location of the target AIBO. When comparing with the weight images of (b-4, 5), the dark blue area is spread wider to the upper area of the image in (b-4), while the dark area is thin in (b-5). Referring from the actual

camera images above these weight images, it can be inferred that such neurons contribute to detect the distance to the target AIBO. The representations also look different from those observed in our previous experiment in which fat AIBO images often appeared in the images for the neurons in the middle hidden layer.

However, the detection of the lateral or forward distance is only a part of the functions that the NN acquired. Neurons in this experiment must acquire a variety of functions such as compensation of light conditions, neglect of background and some other functions that we were not aware of from the connection weights easily. This is very similar to the situation where we cannot understand exactly how the brain functions when we see the excitation pattern of real neurons in the brain.

3 Conclusion

It was shown that although our proposed learning system that consists of one neural network trained by reinforcement learning is very simple, it worked on a real walking robot in a real-world-like environment. Many hidden neurons seem to obtain useful feature extraction abilities through learning without any suggestions from the others. The image processing acquired in this task seems different from the template-like image processing that was found in our previous work in which the robot did not walk and the head could rotate to one of only 9 states. This difference shows the flexible and purposive function emergence in a neural network by our proposed approach.

Acknowledgments. This research was supported by JSPS Grant in-Aid for Scientific Research #19300070. We would like to express our gratitude to Dr. Ito and Mr. Kakiuchi in Naruto University of Education for permitting us to employ their software.

References

1. Shibata, K., Okabe, Y.: Reinforcement Learning When the Visual Signals are Directly Given as Inputs. In: Proc. of ICNN 1997, vol. 3, pp. 1716–1720 (1997)
2. Shibata, K., Iida, M.: Acquisition of Box Pushing by Direct-Vision-Based Reinforcement Learning. In: Proc. of SICE Annual Conf. 2003, 0324.pdf, pp. 1378–1383 (2003)
3. Shibata, K., Kawano, T.: Acquisition of Flexible Image Recognition by Coupling of Reinforcement Learning and a Neural Network. The SICE Journal of Control, Measurement, and System Integration 2(2) (to appear, 2009)
4. Sutton, R.S., Barto, A.: Reinforcement Learning: An Introduction, A Bradford Book. MIT Press, Cambridge (1998)
5. Ito, Y., Kakiuchi, H., Oizumi, K., Kikuchi, A.: Development of Educational Software to Control Walking Motion for Four-legged Robot. The Japanese Society of Technology Education 49(1), 1–9 (2007)
6. Watkins, C.J.C.H.: Learning from Delayed Rewards, PhD thesis, Cambridge University, Cambridge, England (1989)
7. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representation by Error Propagation. In: Parallel Distributed Processing, vol. 1, pp. 318–364. MIT Press, Cambridge (1986)

Brain-Inspired Emergence of Behaviors Based on the Desire for Existence by Reinforcement Learning

Mikio Morita and Masumi Ishikawa

Department of Brain Science and Engineering, Kyushu Institute of Technology
2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan
ishikawa@brain.kyutech.ac.jp

Abstract. To develop truly autonomous mobile robots, we proposed to introduce internal rewards such as the desire for existence, specific curiosity, diversive curiosity, boredom, and novelty into reinforcement learning. They are expected to make mobile robots capable of behaving appropriately without being told what to do. Firstly, we proposed to use multiple sources of rewards to endow mobile robots with ability to behave properly in the real world. Secondly, we proposed task-independent internal rewards. Thirdly, we proposed to attain engineering merit of internal rewards in addition to scientific interest. A pursuit-evasion game comprising a predator and its prey on a robotic field was selected as a testbed to demonstrate the effectiveness of internal rewards in reinforcement learning. The present paper focuses on learning of pursuit timing to maximize accumulated future rewards by Q-learning and SARSA.

1 Introduction

We aim at developing truly autonomous mobile robots capable of behaving appropriately without being told what to do not only at a concrete level but also at an abstract level. This is what mammals do in their daily lives. We believe a developmental approach is effective to realize this. The desire for existence, specific curiosity, diversive curiosity, boredom, and novelty are important factors contributing to learning and development.

Schmidhuber proposed two curiosity principles in response to prediction errors and predictor improvements [1]. Oudeyer et al. proposed architecture based on the second curiosity principle [2]. Stout et al. used both external and internal rewards, but the latter may be interpreted as a subgoal provided by a designer to accelerate learning [3]. If internal rewards are task dependent and are provided by a designer, mobile robots with the internal rewards can no longer claim to be autonomous.

We proposed the following principles, which we believe are important for useful and effective intelligent systems [4][5].

Firstly, it is important to use multiple sources of rewards to endow mobile robots with ability to behave properly in the real world. This is because, we think, external rewards and curiosity rewards alone are not enough for mobile robots to behave properly in the complex and context-dependent real world.

Secondly, it is important to use task-independent internal rewards, because task-dependent ones provided by a designer are not applicable to other tasks as they are, and do not contribute much to genuine autonomy.

Thirdly, it is important to attain engineering merit of internal rewards, in addition to scientific interest.

These principles share one characteristic, i.e., use of reinforcement learning and internal rewards. The reason for this is that reward based learning with trial and error is, we believe, essential in developing intelligent behaviors. On the other hand, hand-crafted programs tend to impose a burden on humans, and supervised learning and self-organizing learning are considered not to be suitable for this.

A pursuit-evasion game comprising a predator and its prey on a robotic field was selected as a testbed. We proposed mathematical models for internal rewards to demonstrate their effectiveness in reinforcement learning. The present paper focuses on learning of pursuit timing to maximize accumulated future rewards by Q-learning and SARSA.

2 Overview of a Task

To evaluate the utility and benefit of internal rewards, we use a predator and its prey on a robotic field in Fig. 1 as a testbed. A task, here, is that the predator learns to capture the prey for its survival, and the prey also learns to escape from the predator for its survival. The desire for existence is represented by an intrinsic effort to keep a battery of an agent from exhaustion. They also learn a map of the environment to make pursuit or evasion more efficient. The following is a prospective task scenario of agents.

When the level of its battery is sufficiently high, an agent efficiently learns the environment motivated by curiosity, boredom and novelty. Under this condition, curiosity, either specific or diversive, motivates the agent to learn the environment through a desire to increase accumulated future rewards. When some part of the environment is sufficiently learned, the agent gets bored and goes away by a negative reward caused by boredom, which provides a stronger motivation than diversive curiosity. When a novel object is observed, novelty motivates the agent to approach and closely observe it, which in turn motivates specific curiosity.

On the other hand, when the level of its battery becomes lower, the predator starts to pursue its prey for its survival by taking advantage of the knowledge acquired through curiosity and novelty.

We carry out simulation experiments on the predator and the prey task to demonstrate the utility and benefit of introducing the desire for existence, specific curiosity, diversive curiosity, boredom, and novelty as internal rewards into reinforcement learning. Skill acquired through learning by simulation is implemented on real mobile robots, WITHs¹.

¹ The mobile robot, WITH, is a omnidirectional robot with 8 IR sensors and 3 motors. IR sensors are arranged radially at equal angles. Motors are also arranged radially at equal angles of 120 degrees, and omni wheels are installed by them.

This paper focuses on emergence of behaviors based on the desire for existence, particularly on learning of pursuit timing to maximize accumulated future rewards, because early capture decreases the amount of reward. This part has not yet been implemented on WITH.

3 Formulation of Internal Rewards

This section presents brief overview of reinforcement learning, and mathematical models for the desire for existence.

3.1 Overview of Reinforcement Learning

Learning is done by Q-learning popular in reinforcement learning. It maximizes accumulated future rewards by trial and error under delayed and intermittent rewards. External reward is given only at capture, and internal rewards are given with higher frequency. An episode here terminates when the predator captures the prey or when the maximum number of steps is reached.

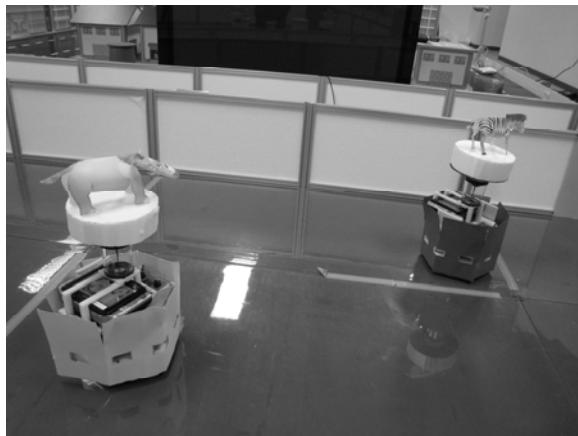


Fig. 1. Robotic field with the predator (lion) and the prey (zebra) using mobile robots, WITH

Q-learning is used for learning by both the predator and the prey. Action-value function, $Q(s,a)$, represents accumulated future rewards starting from the state, s , and taking the action, a . It is updated as,

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_a Q(s',a') - Q(s,a)] \quad (1)$$

where α ($=0.3$) is a learning rate, r ($=2.0$) is a reward from the environment at capturing the prey, and γ ($=0.99$) is a discount rate. For each action, the penalty of -0.005 is given from the environment. An epsilon-greedy algorithm is adopted to promote exploration, which may be interpreted as diversive curiosity.

3.2 Desire for Existence

The speed of the predator is given by,

$$v_{\max}(b) = k(1 - e^{-pb}) \quad (2)$$

where p is a speed preservation rate, k is a speed rate, and $v_{\max}(b)$ is the maximum speed of an agent at a battery level, b . The level of a battery at time t is defined by,

$$b(t) = b(t-1) - m_m v(t)^2 - m_b \quad (3)$$

where $v(t)$ is a speed at t , m_m is a movement metabolic rate, and m_b is a basal metabolic rate.

In a preliminary study, i.e., cases 1 to 3 in Section 4, only a simple constant external reward is provided. In a realistic situation, the predator must learn when to start pursuing of the prey for its survival. This is because if the predator captures the prey too early, i.e., when the battery level is significantly high, the external reward becomes less, hence the accumulated future rewards becomes less. On the other hand, if the predator starts to pursue too late, it can no longer capture the prey due to exhaustion of its battery. Therefore, this timing is an important target of learning. To realize this, the following monotonically decreasing reward is given,

$$r(b) = e^{2(1-b)} - 4 \quad (4)$$

where b is a battery level ranging from 0 to 1, and $r(b)$ is the amount of reward at capture when the battery level is b . When $b > 1 - \ln(2)$, a negative reward is given.

We consider that the learning of pursuit timing is not easy, because explorative actions such as epsilon-greedy tend to capture the prey when the predator is close to the prey. To prevent this, we try SARSA in addition to Q-learning [6]. As is well known, SARSA is an on-policy method and Q-learning is an off-policy method. SARSA is considered to be effective in preventing early capture due to explorative actions. Furthermore, we try progressive SARSA and Q-learning in addition to ordinary ones. By “progressive” we mean that easy samples are provided in the beginning stage, and progressively difficult samples are provided at later stages to make reinforcement learning easier.

3.3 Definition of States and Actions

As is well known, Q-learning has a difficulty of large computational cost. We propose to decrease the number of states by various approximations. The state here is defined by the battery level (20-level) and the distance to the prey (20-level), without absolute coordinates. When there are obstacles, however, they are not enough to define the state, and the distance to an obstacle in the forward direction (10-level) is added to the definition of state.

The speed of the agent (10-level) constitutes action alternatives. The ego-centric orientation of motion (5-level) is added to the definition of the action in the presence of obstacles.

4 Experimental Results

Case 1: Only the predator learns and the prey escapes simply in the opposite direction. Figure 2 illustrates the number of steps needed for the predator to capture the prey in the absence of obstacles. As learning proceeds, the predator captures its prey in about 150 steps. Fig. 3 depicts the resulting Q-values as a function of the distance to the prey and the battery level.

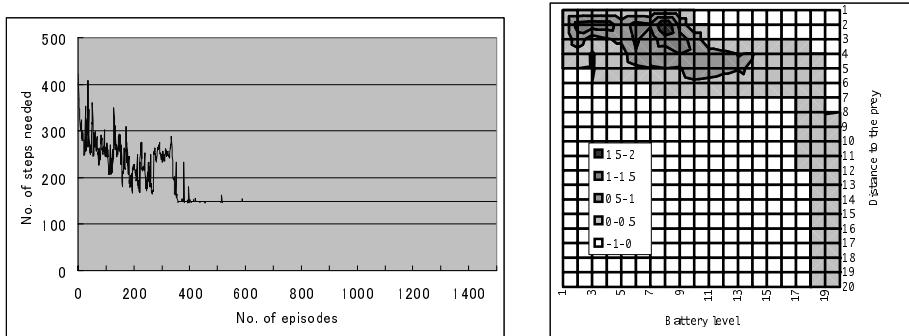


Fig. 2. (a) The number of steps needed to capture the prey in the absence of obstacles, (b) The resulting Q-values (Case 1)

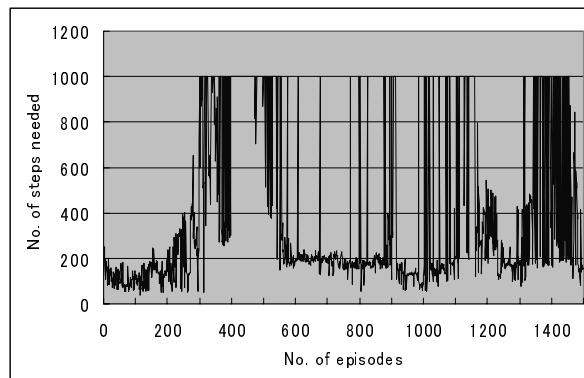


Fig. 3. The number of steps needed to capture the prey in the absence of obstacles (Case 2)

Case 2: Both the predator and the prey learn simultaneously. Fig. 3 illustrates the number of steps needed to capture the prey. Due to insufficient learning, the number of steps needed is not stable; even during the last 500 episodes, the predator captures the prey in about 3/4 of episodes.

Case 3: Only the predator learns and the prey flees in the opposite direction in the robotic field with obstacles in Fig. 4. A simple obstacle avoidance mechanism is implemented in both the predator and the prey. Final Q-values without obstacles (Case 1) are used as initial Q-values to stabilize and accelerate learning. Starting from zero

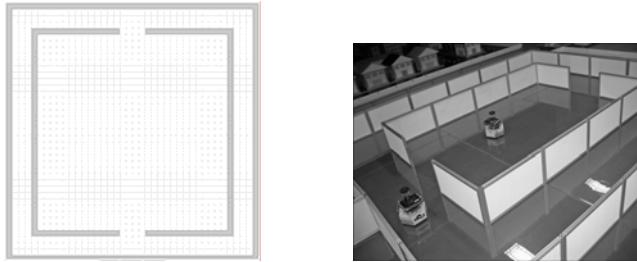


Fig. 4. The robotic field with obstacles

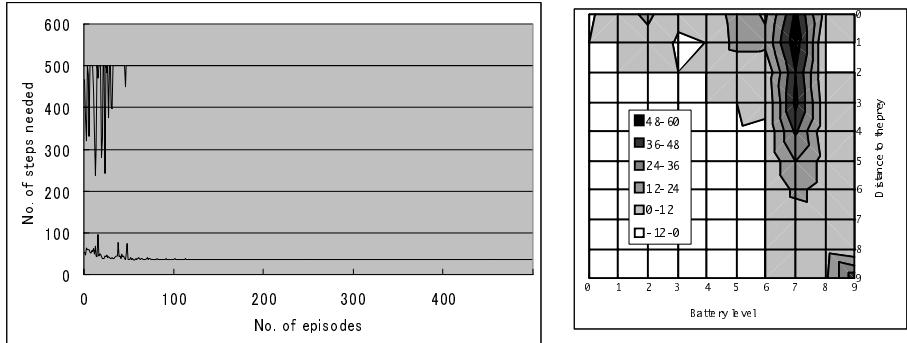


Fig. 5. (a) The number of steps needed to capture the prey in the presence of obstacles. (Case 3) The upper curve shows the results with zero initial Q-values, indicating the inability to learn from scratch. The lower curve shows the results with the final Q-values in Case 1 used as initial Q-values. (b) The resulting Q-values as a function of the distance to the prey and the battery level.

Q-values, the predator cannot capture the prey even after learning. Fig. 5(a) illustrates the number of steps needed for the predator to capture its prey. Fig. 5(b) illustrates the resulting Q-values without learning of pursuit timing.

Case 4: Learning of pursuit timing by Q-learning and SARSA. To make the learning of pursuit timing easier, speed is assumed to be binary here: moves at a constant speed or stops. It is also assumed that the prey is standing still. These assumptions are adopted to clarify the characteristics of learning of pursuit timing.

Fig. 6 (a) illustrates that Q-values have a peak at the north-west corner, i.e., both the distance and the level of a battery are zero after 90000 epochs. This figure are not enough for evaluating the learning performance.

To do rigorous evaluation, we calculated the average battery level at capture and average reward by greedy action based on the resulting Q-values starting randomly from the state space with almost full battery charge.

Table 1 summarizes the learning performance. Fig. 6(b) illustrates the path of the predator in the state space. We anticipated that SARSA is superior to Q-learning, and progressive learning is superior to ordinary learning. Table 1 indicates that, contrary to our expectation, ordinary Q-learning is the best, and progressive Q-learning is the second best.

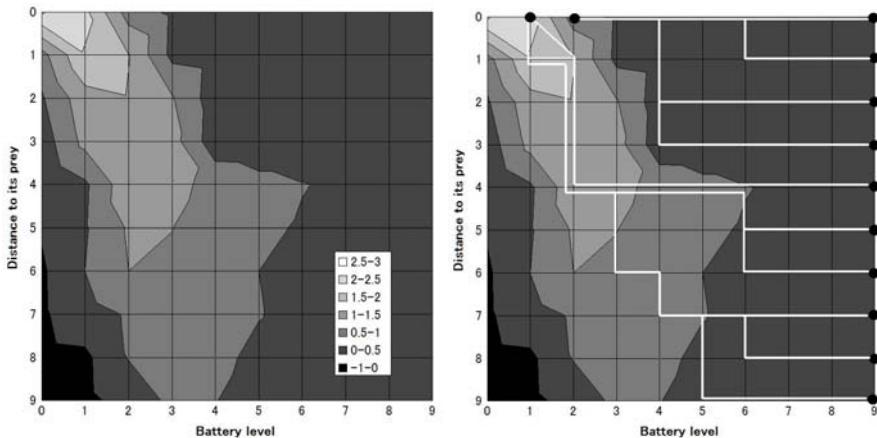


Fig. 6 (a) Resulting Q-values by Q-learning of pursuit timing. (b) Path of predator's state.

Table 1. Summary of learning performance of pursuit timing

learning method	battery level at capture	average reward
Q-learning	0.215	0.845
progressive Q-learning	0.219	0.797
SARSA	0.238	0.593
progressive SARSA	0.264	0.531

5 Conclusions and Discussions

In the present paper, we focused on the desire for existence, particularly on learning of pursuit timing. We found that ordinary Q-learning is superior to progressive Q-learning and SARSA and progressive SARSA. This is contrary to our expectation, and its clarification is left for immediate future study. Furthermore, the performance was not so good as we had expected, because a predator captured its prey when the battery level was far from exhaustion. In other words the phenomenon of early capture was frequently observed.

We discuss here the cause and remedy of the early capture. Due to quantization of the state space, it often happens that the state does not change over time. When an agent chooses an action of “standing still,” this lasts for a long time, which gradually decreases its Q-value due to discount factor in reinforcement learning and to the basal metabolism. This finally makes an action of “movement” to be taken, which causes undesirable early capture. We come up with 3 alternative ways to ameliorate the early capture.

- 1) Set the value of the discount rate to 1. This makes the decrease of Q-values much slower, because Q-values decrease due only to the basal metabolism.
- 2) Modify Q-values only when the state changes. This prevents the decrease of Q-values, but the basal metabolism is not taken into account. It is also dubious that the optimality of reinforcement learning is preserved in this case.

- 3) Use finer grained state space. This ameliorates the early capture with sacrifice of computational cost.

Experiments on amelioration of the early capture are left for further study.

What was done, however, is still a small step towards truly autonomous mobile robots. Whether emergence of behaviors based on internal rewards really leads to better performance in the long term has not yet been evaluated. In this paper, it was assumed that the predator is faster than its prey. If this is not the case, cooperative behaviors among predators become necessary. This is also left for a challenging research.

Acknowledgments. This research was partially supported by a COE program (#J19) granted to Kyushu Institute of Technology by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan. It was also supported by Grant-in-Aid for Scientific Research(C)(18500175) from MEXT, Japan.

References

1. Schmidhuber, J.: Self-motivated development through rewards for predictor errors/ improvements, Developmental Robotics. In: 2005 AAAI Spring Symposium (2005)
2. Oudeyer, P.-Y., Kaplan, F., Hafner, V.V.: Intrinsic Motivation Systems for Autonomous Mental Development. IEEE Trans. EC 11(2), 265–286 (2007)
3. Stout, A., Konidaris, G.D., Barto, A.G.: Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning. In: Developmental Robotics AAAI Spring Symp. (2005)
4. Hagiwara, T., Ishikawa, M.: Emergence of behaviors based on the desire for existence. In: BrainIT 2007, p. 41 (2007)
5. Ishikawa, M., Hagiwara, T., Yamamoto, N., Kiriakie, F.: Brain-inspired emergence of behaviors in mobile robots by reinforcement learning with internal rewards. In: HIS 2008, pp. 138–143 (2008)
6. Sutton, R.S., Barto, A.G.: Reinforcement Learning. MIT Press, Cambridge (1998)

A Neural Network Based Controller for an Outdoor Mobile Robot

Masanori Sato¹, Atsushi Kanda², and Kazuo Ishii²

¹ Fukuoka Industry, Science & Technology Foundation,
3-8-33 Momochihama, Sawara-ku, Fukuoka 814-0001, Japan

² Kyushu Institute of Technology,
2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan
sato@lab-ist.jp, kanda-atushi@edu.brain.kyutech.ac.jp,
ishii@brain.kyutech.ac.jp

Abstract. A wheeled mobile mechanism with a passive and/or active linkage mechanism for travel in rough terrain is developed and evaluated. In our previous research, we developed a switching controller system for wheeled mobile robots in outdoor environment. This system consists of two sub-systems: an environment recognition system using a self-organizing map and an adjusted control system using a neural network. In this paper, we propose a new controller design method based on a neural network. The proposed method involves three kinds of controllers: an elementary controller, adjusted controllers, and simplified controllers. In the experiments, our proposed method results in less oscillatory motion in outdoor environment and performs better than a well tuned PID controller does.

1 Introduction

This work concerns the important issues for mobile robots, controller design for effective mobility. Recently, various mobile mechanisms have been developed that combine wheels and passive and/or active linkage mechanisms. Compared with other mobile systems (e.g. legged robots), wheeled systems have advantages of high energy efficiency, simple mechanics and well-investigated control systems.

On the other hand, such vehicles are generally only able to climb over bumps with a height of 1/2 or less of that of the wheel diameter without inertial forces. Thus wheeled mobile robots have difficulties traveling on rough terrain. Various wheeled robots with high mobility in rough terrain have been proposed; for example, the NASA/JPL developed the Rocker-Bogie mechanism installed in Sojourner [1], Ku-rod़a et al. developed the PEGSUS mechanism installed in Micro 5 [2], the Swiss Federal Institute of Technology in Lausanne (EPFL) developed original passive linkage mechanisms installed in Shrimp [3], and RIKEN developed a prototype vehicle with a Rocker-Bogie mechanism and omni-wheels [4]. The common features of these robots are small diameter wheels, a passive linkage mechanism and high mobility on rough terrain without a reduction in mobility on a flat surface.

In previous research, we developed a wheeled mobile robot “Zaurus”, which had six wheels and a linkage mechanism, and its maneuverability was experimentally

verified. The ability to climb over a 0.20 m high bump, which is twice the height of the wheel diameter of 0.10 m, was achieved; and the newest Zaurus model can climb up 0.15 m high steps. Following this, a control system for Zaurus was developed and evaluated. While many controller designs have been proposed, we focused on a nonlinear control system using a neural network [5, 6]. We have already applied neural networks to controllers for underwater vehicles with nonlinear dynamics, and these have shown good performance in spite of disturbances [7]. Here, control systems using neural networks are designed so that the neural networks express the inverse model/dynamics of Zaurus. The control systems are evaluated in simulations [8] and in experiments using Zaurus [9]. We propose an environment recognition system, because a single control system is limited in terms of the variety of rough terrain environments in which it can be used. An environment recognition system using a self-organizing map (SOM) [10] determines the traveling environment and selects an appropriately modified controller for the conditions [11].

In this paper, we propose a new controller design method based on a neural network. The proposed method involves three kinds of controllers: an elementary controller, adjusted controllers, and simplified controllers. In the experiments, our controller produces less oscillatory motion in outdoor environment and has better performance without heuristic parameter tuning than a well tuned PID controller does.

2 The Wheeled Mobile Robot “Zaurus”

A wheeled robot capable of rough terrain movement, Zaurus, has been developed using six small diameter wheels and linkage structures. Fig. 1(a) is an overview of Zaurus. Zaurus has six wheels with diameters of 0.10 m, and each wheel can be driven independently. The front wheel is fixed to the body and the two pairs of side wheels are connected to the body through side links. The rear link is fixed to the body to maintain the attitude of Zaurus. Zaurus can adapt its shape to terrain conditions because the front fork and the side links are connected to the body with passive joints. Fig. 1(b) shows the 3D CAD and design framework. θ_f is the front fork angle toward the body, θ_s is the side link angle toward the body, and θ_p is the pitching angle of the body toward the ground.

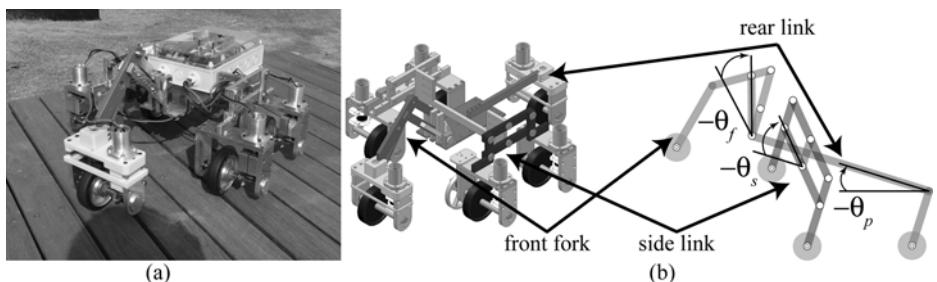


Fig. 1. (a) Overview of Zaurus. (b) 3D-CAD model of Zaurus.

Table 1 gives Zaurus specifications. Zaurus has two control modes, an autonomous mode using the proposed control system and a remote control mode using a host PC or an IrDA remote controller. Two microcomputers integrate data from sensors such as the velocity sensor measuring the traveling velocity of Zaurus, angle sensors measuring the front fork and side link angles, the attitude sensor measuring the pitching angle, and the current sensors measuring the driving currents of each motor. The motor driver controls the angular velocity of the motors. The Ni-MH 14.4 V 3.3 Ah has an operating life of about 1.5 h.

Table 1. Specification of Zaurus

Size (L x W x H)	0.66 x 0.53 x 0.31m
Weight	13kg
Computer System	Laptop PC (Robot PC) PIC18F8720, PIC18F252
Communication	Wireless LAN IrDA Remote Control
Sensor	Angle Sensor (x 3) Attitude Sensor Velocity Sensor Current Sensor (x 6)
Battery	Ni-MH 14.4V 3.3Ah
Actuator	DC Motor 12V 10W (x 6)
Wheel Diameter	0.10m

3 Controller Design for a Wheeled Mobile Robot

To evaluate the maneuverability of Zaurus, three-step stairs and four evaluation functions are prepared. Eq. (1) describes the evaluation functions; E_v evaluates the velocity control, $E_{\dot{v}}$ evaluates the oscillation of the velocity, $E_{\dot{\theta}_p}$ evaluates the oscillation of the pitching angular velocity, and E_J evaluates the consumption energy for climbing up the stairs.

$$\begin{aligned} E_v &= \frac{1}{N} \sum_{n=1}^N (v_r - v(n))^2, & E_{\dot{v}} &= \frac{1}{N} \sum_{n=1}^N (\dot{v}(n))^2, \\ E_{\dot{\theta}_p} &= \frac{1}{N} \sum_{n=1}^N (\dot{\theta}_p(n))^2, & E_J &= \sum_{l=1}^6 \sum_{n=1}^N V |I_l(n)| \Delta t \end{aligned} \quad (1)$$

where v_r is the target velocity of the controller, $v(n)$ is the traveling velocity of Zaurus at the n -th step, θ_p is the pitching angle of Zaurus, V is the supply voltage, $I_l(n)$ is the driving current of the l -th wheel at the n -th step, and Δt is the sampling time, which is 0.05 s.

3.1 Concept of the Proposed Design Method

In general methods to design a control system for a mobile robot in outdoor environment, it is difficult to consider the entirety of the environment. Furthermore, it is difficult to design/adjust a control system for each environment encountered.

In designing a control system, we introduce a neural network. A neural network can express a nonlinear system and has learning and interpolation functions. Fig. 2 shows the design flow for the controller in the proposed method.

First, we design an elementary controller that learns the dynamics of Zaurus in basic environments. This elementary controller has “common sense” for traveling through rough terrain because the neural network learns and complements the basic dynamics of Zaurus.

Second, the controllers are adjusted for different environments, which include not only basic environments but also unlearned environments such as stairs. These adjusted controllers also have “common sense” for traveling through rough terrain because they are modifications of the elementary controller. To achieve efficient rough terrain traveling, Zaurus should memorize the parameters of the adjusted controllers in advance. Furthermore, the central processing unit needs the calculation cost, such as nonlinear function or matrix operation.

Finally, simplified controllers are designed by linearizing the adjusted controllers. The adjusted controllers are able to be linearized around the determinate input because each adjusted controller is used in a specific environment. Therefore, using a simplified controller can reduce the calculation costs.

The adjusted controllers focus on the control system performance. On the other hand, the simplified controllers focus on the calculation costs. In our method, the user can choose between these controllers depending on the user’s needs.

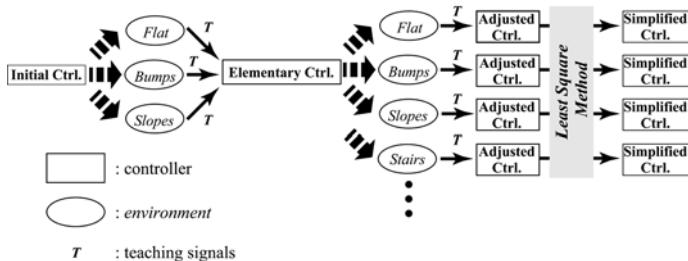


Fig. 2. The design flow for the controller in the proposed method

3.2 Elementary Controller

The learning process of the neural network is carried out to express the inverse model/dynamics of Zaurus. The system identification plays an important role in the neural network learning, and obtaining teaching signals is central to this.

In this research, the teaching signals are obtained from simulations using the mechanical analysis software Dynamics Analysis and Design System (DADS). To obtain the teaching signals, we designed the initial controller as

$$\omega_r = \int \dot{\omega}_r dt, \quad (\dot{\omega}_r = K_P(v_r - v)) \quad (2)$$

where ω_r is the target angular velocity of the motor drivers and the controller parameter $K_P = 30$.

Using the initial controller, various simulations are carried out for basic environments. Here, the basic environments consist of a flat surface, 0.05, 0.10, 0.15 and 0.20 m high bumps, and 10, 20, and 30 degree slopes.

The teaching signal \mathbf{S} consists of the input of the PID controller and the state values of Zaurus. The teaching signal \mathbf{T} consists of the target angular acceleration, $\dot{\omega}_r$, as shown in Eq. (3).

$$\begin{aligned} \mathbf{S} &= [e_v \quad \int e_v dt \quad \dot{e}_v \quad v \quad \theta_p \quad \dot{\theta}_p \quad \theta_f \quad \dot{\theta}_f \quad \theta_s] \quad (e_v = v_r - v) \\ \mathbf{T} &= [\dot{\omega}_r] \end{aligned} \quad (3)$$

where e_v is the difference between the target velocity v_r and the traveling velocity v . θ_p is the pitching angle of Zaurus, θ_f is the front fork angle, and θ_s is the side link angle. The teaching signal \mathbf{S} is the output of Zaurus, and the teaching signal \mathbf{T} is the input of Zaurus. Therefore, the teaching signal \mathbf{S} is the input of the neural network and the teaching signal \mathbf{T} is the output of the neural network because the neural network expresses the inverse model/dynamics of Zaurus. The learning algorithm is back propagation. Fig. 3 shows a block diagram of the elementary controller.

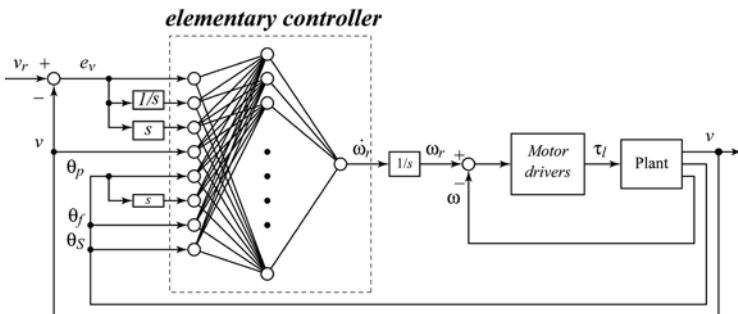


Fig. 3. The block diagram of the elementary controller

3.3 Adjusted Controller

The adjusted controller is designed to focus on the control performance for specific environments.

First, the elementary controller is applied to various environments, such as a flat surface, bumps, slopes, and stairs, and the teaching signals are obtained.

Next, using the teaching signals for each environment, the elementary controller is tuned as the adjusted controller for each environment. The tuning method is back propagation to minimize the error given by

$$E_{err} = \frac{1}{N} \sum_{n=1}^N (v_r - v(n))^2 \quad (4)$$

Finally, the adjusted controllers are obtained by tuning the elementary controller in the specific environments. The adjusted controllers and the elementary controller have the same neural network structure.

3.4 Simplified Controller

The adjusted controllers' input has limitations because the controllers are used in specific environments. Therefore, the adjusted controllers are able to be linearized around the determinate input.

The average \bar{x} and variance σ of the each input is obtained. Concerning one input of the adjusted controller, e.g. e_v , the general input is prepared; one input, e_v , is $[-1, 1]$ and the others are 0. The general input is given to the adjusted controller, and the gradient of the adjusted controller's output is estimated by the least squares method within the range of $[\bar{x} - 2\sigma, \bar{x} + 2\sigma]$.

Fig. 4 shows an example of designing the simplified controller's parameter e_v . The bars are a histogram of the input e_v , the solid line shows the output of the adjusted controller, and the dashed line shows the estimated gradient of the adjusted controller's output. The estimated function is given as

$$y = \kappa x + \lambda \quad (5)$$

Using these parameters, the adjusted controllers are linearized as the simplified controller using

$$\begin{cases} \dot{\omega}_r = \kappa_{e_v} e_v + \kappa_{\dot{e}_v} \dot{e}_v + \kappa_v v + \kappa_{\theta_p} \theta_p + \kappa_{\dot{\theta}_p} \dot{\theta}_p + \kappa_{\theta_f} \theta_f + \kappa_{\dot{\theta}_s} \dot{\theta}_s + \sum \lambda \\ \omega_r = \int \dot{\omega}_r dt \end{cases} \quad (6)$$

3.5 Evaluation of the Controllers

To verify the effectiveness of the proposed method, experiments were executed for the three-step stairs. Fig. 5 shows the experimental results using the adjusted controller for climbing the stairs. As shown in Fig. 5(a), velocity control is achieved by the proposed control system; that is, velocity control by the motor drivers and pseudo-torque control by the adjusted controller.

Table 2 compares the performance of different controllers climbing the stairs. The adjusted controller has better performance than the elementary controller has. This result demonstrates the effectiveness of the tuning of the elementary controller. The simplified controller has better performance than the well tuned PID controller. However, the simplified controller has slightly worse performance than the adjusted controller.

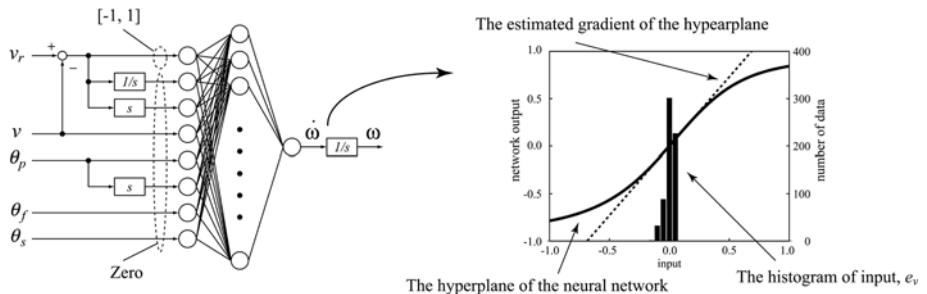


Fig. 4. An example of designing the simplified controller's parameter by least square method

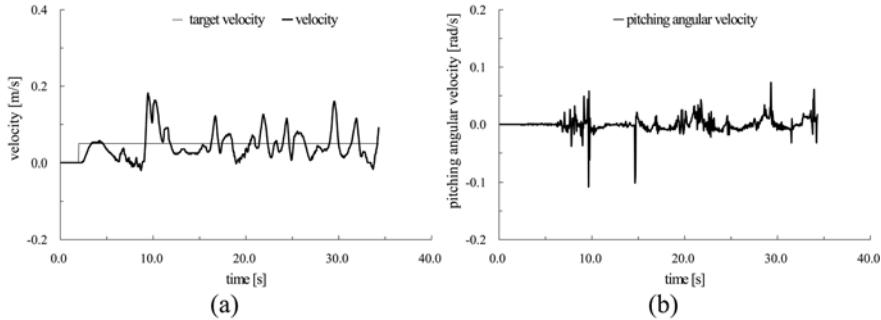


Fig. 5. Experimental results of Zaurus climbing up the three-step stairs by the adjusted controller. (a) The target velocity and the traveling velocity of Zaurus. (b) The pitching angular velocity of Zaurus.

Table 2. Comparison of the performance of well-tuned PID controller, elementary controller, adjusted controller and simplified controller by results of climbing the stairs. The evaluate values are the average of five-trial experimental results.

	E_v ($\times 10^{-3}$)	$E_{\dot{v}}$ ($\times 10^{-5}$)	$E_{\dot{\theta}_p}$ ($\times 10^{-4}$)	E_J ($\times 10^3$)
Well-tuned PID Ctrl.	2.87	7.90	2.41	1.65
Elementary Ctrl.	1.86	6.23	1.76	1.68
Adjusted Ctrl.	1.53	4.63	1.39	1.70
Simplified Ctrl.	2.06	6.19	1.63	1.44

4 Conclusion

In this paper, we proposed a controller design method based on a neural network. The proposed method involves three kinds of controllers: the elementary controller, the adjusted controller and the simplified controller. The elementary controller expresses the inverse model of the wheeled mobile robot through learning the simulation results carried out in basic environments. Therefore, the elementary controller has “common sense” for traveling in rough terrain. The adjusted controller is tuned by the elementary controller in various environments. The adjusted controller also has “common sense” for traveling in rough terrain because the adjusted controller is obtained from the elementary controller. The simplified controller is obtained from the linearized adjusted controller. The adjusted controller can be linearized because it is used in specific environments.

In the experiments, the proposed method achieved velocity control and had less oscillatory motion when climbing stairs. The controller of the proposed method performed better than a well-tuned PID controller did in the experiments.

Acknowledgement

This work was partly supported by a grant of Knowledge Cluster Initiative implemented by Ministry of Education, Cluster, Sports, Science and Technology (MEXT).

References

1. Volpe, R., Balaram, J., Ohm, T., Ivlev, R.: The Rocky 7 Mars rover prototype. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Osaka, Japan, pp. 1558–1564 (1996)
2. Kuroda, Y., Teshima, T., Sato, Y., Kubota, T.: Mobility Performance Evaluation of Planetary Rover with Similarity Model Experiment. In: Proc. of the IEEE International Conference on Robotics and Automation, LA, USA, pp. 2098–2103 (2004)
3. Siegwart, R., Lamon, P., Estier, T., Lauria, M., Piguet, R.: Innovative design for wheeled locomotion in rough terrain. *Journal of Robotics and Autonomous Systems* 40(2), 151–162 (2002)
4. Chugo, D., Kawabata, K., Kaetsu, H., Asama, H., Mishima, T.: Development of a Control System for an Omni directional Vehicle with Step-Climbing Ability. *Advanced Robotics* 19(1), 51–71 (2005)
5. Levin, A.U., Narendra, K.S.: Control of nonlinear dynamical systems using neural networks: Controllability and stabilization. *IEEE Trans. Neural Networks* 4(2), 192–206 (1993)
6. Levin, A.U., Narendra, K.S.: Control of Nonlinear Dynamical Systems Using Neural Networks-Part II: Observability, Identification, and Cotrol. *IEEE Trans. Neural Networks* 7(1), 30–42 (1996)
7. Ishii, K., Fujii, T., Ura, T.: On-line Adaptation Method in a Neural Network Based Control System for AUVs. *IEEE Journal of Oceanic Engineering* 20(3), 221–228 (1995)
8. Sato, M., Ishii, K.: A Neural Network Based Controller System for a Wheel Type Mobile Robot. In: *Brain Inspired Information Technology II*, vol. 1291, pp. 261–264. Elsevier, Amsterdam (2006)
9. Sato, M., Kanda, A., Ishii, K.: Performance Evaluation of a Neural Network Controller for a Wheel Type Mobile Robot. In: *Brain Inspired Information Technology III*, vol. 1301, pp. 160–163. Elsevier, Amsterdam (2007)
10. Kohonen, T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 59–69 (1982)
11. Sato, M., Kanda, A., Ishii, K.: A switching controller system for a wheeled mobile robot. *Journal of Bionic Engineering* 4, 281–289 (2007)

Depth Perception Using a Monocular Vision System

Xuebing Wang* and Kazuo Ishii

2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan

wang-xuebing@edu.brain.kyutech.ac.jp, ishii@brain.kyutech.ac.jp

Abstract. Sensing the distance to an object is a very important clue in many fields and is an area of active research. In our research we propose an alternative method to tell the distance by using motion parallax method that is based on monocular vision and a moving observer with a constant acceleration approximately in an unknown environment. In this paper, our proposed method is applied for the outdoor environment and showed well performance. This is a good point for mobile robots or vehicles to obtain depth by using this method as long as the observer acceleration is known.

1 Introduction

In the sense of vision, the great majority of changes in sensory excitation, that is, in the image, a result from the motion of the light source, of the imaged objects or of the observer. And an important task of vision is to discern the distance of individual points from the observer. In most tasks for movable robots or vehicles depth perception is more valuable than other information. For depth perception there are several methods by using a vision system. To obtain the absolute distances or relative distances from objects in an unknown environment the image information is used such as size, inter-position, perspective etc. Most work on visual 3-D distance detection has been focused on binocular vision which is an accurate method to tell the distance for the objects near to the observer. But for distant objects we can not obtain the accurate depth exactly, also like micro robot or small unmanned vehicles which eyes are close together, and which therefore do not have good stereoscopic vision. There is however another important cue for depth perception, not as commonly used as stereo vision, and that can reveal depth information from a monocular motion field. It is referred to as structure from motion or kinetic depth effect when the object that is observed is moving [1]. The former gives rise to local depth information and the latter can determine distances to objects. There is however an important requirement for determining absolute distance to objects, i.e. the observer motion, with the perceived motion of any object point present in the scene [2]. In a practical scenario the object points belong to image structures of anything from object contours, corners, to local textured regions, e.g. patches of the scene that can be distinguished from the background. Also a system that measures distance for a stationary object by moving the observer has been demonstrated for a 1D sensor array [3]. In our former work we did a Monte-Carlo simulation for depth from motion parallax, also an experiment were performed by fusing a 3-D inertial sensor

* Corresponding author.

and a silicon retina which acceleration was created by system's gravity along a slide to percept depth of a high contrast object in the range of 3 meters[4][5].

We plan to extend this work by using a 2D motion field in order to map the entire field of view into a distance image. And an experiment was performed on the road in this time that one fixed the system on the head to get side view and riding a small motor vehicle to obtain the acceleration. In this work a Logitech web camera with an auto iris technology was used to collect the image data, and a 3-D inertial sensor is used to obtain the acceleration data. In this paper the algorithmic of depth from motion parallax will also be described and emphasize at how to fuse vision and inertial data and finally how to estimate distance from the motion parallax.

2 Depth from Motion Parallax

For a monocular system would correspond to motion parallax or dynamic occlusion etc. In the real world as you move in the environment around this object will demonstrate a phenomenon called motion parallax. Objects closer to one you have fixated will appear to move in the opposite direction to yourself, whereas more distant objects will appear to move more slowly but in the same direction as yourself. The relative apparent motion of objects within your field of view (as you move) provides a strong cue to the relative distance of objects from the observer. If the velocity of the observer is known the distance to static objects can be determined. If also the acceleration of the observer is known the distance can be estimated to objects moving with constant velocity.

The depth algorithm will be derived from the lens geometry based on the static objects or the objects that move with a constant velocity. Fig.1 shows the lens geometry. And we consider a converging thin lens as indicated in equation (1).

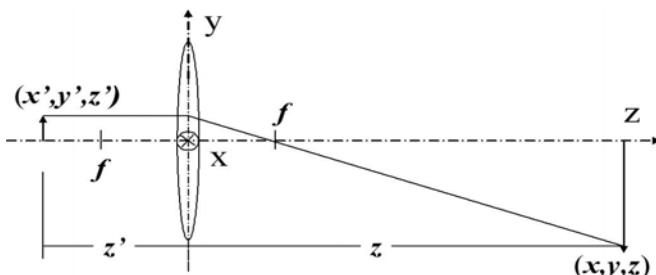


Fig. 1. Thin lens geometry is used to derivation the depth from motion parallax

$$\frac{1}{f} = \frac{1}{z} + \frac{1}{z'} \Rightarrow z' = \frac{z \cdot f}{z - f} = M(z) \cdot z \quad (1)$$

From the lens equation we can derive an equation with lens magnification $M(z) = f/(z-f)$, then for an object at distance the perceived object point is projected as shown in equation (2);

$$(x', y') = (M(z) \cdot x, M(z) \cdot y) \quad (2)$$

Where the prime indicates coordinates as perceived on the focal plane image. The relative velocity in the direction of the observer motion is $\Delta v = v_{obs} + v_o$ and the acceleration difference is $\Delta a = a_{obs} + a_o$, where a_o and v_o are components of object motion, a_{obs} and v_{obs} are components of observer motion, that all are parallel with the direction of observer motion. Both parameters are assumed to be zero, i.e. $v_{obs} = a_o = 0$, meaning that the initial velocity of the accelerating observer is zero and that the perceived object is moving with constant velocity. The relative velocity to the object is dependent on the acceleration so that:

$$\Delta v(t) = \int_0^t \Delta a(t') dt' = a_{obs} t + v_o \quad (3)$$

The relative object position along the direction of observer motion $s(t)$ is one linear path on the X-Y plane. It can be obtained by integration of the velocity difference:

$$s(t) = \int_0^t \Delta v(t') dt' = \frac{1}{2} a_{obs} t^2 + v_o t + s_o \quad (4)$$

Where the perceived position can be calculated from (4) by using the relation (2) so that:

$$s_p(t) = M(z) \left(\frac{1}{2} a_{obs} t^2 + v_o t + s_o \right) \quad (5)$$

The acceleration as it is perceived by the observer can be obtained by first calculating the velocities at the two instances $t + \Delta t$ and $t + 2\Delta t$:

$$\begin{aligned} v_p(t + \Delta t) &\approx \frac{s_p(t + 2\Delta t) - s_p(t)}{\Delta t} = \\ &= M(z) \left(\frac{1}{2} a_{obs} (2t - \Delta t) + v_o \right) \end{aligned} \quad (6)$$

$$\begin{aligned} v_p(t + 2\Delta t) &\approx \frac{s_p(t + 2\Delta t) - s_p(t + \Delta t)}{\Delta t} = \\ &= M(z) \left(\frac{1}{2} a_{obs} (2t + \Delta t) + v_o \right) \end{aligned} \quad (7)$$

Then, by using the perceived velocities at $t + \Delta t$ and $t + 2\Delta t$ the perceived acceleration can be calculated as:

$$a_p(t+2\Delta t) \approx \frac{v_p(t+2\Delta t) - v_p(t+\Delta t)}{\Delta t} = M(z)a_{obs} \quad (8)$$

This can also be written as:

$$a_p = \frac{f}{z-f} a_{obs} \quad (9)$$

Finally the distance as perceived by the observer can be obtained by solving for depth z ,

$$z = \frac{a_{obs}f}{a_p} + f \approx \frac{a_{obs}f}{a_p} \quad (10)$$

From this equation we can calculate the distance from the object to observer by using the observer's acceleration and the perceived acceleration of the object on the image. In this case we do not need to care about the how big the velocity of the object is, or if the velocity is 0, as long as there is no acceleration involved.

3 Simulation

In this work a simulation was performed by using Monte Carlo method to realize the distance from motion parallax algorithm in MATLAB. Gaussian noise was added into the observer acceleration model and the perceived acceleration model in order to be closer to reality.

In the simulation we set the 200*500(rows*columns) points on the ground. The width of one row is 200cm; the length of one column is 500cm. By using ray tracing algorithm we can get a simulation environment like Fig.2. Depths of objects that want to be detected were set to 0.5m, 0.7m, 0.9m ... 5.6m, 5.8m, 6m. The observer acceleration is set to 0.5G. Camera resolution is 320 pixels in horizontal direction. The update rate of camera is set to be 30fps. The focal length of camera is 4.7mm.

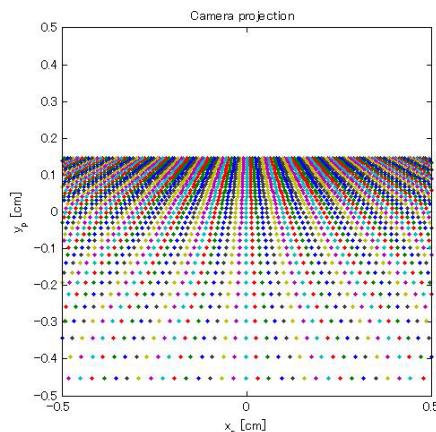


Fig. 2. Camera projection result using ray tracing algorithm

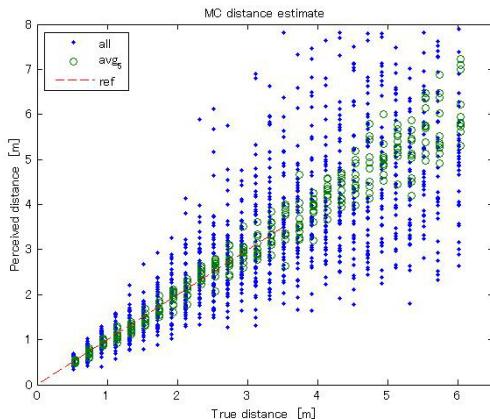


Fig. 3. Depth from motion parallax simulation result, the points are stand for all estimation by using all samples, circles mean the estimation averaging 5 samples in one trajectory

Noise setting is ± 0.1 pixels in perceived image and $\pm 0.5\text{m/s}^2$ separately. The simulation result is shown as Fig.3. For near objects the error contribution from the observer acceleration is a lot bigger than the perceived acceleration noise, because for the near object the perceived acceleration gives a big value in the denominator, so the influence of noise is small, and consequently the error contribution is small. Oppositely, for far objects the error contribution from observer acceleration measurement noise is smaller than that from the perceived acceleration.

4 Experiment

4.1 Experimental Set-Up

An experiment of depth from motion parallax was performed that fixed the system on the driver's head and the driver rides a small motor bike to accelerate for collecting image information and inertial sensing information (Fig.4). The experiment start from



Fig. 4. The inertial sensor and web camera was fixed together and putted on the helmet. Experimental environment is showed as right figure.

static condition and the driving line is parallel to the trees beside the driver. The camera system is putted on the helmet on left side i.e. the optic axis of camera is perpendicular to the driving line. The distance of two parallel lines which are driving line and line of trees is about 2.5 meters. In this experiment 10 times accelerating processes were measured in all.

4.2 Experimental Data Analysis

In the former work we use a high contrast object, so the centroid method is used to calculate the movable distance in focal image, and then we can get the perceived velocity and perceived acceleration. At this time the environment is unknown and objects' centers of gravity are difficult to detect exactly, thus the optical flow method [6] is used to calculate objects' velocities directly [7]. Blob analysis and labeling are also performed for tracking objects [8], so we do not need to care about the center of objects and just focus on the optical flows in this area (Fig.5). Besides the image data the acceleration data is also important for this algorithm (showed in Fig.6). In this experiment the depths to line of trees which are parallel to the observer's driving direction are tested. As shown in the Fig.7 we got the results near to the simulation.

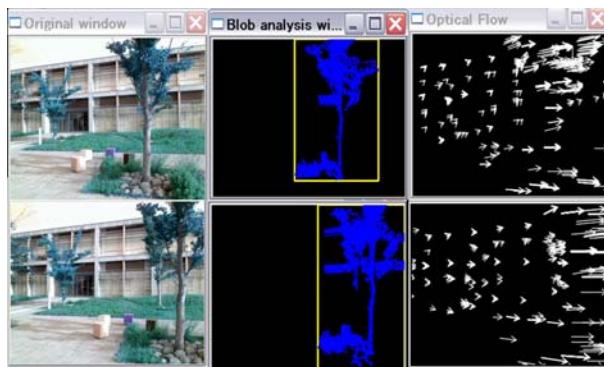


Fig. 5. In the upper windows from left to right there are 91-frame's focal image information from camera, blob analysis window and optical flow window; the 92-frame's information corresponding to 91-frame's is shown in downside figures

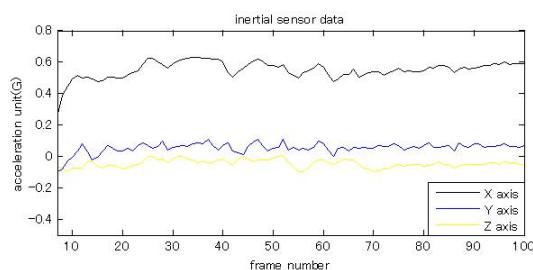


Fig. 6. 3-D Inertial sensor's data

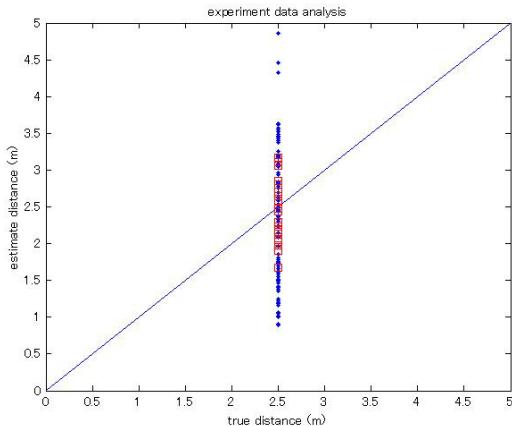


Fig. 7. Experiment data analysis is performed by using all 100 samples. Average data that are calculated by using 5 samples shows by red square.

In the result some values are near to 2 times bigger than the reference in respect that the backgrounds near to the trees in the focal image are not separate clearly. Also the vibration from the bike's engine is a big reason. For solving these problem, new tracking and label model should be added in order to separate the background more detail, more accurately and tracking better.

5 Conclusion

Fusing monocular vision measurements of perceived acceleration with inertial sensor measurements generates a feasible sensing strategy for determining the position between a moving observer and stationary objects or moving non-accelerating objects. From the derivation of the distance equation we know only the observer acceleration for this system is needed in order to calculate the distance to an object, which is a good point for mobile robots or vehicles.

We measured the perceived position of the object on the focal image about every 50ms to calculate the perceived acceleration and also measured the observer acceleration by using the inertial sensor correspondingly. But certainly these measurements are imprecise. For solving these problems, on the one hand a stable acceleration making model is needed. And with it we do not need to care the mobile robots or vehicles' movement, i.e. although the robots or vehicles is static or move with a constant velocity we also can tell the absolute depth from the objects by only using one camera. On the other hand improving image sensing part by using a 2-D optical flow sensor in order to realize a real time system for better use in robotics field is also a future work for us.

References

1. Mallot, H.A.: Computational Vision: Information Processing in Perception and Visual Behavior. MIT Press, Cambridge (2000)
2. Huster, A.: Relative Position Sensing by Fusing Monocular Vision and Inertial Rate Sensors, PhD. Dissertation, Stanford University, USA (2003)
3. Karri, S.S., Titus, A.H.: Toward an analog very large scale integration system for perceiving depth through motion parallax. Optical Engineering 44(5) (2005)
4. Wang, X.: Determining distance from motion parallax with a silicon retina. Master thesis, Kyushu Institute of Technology, Japan (2006)
5. Bejanovic, V., Wang, X.: A vision system for depth perception that uses inertial sensing and motion parallax. In: Mechatronics and Machine Vision in Practice, pp. 65–76. Springer, Heidelberg (2007)
6. Horn, B.K.P., Shunck, B.G.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
7. Bouguet, J.Y.: Pyramidal implementation of the Lucas-Kanade feature tracker description of the algorithm, Intel Corporation, Microprocessor Research Labs, OpenCV Documents (1999)
8. Horn, B.K.P.: Robot vision, pp. 69–71. MIT Press, Cambridge (1986)

Trajectory Planning with Dynamics Constraints for an Underactuated Manipulator

Yuya Nishida and Masahiro Nagamatu

Kyushu Institute of Technology, 2-4 Hibikino,
Wakamatsu-ku, Kitakyushu-shi, 808-0196, Japan
nishida-yuuya@edu.brain.kyutech.ac.jp,
Nagamatu@brain.kyutech.ac.jp

Abstract. The attitude control of a horizontal underactuated Manipulator (UAM) is a difficult control problem because of its second-order nonholonomic constraints. This paper proposes a new method for the trajectory planning of the UAM. In the proposed method, the trajectory planning is transformed to a constraint satisfaction problem, and the constraint satisfaction problem is solved by using a sequence of multi-layer perceptrons which are trained to be a forward model of the UAM. We show the effectiveness of our method by several experimental results.

Keywords: Underactuated manipulator, trajectory planning, constraint satisfaction, multi-layer perceptron and Lagrangian method.

1 Introduction

Recently, the control and the analysis of an underactuated manipulator (UAM) which has strongly nonlinearity have been attracted. Usually each joints of a general manipulator have an actuator one by one. On the other hand, an UAM has fewer actuators than the degree of freedom of it, i.e. the UAM has passive joints which are not actuated. The UAM is used when we want to save the weight and the electric power consumption in a system, by using fewer actuators [1][2].

When a manipulator has a passive link, it cannot apply torque to the link. Torque is transferred to the passive link by inertia force. To control the UAM, we should use inertia force that is rather considered to be a disturbance in the traditional control system. Thus, it is difficult to control the attitude of the UAM. Mechanically, it is well known that the UAM has nonholonomic constraints. It is also known that a horizontally-located UAM the second joint of which is passive has second-order nonholonomic constraints [2].

We are studying the attitude control of this type of the UAM. For the control we use the combination of the reinforcement learning and the trajectory planning. The control task is to move the UAM from an initial state to a target state. The reinforcement learning is an unsupervised learning, and well used for automatic acquisition of control rules for many robots, cars, and manipulators [3][4]. However, generally, the reinforcement

learning needs huge amounts of time to acquire the control rules, because basically it learns only from huge number of trial-and-errors. On the other hand, the trajectory planning plans a trajectory under several constraints and an objective function [5][6][7]. The time needed to plan a trajectory for the trajectory planning is very short compared with the time to acquire rules for the reinforcement learning. However, after the training is finished, the reinforcement learning has acquired the control rules from all initial states to the goal state, while the trajectory planning plans a trajectory from only one initial state at a time. When the trajectory planning is combined to the reinforcement learning, the former can give the obtained trajectory to the latter as a successful sample which might shorten the training time of the latter. Furthermore even if small errors are included in the trajectory, the latter might adjust the error.

In this paper, we propose a new method for the trajectory planning of a horizontal two-link UAM, the second joint of which is passive. Already proposed trajectory planning algorithms for nonholonomic cars and manipulators can use only limited kind of movements for trajectory [6][8], while our method does not restrict the movements. Hence, we think our methods are more flexible and powerful. We show the effectiveness by experiments. The combination of our methods with the reinforcement learning is not implemented yet, hence it is not included in this paper.

2 Underactuated Manipulator (UAM)

Figure 1 is shown overview of the UAM used in this research. The first joint is the active joint, and the second joint is the passive joint. The length of the first and the second links is 200 millimeter. The torque τ applied to the active joint is limited to $-0.1 \leq \tau \leq 0.1$ Newton, angle θ of each joint, $-2\pi \leq \theta \leq 2\pi$ radians, and angular velocity $\dot{\theta}$ of each joint, $-20 \leq \dot{\theta} \leq 20$ radians per second. Both of first and second links are of uniform mass (0.108 grams). First and second joints are applied static friction ($f_{s1}=0.01$, $f_{s2}=0.001$) and dynamics friction ($f_{d1}=0.0015$, $f_{d2}=0.00015$). The Lagrange's equation for motion of this UAM is described as follows. In addition, applied torque τ_2 to the second joint is always zero Newton, because the second joint is the passive.

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} + \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (1)$$

$$\begin{cases} M_{11} = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2) + I_{c1} + I_{c2} \\ M_{12} = M_{21} = m_2 (l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2) + I_{c2} \\ M_{22} = m_2 l_{c2}^2 + I_2 \end{cases} \quad (2)$$

$$\begin{cases} C_1 = -m_2 l_1 l_{c2} (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \sin \theta_2 \\ C_2 = m_2 l_1 l_{c2} \dot{\theta}_1^2 \sin \theta_2 \end{cases} \quad (3)$$

$$\begin{cases} F_1 = f_{s1} \text{sign}(\dot{\theta}_1) + f_{d1} \dot{\theta}_1 \\ F_2 = f_{s2} \text{sign}(\dot{\theta}_2) + f_{d2} \dot{\theta}_2 \end{cases} \quad (4)$$

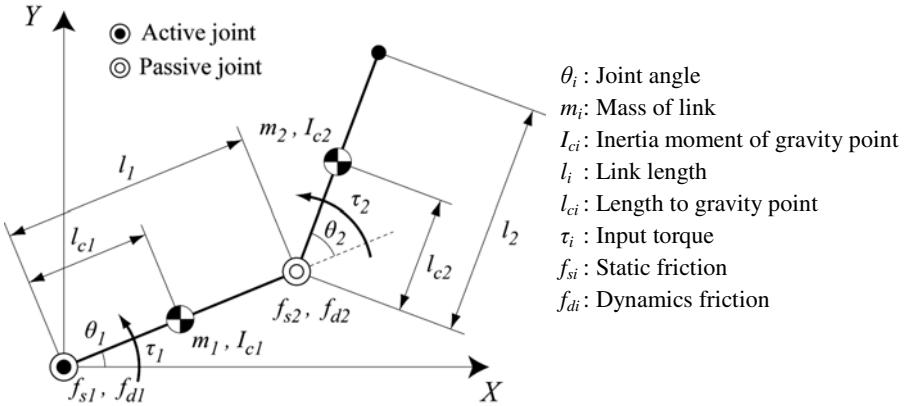


Fig. 1. Overview of a UAM in this research

3 Trajectory Planning

For the trajectory planning we use a network in which n multi-layer perceptrons (MLPs) are connected in series as shown in Fig.2. Each MLP has learned the forward model of the UAM, i.e., if a state of the UAM $s(t)=[\theta_1(t), \theta_2(t), \dot{\theta}_1(t), \dot{\theta}_2(t)]^T$, and torque $\tau_1(0)$ are input to a MLP, it outputs the next state $s(t)=[\theta_1(t+1), \theta_2(t+1), \dot{\theta}_1(t+1), \dot{\theta}_2(t+1)]^T$ of the UAM (the unit time is determined appropriately). Thus, if the initial state $s(0)$ and a sequence of torques $T=\{\tau_1(0), \tau_1(1), \dots, \tau_1(n)\}$ are input to the MLPs as shown in Fig.2, we can obtain a sequence of states $S=\{s(0), s(1), \dots, s(n+1)\}$.

$$T = \{\tau_1(0), \tau_1(1), \dots, \tau_1(n-1), \tau_1(n)\} \quad (5)$$

$$S = \{s(0), s(1), \dots, s(n), s(n+1)\} \quad (6)$$

If the state $s(n+1)$ of the final output of the network is the target state s^* , then the sequence of states $S=\{s(0), s(1), \dots, s(n+1)\}$ is the trajectory which is the solution of the trajectory planning problem. Thus, our objective is to find the sequence of torques $T^*=\{\tau_1^*(0), \tau_1^*(1), \dots, \tau_1^*(n)\}$ which make the final state $s(n+1)$ equal to the target state.

At first, in this chapter the learning method of forward model of the UAM is explained. Then, the trajectory planning using the network shown in Fig.2 is explained.

3.1 Learning of Forward Model

The multilayer perceptron (MLP) used for learning forward model is shown in Fig.3. The inputs to the MLP are input torque $\tau_1(t)$ to the active joint, the angles $\theta(t)$ s and the angular velocities $\dot{\theta}(t)$ s of the first and the second links. The teacher data for the outputs are the angles $\theta(t+1)$ s and the angular velocities $\dot{\theta}(t+1)$ s which are obtained by a simulation of the Lagranges equation of motion (1,2,3,4).. MLP learns the

forward model by the online training mode where training data are generated randomly. Usually it is difficult to reduce the learning errors to zero, hence the training is terminated appropriately. In our method, the trained MLPs are connected as described above. Hence, even if the learning error of the single MLP is small, there is a possibility that errors are accumulated and grow large at last. To prevent this, after the online training is terminated, MLP is trained again in batch mode using training data obtained from sample trajectories. These sample trajectories are made by giving some patterns of torques to the forward model for appropriate time durations.

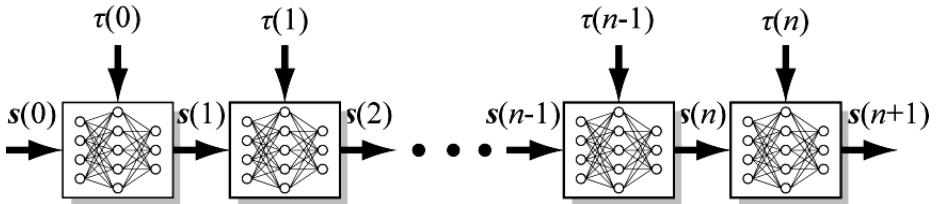


Fig. 2. Network that neural network of n were combined in series

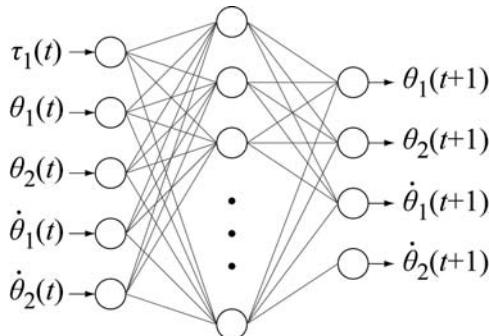


Fig. 3. Multilayer perceptron trained for the forward model of the UAM

3.2 Trajectory Planning Using Sequence of MLPs

The trajectory planning is to find the sequence of input torques T^* for which the final output of the network as Fig.2 is the target state. This can be described as an constraint satisfaction problem as follows.

$$\begin{aligned}
 \text{Find} & \quad \mathbf{T}^* = \{\tau_1^*(0), \tau_1^*(1), \dots, \tau_1^*(n)\} \\
 \text{which satisfies} & \quad h(\mathbf{T}^*) = \mathbf{0} \\
 \text{where} & \quad h_i(\mathbf{T}) = \frac{1}{2} (s_i^* - s_i(n+1))^2 \\
 \text{and} & \quad s_i = \theta_1, \theta_2, \dot{\theta}_1, \text{ or } \dot{\theta}_2
 \end{aligned} \tag{7}$$

This constraint $h(T)$ is a vector of squares of differences between final output $s(n+1)$ and the target state. Now, this constraint satisfaction problem is transformed to a Lagrangian function as follows.

$$E(\tau_1, w) = w^T h \quad (8)$$

$w (\geq 0)$ is a vector of Lagrange multipliers (they also can be considered to be weights of constraints). $E(T, w) \geq 0$ and $E(T^*, w) = 0$ where T^* is a solution of the constraint satisfaction problem.

To solve the constraint satisfaction problem, we use the Lagrangian method instead of using steepest descent method for fixed w . The steepest descent method is usually trapped by local minima while the Lagrangian method does not trapped by any point which is not the solution of the constraint satisfaction problem[8][9]. Dynamics of the Lagrangian method is described as follows. We call this dynamics “LPPH” following [8] and[9].

$$\frac{d\tau}{dt} = -\eta \frac{\partial E(\tau_1, w)}{\partial \tau} \quad (9)$$

$$\frac{dw}{dt} = \frac{\partial E(\tau_1, w)}{\partial w} - \alpha w \quad (10)$$

4 Experiments

We applied our method to the UAM shown in Fig. 4. In this figure, (a), (b) and (c) are experimental results. In all experiments the initial state is $s(0) = [0.0, 0.0, 0.0, 0.0]^T$, and the target states are determined as shown in Table 1., In (a), (b) and (c), “Start” and “Target” show the positions of initial and target states. The numbers of MLPs used are 100, 200, and 300, respectively, and the unit time used is 0.1 sec, this means that the elapsed times of these trajectories are 1 sec, 2 sec, and 3 sec, respectively. The final states of each trajectory are shown in Table 2. From this table we can see that there remain small errors, i.e., the final states of the trajectories found do not coincide the target state precisely. Especially, angle and angular velocity of second link include relatively-great errors.

We think these errors are caused by the deficiency of the training of the MLP. In the training learning data are made by randomly generating input torque $\tau_1(t)$, the angles $\theta(t)$ s and the angular velocities $\dot{\theta}(t)$ s of the first and the second links, and the teacher data $\theta(t+1)$ s and $\dot{\theta}(t+1)$ s are obtained by a simulation of the Lagranges equation of motion (1,2,3,4). This means that it seldom occur that the teacher data become the state shown in Table 1. We have to devise better way of generating training data.

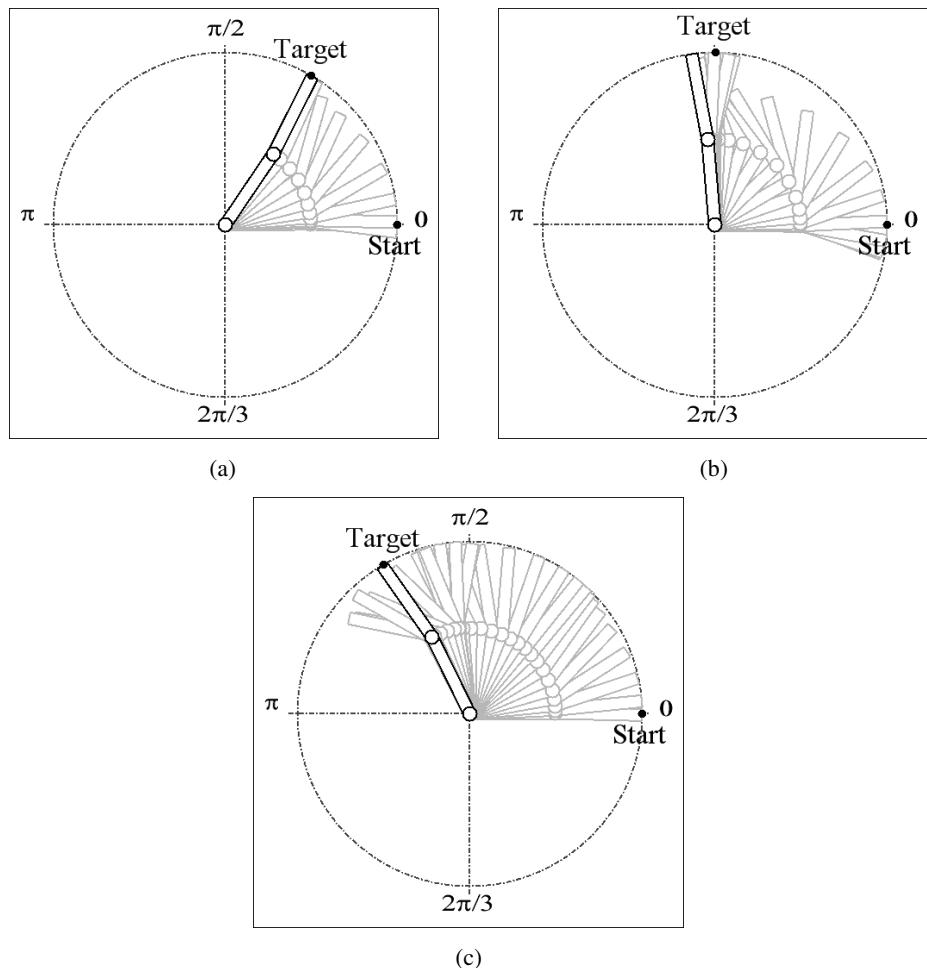


Fig. 4. Experimental results

Table 1. Target states

	θ_1^*	$\dot{\theta}_1^*$	θ_2^*	$\dot{\theta}_2^*$
(a)	1.05	0.00	0.00	0.00
(b)	1.57	0.00	0.00	0.00
(c)	2.09	0.00	0.00	0.00

Table 2. Final outputs

\backslash	θ_1^*	$\dot{\theta}_1^*$	θ_2^*	$\dot{\theta}_2^*$
(a)	0.98	-0.07	0.13	0.19
(b)	1.65	0.08	0.10	0.13
(c)	2.00	0.00	0.28	0.03

5 Conclusion

In this paper, we proposed a new method for the trajectory planning of the UAM which has second-order nonholonomic constraints. In our method the problem of trajectory planning is transformed to a constraint satisfaction problem, and the constraint satisfaction problem are solved by the Lagrangian method. By experiments we showed that trajectory to arbitrary target states can be found effectively. The advantage of our method is that the state transitions available are less restricted than other methods. However, from the experiments we also know that if a trajectory requires state transitions which is not used in the training phase of the MLP, this method cannot find the precise trajectory but stops at the one which has error in some extents. This means that this trajectory planning method needs MLPs which learnt an exact forward model.

For future work, we have to study the learning method of forward model by the MLP. Also, we have to develop a method to combine the trajectory planning and the reinforcement learning.

References

- Oriolo, G., Nakamura, Y.: Free-Joint Manipulators: Motion Control under Second-Order Nonholonomic Constraints. In: Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS 1991), pp. 1248–1253 (1991)
- Arai, H.: Noholonomic Control of Underactuated Manipulator. System / Control / Information 43(10), 360–553 (1999)
- Shibuya, T., Shimada, S., Hamagami, T.: Swinging up Acrobot using Complex-Valued Reinforcement Learning. In: The 21st Annual Conference of the Japanese Society for Artificial Intelligence (2007)
- Yoshimoto, J., Ishii, S., Sato, M.-a.: Application of Reinforcement Learning Based on On-Line EM Algorithm to Balancing of Acrobot, Institute of Electronics. Information and Communication Engineers, J83-DII (3), 1024–1033 (2000)
- Blackmore, L., Williams, B.C.: Optimal Manipulator Path Planning with Obstacles Using Disjunctive
- Shiroma, N., Lynch, K.M., Arai, H., Tanie, K.: Motion Planning for a Three-Axis Planar Manipulator with a Passive Revolute Joint. Transactions of the Japan Society of Mechanical Engineers (C) 66(642), 545–552 (2000)
- Shiroma, N., Arai, H., Tanie, K.: Noholonomic Motion Planning for Coupled Planar Rigid Bodies with Passive Revolute Joints. Japan Society of Mechanical Engineers (C) 66(642), 545–552 (2000)
- Barraquand, J., Latombe, J.-C.: Nonholonomic Multibody Mobile Robots: Controllability and Motion Planning in the Presence of Obstacles. Algorithmica 10, 121–155 (1993)
- Nagatatu, M., Yannaru, T.: On the stability of Lagrange programming neural networks of satisfiability problems of propositional calculus. Neurocomputing 13, 119–133 (1995)
- Zhang, K., Nagamatu, M.: Solving SAT by Mixed Parallel Execution with Biased Population Ratio. In: Proceedings of the 12th International Conference on Neural Information Processing (ICONIP 2005), pp. 139–144 (2005)

Neural Networks That Mimic the Human Brain: Turing Machines versus Machines That Generate Conscious Sensations

Alan Rosen and David B. Rosen

Machine Consciousness Inc. Redondo Beach California, USA
arosen@mcon.org, dbenrosen@mcon.org
www.mcon.org

Abstract. This paper shows that neural-net based machines may be designed to mimic the consciousness-sensations generated by the human brain. It is shown that the standard definition of biological modalities of the tactile and visual receptors, coupled with the law of specific nerve energy, leads to a fundamental relationship that relates human subjective experiences, or consciousness, to explicit neuronal activity. Such a relationship is a giant leap forward in the study of consciousness since it converts the parameters of consciousness, which have never been amenable to mathematical calculations, into mathematically calculable functions.

1 Introduction

It is highly significant that for the past 150 years the modalities of the biological receptors have been defined by the scientific community and in all neuroscience textbooks [1] to be tightly coupled with subjective experiences of conscious sensations (sensations of touch-feel, visual-seeing, auditory-hearing, olfactory smelling, and gustatory tasting). Specific conscious sensations are associated with each biological receptor. This paper examines the conscious sensation, or subjective experience associated with the modalities of the biological tactile and visual receptors. The examination is performed by reverse engineering the modalities of the tactile and visual receptors, the connectivity of the receptors and afferent axons, and the central connections in the brain activated by the receptors. The central connections in the brain have been reverse engineered by means of a neural net based robotic controller, a sensory motor control system that mimics the human brain in controlling a humanoid robotic system. The reverse engineered design of a neural-net based human-like robotic sensing machine, and the reverse engineered robotic controller, known as a Neuronal Correlate of a Modality (NCM)-circuit has been presented at scientific meetings and published in the peer reviewed literature [2]. For example, Figure 1 is a pictorial representation of a neural net based humanoid robotic “seeing” machine that operates as a visual automobile-driving robot. It is part of a reduction to practice experiment proposed to DARPA by MCon Inc.

This paper examines the question whether the NCM-controller, and the robotic body controlled by the NCM-controller, may generate the same subjective experiences of



Fig. 1. A pictorial representation of a neural net based humanoid robotic “seeing” machine that operates as a visual automobile-driving robot. It is part of a reduction to practice experiment proposed to DARPA by MCon Inc.

“conscious sensations” that are known to be generated by the analogous biological receptors and the central connections in the brain activated by the receptors.

2 Method

The biological “conscious sensations,” or modalities of tactile and visual receptors consist of feelings of touch-pain, touch-feel, and visual-seeing of colored pinpoints of light¹. In the design of the NCM-circuit, the tactile modalities have been used to determine the sensory motor controlling reaction forces exerted on the robotic system [2]. Whereas the visual modalities have been used to give the robot a visual-seeing navigational capability[3]. This paper compares the biological visual modality, which is an established visual “conscious” sensation generating mechanism, with the NCM-controller that mimics the brain.

The 3D-Wheatstone illusion: In the reverse engineered neural-net based visual model, a 3-D-Wheatstone-type [5] image is formed that is a high fidelity representation of 3D-objects that gave rise to that image. Two camera CCD receptors separated by binocular disparity, receive two 2D-images on the CCD-arrays of both cameras. The neural net based controller-brain converts these two 2D-images into a

¹ The modalities of the biological receptors are the only subjective experiences or forms of “consciousness” that have been scientifically categorized, measured and published in peer reviewed scientific journals and scientific text books over the past 150 years [1]. A field of science, known as psychophysics[4], has been devoted specifically to the study of the subjective experiences generated at the biological receptors.

3D-photometric image that is a high fidelity representation of the object that gave rise to the two 2D-images [3]. In the biological brain, this 3D-image is sensed as a subjective experience, a “conscious-illusion.” Figure 2 shows the “conscious-illusion” generated by the Wheatstone stereoscope. This illusion, defined by the collective modalities of the visual receptors, generally represents the viewed objects that gave rise to the illusion.

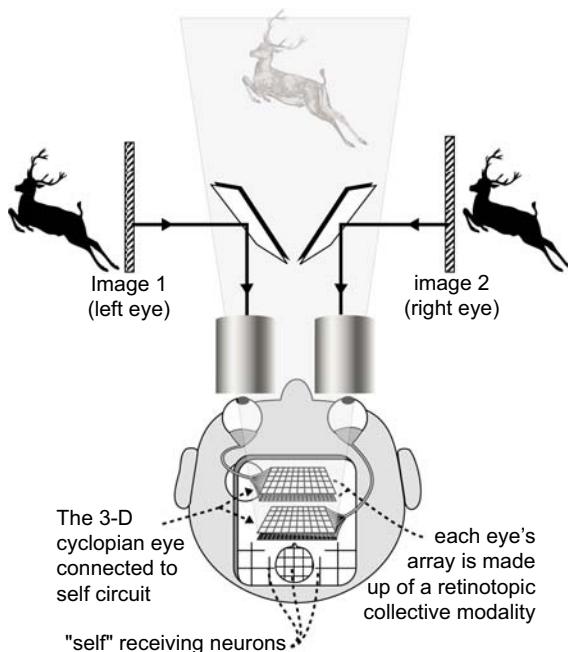


Fig. 2. A Wheatstone stereoscope 3-D Illusion. The illusion is a conscious experience defined by the collective modalities of the visual receptors.

The two-camera mechanical visual system reverse engineers the biological two-retinal arrays of receptors separated by the biological binocular disparity of the 2-eyes. In the biological system the two eyes receive two 2D-images on the retinas of both eyes, whereas the brain converts these two 2D-images into a 3D-illusion that is a high fidelity representation of the object that gave rise to the two 2D-images [3], [5].

Both the reverse engineered NCM-visual system, shown in Figure 3, and the human brain represented in Figure 2, generates a 3D-image/illusion, based on binocular disparity data. In the NCM-visual system the 3D-image/illusion is a real 3D-photometric image that is projected onto the 3D-coordinate frame in which the robot is operating (see Figure 3). In the biological brain the 3D-illusion, which is perceived by each individual is presumably due to neuronal activity, a neuronal circuit in the brain that generates the “consciousness” sensation. The sensation is a high fidelity 3D-image/illusion based on the two 2D-images falling on the retinas of both eyes.

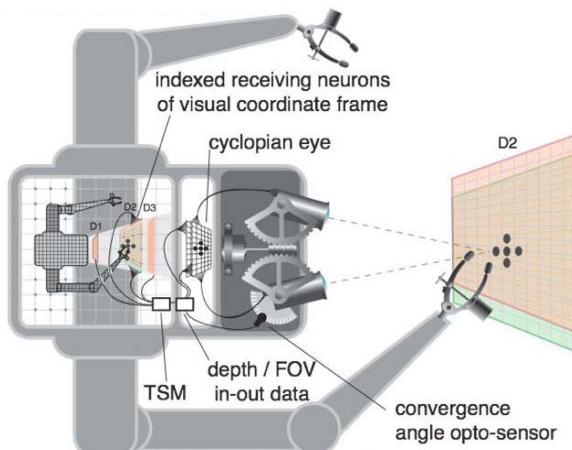


Fig. 3. The reverse engineered 3D-photometric image generated by the electronic NCM visual system

The question of how the brain converts neuronal activity into a subjective experience-illusion is discussed in the conclusion section.

The transduction process: The conversion of subjective experience into neuronal activity may be viewed as a transduction process. Two forms of transduction are associated with the modality of each receptor, a localized sensory transduction process and a “circuit” transduction process generated in the circuit associated with the central connections in the brain.

The localized sensory transduction occurs at the receptor. In all cases the receptor converts the incident energy into neuronal activity. The neuronal activity does not differentiate between the various forms of subjective experiences associated with the various receptors. The same, undifferentiated neuronal activity is generated by a) Pressure or force exerted on the tactile receptors, b) Light energy falling on the rod/cone visual receptor, c) Sound energy falling on the hair follicles of the ear, and d) the application of gaseous or liquid chemicals falling on the olfactory or gustatory receptors[1]. According to the law of specific nerve energy, it is the neuronal circuit or central connections associated with the receptor that differentiates between the various modalities² [4], [1].

The “circuit” transduction process occurs either in the central connections in the brain or the NCM-circuit that reverse engineers the central connections in the brain. The central connections in the brain or the NCM circuit may be viewed as a transformation circuit that converts modality-type subjective experiences, denoted by the letter S, into neuronal activity, denoted by the letter N. The transformation is expressed mathematically by, $S=f(N)$, denoting that the subjective experience is a

² In the field of psychophysics it has been proved by the law of specific nerve energy, that it is the biological design of the central connections in the brain that give rise to the subjective experience of “conscious sensations” [1], [4].

function of neuronal activity. The mathematical relationship is particularly important in the study of “consciousness,” since it is the only scientific mathematical relationship that relates a subjective experience, not amenable to mathematical calculations, to neuronal activity that is mathematically amenable to calculations in the physical reality domain (see the conclusion section).

The subjective experience domain, S, versus the physical reality domain, N: The letter S denotes the subjective experience domain, a domain whose members are all the subjective experiences that a human may experience. Namely feelings of pain, touch-feel, visual-seeing, auditory-hearing, olfactory-smelling, gustatory-tasting, and all emotional feelings such as anger, hate, love contentment, etc. The members of the subjective experience domain are not amenable to calculations by the ordinary laws of mathematics.

The letter N denotes the physical reality domain, a domain whose members are all the calculable parameters of science and mathematics. These parameters include the magnitude and frequency of bioelectric activity, the magnitude and direction of mechanical parameters of force, mass, and time, thermodynamic parameters, sound parameters, electromagnetic parameters, and chemical composition parameters. All the parameters in the physical reality domain may be denoted by symbols that form calculable functions that obey the laws of mathematics.

The circuit that transforms neuronal activity f(N) into a subjective experience: For the 5-external sensory modalities, the S=f (N)-circuit transforms the neuronal activity in the circuit into a subjective experience-sensation that is defined to be the modality of that receptor. The neuronal activity circuit is made up of the receptor, the afferent axons connected to the receptor and the central connections in the brain. This circuit has been reverse engineered by means of electronic neural networks and is called a Neuronal Correlate of a Modality (NCM)-circuit [2].

It is a unique circuit associated with each receptor and its central connections. Each circuit is specifically designed to generate the unique modality associated with the receptor. The authors identify the NCM-circuit as a special case of the long sought after Neuronal Correlate of Consciousness (NCC) [6], wherein the NCM-circuit= f(N)=a member of the S-set.

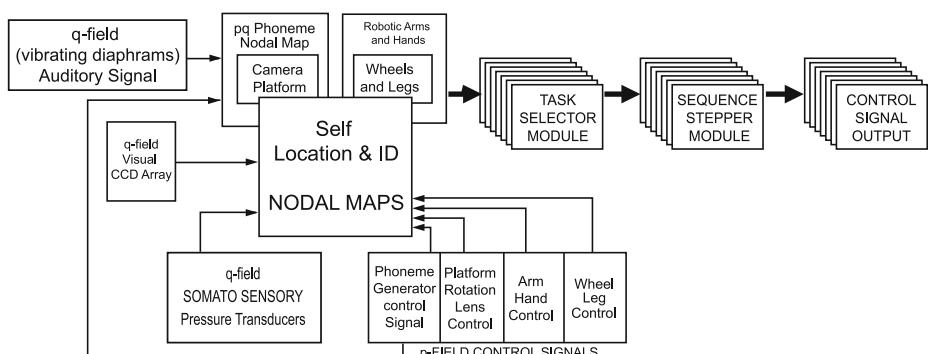


Fig. 4. A block diagram showing the design of the electronic NCM circuit

The authors postulate that the electronic NCM-circuit is a sensation generating mechanism just as the $S=f(N)$ neuronal circuit in the brain, represented by the central connections in the brain, is the sensation generating mechanism that generates the sensation defined by the modality of the receptor.

3 Conclusion

Turing machines [7] that operate only on parameters that are members of the physical domain can never generate subjective experiences or feelings that are members of the subjective experience domain. It is impossible to build a “conscious Turing-machine” [7] without expressing subjective experiences in terms of neuronal activity ($S=f(N)$). The most scientific, non-speculative approach at finding such a function is to utilize the accepted definition of biological receptors, the law of specific nerve energy, and the existence of an NCM-transformation circuit (in the form of $S=f(N)$). Such functional relationships have been experimented on, and studied for over 150 years in the field of psychophysics [4]. NCM-circuits have been designed and published for the tactile modality [2] and for the visual modality [3]. In the case of the 5-biological external receptors, the NCM-circuits are neural-net based circuits that present a robot with self location and identification form of knowledge in a coordinate frame located within the controller of the robot. Therefore one component of the subjective experience, experienced by the subject, is the feeling of knowing the location of the modality relative to the location of each and every part of the robotic “self” [2], [3].

Reverse Engineering a sensation generating mechanism in the brain: How does the brain convert neuronal activity into subjective experience/illusions: An important clue comes from the design of the NCM-sensory motor control system. All sensory perceptions of the NCM-robot, and all responses of that robot occur in a coordinate frame in which the robotic “self” is at the center of the frame of reference. The robot is programmed, or learns, to react to the environment in terms of perceptions of self-location and identification of all body parts in a coordinate frame located inside the controller and centered at the center of mass of the robot. All modality/perceptions are perceived with respect to the “self,” and all learned actions are with respect to the modalities of the self-location and identification circuit. The NCM-circuit facilitates the programming or learning of “self knowledge,” since all learned actions are preformed with respect to the “self.”

The reverse engineered NCM-circuit, operating wholly on the basis of modality/receptor signals, controls the robotic body with parallel processing, control of all motor/muscles simultaneously, and mimics the operation of the human brain. In the human brain the $S=f(N)$ -circuit is operable since humans experience “conscious”-subjective experiences. In the NCM-robot, even if the robot did not experience “conscious” subjective experiences, the operational-behavior would be indistinguishable from human behavior. That is, the response of the robot to tactile “pain” or visual “obstacle” would be identical to the human response. It is not possible to prove that an NCM-machine experiences subjective experience, just as it is not possible to prove that humans experience the subjective experiences that they report to another human examiner.

Generalizing the S=f(N)-circuit to emotional modalities: There are many subjective experiences, members of the set of subjective experiences, the S-set, which do not have a well-defined receptor associated with them (For example the emotions of rage, fear or love). In an article titled “The design of the Neuronal Correlate of Emotions” the authors [8] utilize William James’ assumption [9] that emotions are the modalities of internal sensors associated with the individual’s involuntary autonomic systems (e.g. internal sensors that increase heart rate, alertness (adrenalin), blood pressure, etc). Following William James, it may be hypothesized that subjective experiences, members of the S-set, other than modalities of the 5-biological external sensors, may be modalities of, yet undiscovered, internal sensors. Thus, it may be hypothesized that all subjective experiences (including emotions), members of the S-set, are modalities of biological internal and external sensors. And that there exists an NCM-circuit associated with the internal sensor modalities as well as the external sensor modalities [8].

Darwin’s law of consciousness: A giant leap forward in the study of consciousness occurs with the recognition that the existence of an S=f(N)-circuit in the brain is a Darwinian adaptation that developed in many species that have a central nervous system³. Reverse engineering the biological adaptation of the NCM-S=f(N)-circuit in the brain is a giant step forward in the study of the last un-known frontier of neuroscience, the frontier of brain-consciousness. A S=f(N)-circuit, discovered for each “subjective experience,” is a quantitative tool that brings members of the subjective experience domain into the physical reality domain.

The authors postulate and hypothesize that any animal or robot controlled by an NCM-circuit, operating exclusively on modality-“self-knowledge”-data may experience a “subjective experience” sensation similar to, but not identical to those experienced by humans.

References

1. All text books in Neural Science and Medical physiology. Some examples are: Kandel, E.R., Schwartz, J.H., Jessell, T.M. (eds.) (1991) Principles of Neural Science. Norwalk Conn; Haines, D.E. (ed.) (2002) Fundamental Neuroscience 2nd ed. Churchill Livingston: Philadelphia PA; Guyton, A.C. (1991). Textbook of Medical Physiology. Philadelphia: W.B. Saunders Co.
2. Rosen A., Rosen D.B.: (2006a). The Design of a Sensation-generating Mechanism (SgM) in the Brain: A first step towards a quantitative definition of consciousness. submitted for publication in the ASSC- Consciousness and Cognition Journal. (See ASSC-e-archive and available for viewing at www.mcon.org); (2006b). A Neural Network Model of the Connectivity of the Biological Somatic Sensors. IJCNN: Proceedings of the IEEE-WCCI. July 16-21, 2006 Vancouver Canada; (2006c). An Electromechanical Neural Network Robotic Model of the Human Body and Brain: Proceedings of the ICONIP- 2006 Hong Kong; King, et al. (eds.) LNCS Part 1 pp.105–116. Springer, Heidelberg

³ It is assumed that the adaptation of the brain organ, and in particular the S=f(N)-circuit in the brain, adheres to Darwin’s law. A more quantitative re-statement of Darwin’s law is: “All members of extant specie are genetically designed with adaptations of organs and drives that statistically enhance the survival of that specie in its environmental niche.”

3. Rosen A., Rosen D.B.: (2007a) A neural Net Based Robotic Optical Circuit that Generates 3D-visual Images. IEEE-xplore proceedings of the IJCNN-Orlando 2007; (2007b) A Robotic Neural Net Based Visual sensory Motor Control system that reverse engineers the Motor Control functions of the Human Brain. IEEE-Xplore IJCNN-Orlando 2007; (2007c). A Neural Net Based Robotic Optical Circuit that Generates 3D-visual Images: Reverse engineering the neurophysiology of the modalities of the retinal receptors. To be published in the Neural Network Journal (See preview at <http://www.mcon.org>)
4. Fechner, G.: (1860). Elements of Psychophysics, vol. 1. Howes, D.H., Boring, E.G. (eds.), HE Adler (trans) New York: Holt, Rinehart and Winston; Weber, E.H. (1846). Der Tastsinn und das Gemeingefühl. In: Wagner, R. (ed.) Handwörterbuch der Physiologie, vol. 3, Part 2 pp. 481–588, 709–728. Vieweg, Braunschweig
5. Charles, W.: Contributions to the physiology of vision. On some remarkable and hitherto un-observed phenomena of binocular vision. *The Philosophical Magazine* 3, 241–267, 504–523 (1852)
6. Metzinger, T. (ed.): Neural correlates of Consciousness. MIT Press, Cambridge (2000); Chalmers, D.J.: What is a Neural Correlate of Consciousness (2002); In: Metzinger, pp. 13–39 (2000); Crick F., Koch, C.: A framework for Consciousness. *Nat. Neurosci.* 6, 119–126 (2003)
7. Turing, A.M.: Computing Machinery and Intelligence. *Mind* 59, 433–460 (1950)
8. Rosen, A., Rosen, D.B.: The Design of the Neuronal Correlate of Emotions: A hedonic Motivational system in the Brain. *MCTJ-android Engineering* 1(1), 74–111 (2005a); The Design of the NCC-circuit for Audition and sound generation: Comprehension, verbalization, conceptualization and declarative Memory. *MCTJ-android Engineering* 1(1), 112–134 (2005b)
9. James, W.: What is an Emotion? *Mind* 9, 188–205 (1884)

Part VIII

Special Session: Lifelong Incremental Learning for Intelligent Systems

A Vector Quantization Approach for Life-Long Learning of Categories

Stephan Kirstein^{1,2}, Heiko Wersing², Horst-Michael Gross¹, and Edgar Körner²

¹ Ilmenau University of Technology,
Neuroinformatics and Cognitive Robotics Lab,
P.O.B. 100565, 98684 Ilmenau, Germany

{stephan.kirstein,horst-michael.gross}@tu-ilmenau.de

² Honda Research Institute Europe GmbH,
Carl-Legien-Str. 30 63073 Offenbach am Main, Germany
{heiko.wersing,edgar.koerner}@honda-ri.de

Abstract. We present a category learning vector quantization (cLVQ) approach for incremental and life-long learning of multiple visual categories where we focus on approaching the stability-plasticity dilemma. To achieve the life-long learning ability an incremental learning vector quantization approach is combined with a category-specific feature selection method in a novel way to allow several metrical “views” on the representation space for the same cLVQ nodes.

1 Introduction

The target of our proposed architecture is to perform supervised, interactive, and incremental life-long learning of several visual categories by combining incremental learning of learning vector quantization (LVQ) nodes with a category-specific forward feature selection method. Additionally we are approaching the so-called “stability-plasticity dilemma”, which occurs when neural networks are trained with a limited and changing training ensemble, causing the well known “catastrophic forgetting effect” [1]. A common strategy for life-long learning architectures (e.g. [3,7]) is the usage of an individual node learning rate combined with incremental node insertion. This permits plasticity of newly inserted neurons, while the stability of matured neurons is preserved. The major drawback of those architectures commonly used for classification tasks is the lack of separating cooccurring visual categories. This means for natural objects which typically belong to several different categories (e.g. red-white car) a decoupled representation for each category (for category red, white and car) should be learned. This leads to a compact representation and higher generalization performance compared to classification, which can’t be achieved with standard life-long learning architectures. Another approach to the “stability-plasticity dilemma” was proposed by [10]. Here representative input-output pairs are stored into a long term memory for stabilizing an incremental radial basis function (RBF) like network. Additionally it also accounts for a feature selection mechanism based on incremental principal component analysis, but no category-specific feature selection is applied, which makes it unsuitable for categorization tasks without modification.

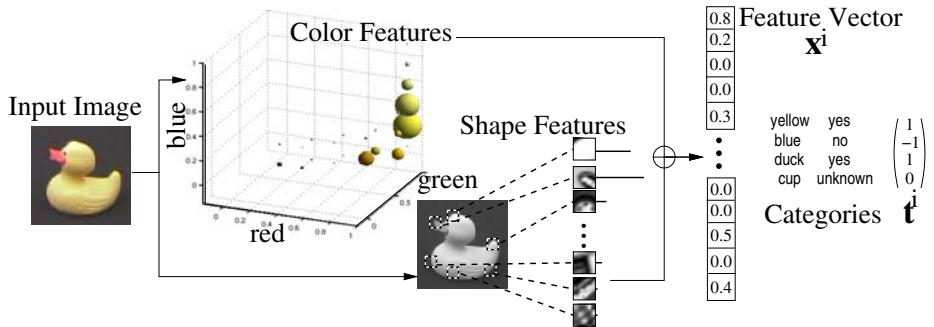


Fig. 1. Feature extraction. Color features are extracted as histogram bins in RGB space. Shape features are obtained from parts-based feature detector. Shape and color features are concatenated into a single “flat” vector representation. The target categories are represented in a category vector \mathbf{t}^i for each feature vector \mathbf{x}^i .

The manuscript is structured as follows: in Section 2 we describe the feature extraction model using shape and color features, and introduce our category learning architecture. We show the application to a visual categorization task in Section 3 and discuss results and related work in Section 4.

2 Memory Architecture for Learning of Visual Categories

2.1 Feature Extraction

Color Features. For the representation of color information we use the common histogram binning method combining robustness against view and scale changes with computational efficiency [11]. We use $6 \times 6 \times 6 = 216$ histogram bins within the RGB space, where typically only a small amount of features is active.

Shape Features. Our shape features are a set of preselected SIFT-descriptors defining parts-based detectors as proposed by [5]. For each new object view the response of those detectors is calculated at each location in the image using the dot product as similarity measure. The maximum response per feature detector is kept and stored in an activity vector, which neglects all spatial information. The offline feature selection scheme follows the approach described in [5], where all SIFT-descriptors of each training image are clustered into 100 components. Out of the large number of resulting clusters an iterative scheme selects at each step a SIFT-descriptor as new detector until a given number is reached (e.g. 500 in our case). The choice of detectors is independent from our category learning method and is based on the highest additional gain for a certain shape category.

Combined Feature Representation. For our categorization task the color histogram and parts-based shape representation are combined into a single structureless feature vector $\mathbf{x}^i = (x_1^i, \dots, x_F^i)$ for each image with $F = 716$. Each vector \mathbf{x}^i is assigned to a list $\mathbf{t}^i = (t_1^i, \dots, t_C^i)$ of several color and shape categories, where each $t_c \in \{-1, 0, +1\}$ labels a \mathbf{x}^i as positive or negative example of category c .

The third state $t_c = 0$ means unknown category membership and is required, because we do not assume that all category labels are provided by the tutor. Due to the nature of our feature preprocessing the data in the \mathbf{x}^i is sparse and non-negative. For our categorization task we are particularly interested in discovering the structure from the high-dimensional feature vectors \mathbf{x}^i . Therefore we do not give the qualitative separation of the extracted features to the learning system as a priori information, but rather want to obtain a flexible metrical adaptation for the categorization. Assume you want to learn the category “fire engine”. If only shape features are used it would be difficult to distinguish this category from other cars and trucks, because the most distinctive feature, the red color, is not included in the feature representation. Therefore we let the learning algorithm decide which feature combinations are most suitable to represent a category.

2.2 Memory Architecture for Learning Visual Categories

Our memory architecture is based on a forward feature selection method combined with an incremental learning exemplar-based network to allow life-long learning of several visual categories. Both parts are optimized together to find a balance between insertion of features and allocation of nodes, while using as few resources as possible, which is crucial for interactive and online learning with respect to the required computation time. In the following we refer to this architecture as category learning vector quantization (cLVQ). The used wrapper method for category-specific forward feature selection enables the separation of cooccurring categories, because it defines category specific metrical “views” on the nodes of the exemplar-based network. There are three groups of feature selection methods (see [2] for an overview). The first group are filter methods, where subsets of features are selected as a preprocessing step, independently of the chosen classifier architecture. The second group are wrapper methods, as used in our memory architecture. Here the methods use the learning architecture as a black box, to score different feature subsets, but are independent of the learning architecture. The last group are embedded methods where the feature selection process is an integrated part of the learning architecture, typically realized with sparsity constraints added to the error function.

Exemplar-Based Network as Memory Architecture. The exemplar-based network part of our memory architecture is motivated from the iLVQ [7] and is extended to deal with categorization tasks.

We denote the incrementally built up set of cLVQ representative vectors \mathbf{w}^k as $W = \{\mathbf{w}^k\}_{k=1,\dots,W_n}$. Each \mathbf{w}^k has attached an label vector \mathbf{u}^k where $u_c^k \in \{-1, 1, 0\}$ is the model target output for category c , representing positive, negative, and missing label output, respectively. Each cLVQ node \mathbf{w}^k is therefore assigned to a vector \mathbf{u}^k of several categories. For an input \mathbf{x}^i the output is determined for each category by the winning node $\mathbf{w}^{k_{\min}(c)}$ with $k_{\min}(c) = \arg \min_{k \in A_c} (d_c(\mathbf{w}^k, \mathbf{x}^i))$, where the minimum is only determined for representatives in the set $A_c = \{k | u_c^k \neq 0\}$ of known category-labeled vectors. The final model output is then given as $o_c^i(\mathbf{x}^i) = u_c^{k_{\min}(c)}$.

A key element in the cLVQ architecture is the adaptive category-specific distance computation d_c that is strongly dependent on the integrated incremental feature selection process. We use a weighting of Euclidean dimensions [4] with specific weight factors λ_{cf} according to:

$$d_c(\mathbf{x}^i, \mathbf{w}^k) = \sum_{f=1}^F \lambda_{cf} (x_f^i - w_f^k)^2. \quad (1)$$

The category-specific weights λ_{cf} are updated incrementally, which we describe in the following sections in more detail. We denote the set of active categories as $C = \{c\}_{c=1,\dots,C_n}$, and the set of active features for an active category $c \in C$ as S_c . We choose $\lambda_{cf} = 0$ for all $f \notin S_c$, and otherwise adjust it according to a scoring procedure explained later. The winning nodes $\mathbf{w}^{k_{\min}(c)}(\mathbf{x}^i)$ are calculated independently for each category c . Each $\mathbf{w}^{k_{\min}(c)}(\mathbf{x}^i)$ is updated based on the standard LVQ learning rule [8], but is restricted to feature dimensions $f \in S_c$:

$$w_f^{k_{\min}(c)} := w_f^{k_{\min}(c)} + \mu \Theta^{k_{\min}(c)} (x_f^i - w_f^{k_{\min}(c)}) \quad \forall f \in S_c, \quad (2)$$

where $\mu = 1$ if the categorization decision for \mathbf{x}^i was correct, otherwise $\mu = -1$ and the winning node $\mathbf{w}^{k_{\min}(c)}$ will be shifted into the opposite direction as \mathbf{x}^i . Additionally Θ^k is the node-dependent learning rate as proposed in [7].

The learning dynamics of the cLVQ memory architecture is organized in training epochs composed of the following steps:

Step 0: Training Objects. For each epoch only a limited number of objects are visible to the architecture, emulating a limited short term memory (STM). All learning in the cLVQ network is based on the training feature inputs $X = \{\mathbf{x}^i\}_{i=1,\dots,X_n}$ and target category values $\{\mathbf{t}^i\}_{i=1,\dots,X_n}$ of these object views.

Step 1: Feature Scoring Update. For the later feature selection in Step 3 we compute for each feature f and available category c a score $r_{cf} = M_{fc}/(M_{fc} + \bar{M}_{fc})$, where M_{fc} and \bar{M}_{fc} are the number of previously seen positive and negative training examples respectively. For each new training example the values are updated with $M_{fc} := M_{fc} + 1$ for all \mathbf{x}^i with $t_c^i = +1$ and $x_f^i > 0$ and $\bar{M}_{fc} := \bar{M}_{fc} + 1$ if $t_c^i = -1$ and $x_f^i > 0$. The score defines the metrical weighting in the cLVQ representation space. We thus choose $\lambda_{cf} = r_{cf}$ for all $f \in S_c$ and $\lambda_{cf} = 0$ otherwise.

Step 2: Initialization of Categories. If a category c with training vectors \mathbf{x}^i and corresponding category label $t_c^i = +1$ occurs the first time, we initialize this category c with a single feature and one cLVQ node. We select the feature $v_c = \arg \max_f(r_{cf})$ with the largest score value and initialize $S_c = \{v_c\}$. As the initial cLVQ node for category c we select the training vector \mathbf{x}^i , where the selected feature v_c is highest activated, i.e. $\mathbf{w}^{W_n+1} = \mathbf{x}^j$ with $x_{v_c}^j > x_{v_c}^i$ for all i . The attached label vector is chosen as $u_{v_c}^{W_n+1} = 1$ and zero for all other vector entries.

Step 3: Feature Testing. The target of this step is the addition or removal of features for the category-specific metrics, based on the observable training set

errors. For each category c we determine the set of positive errors $E_c^+ = \{i | t_c^i = 1 \wedge t_c^i \neq o_c^i\}$ and negative errors $E_c^- = \{i | t_c^i = -1 \wedge t_c^i \neq o_c^i\}$. If $\#E_c^+ > \#E_c^-$ then we compute $l_{cf}^+ = \sum_{i \in E_c^+} H(x_f^i) / \sum_{i \in E_c^+} 1$, where H is the Heaviside function.

The score l_{cf}^+ is the ratio of active feature entries for feature f among the positive training errors of class c . We want to add now a feature to the category feature set S_c , which both contributes to the class c in the training data and is very active for the encountered error set E_c^+ . Therefore we choose $v_c = \arg \max_{f \notin S_c} (l_{cf}^+ + r_{cf})$ and add $S_c := S_c \cup \{v_c\}$. The added feature dimension modifies the cLVQ metrics and we can now compute the total categorization error on the training set before and after the change. If the performance increase for category c is larger than a threshold ϵ , then v_c is permanently added. Otherwise it is removed and excluded for further training iterations of this epoch. An analog step is performed, if the number of negative errors is larger than the number of positive errors ($\#E_c^+ < \#E_c^-$), with the difference that the feature is removed and then again the performance gain is computed for the final decision on the removal.

Step 4: cLVQ Node Testing. Similar to Step 3 we test new cLVQ nodes only for erroneous categories. In contrast to the node insertion rule proposed in [7], where nodes are inserted for training vectors with smallest distance to wrong winning nodes, we propose to insert new cLVQ nodes based on training vectors x^i with most categorization errors over all categories, until for each erroneous category c at least one new node is inserted. The corresponding attached label output vector $\mathbf{u}^k = \mathbf{t}^i$ is only filled at category positions, where category informations are available and is otherwise zero. This leads to a more compact representation, because a single node typically improves the representation of several categories.

Again we calculate the performance increase based on all currently available training vectors. If this increase for category c is above the threshold ϵ , we make no modifications to cLVQ node labels of the corresponding newly inserted nodes. Otherwise we set the corresponding labels u_c^k of the newly inserted nodes \mathbf{w}^k to zero and remove nodes where all u_c^k are zero.

Step 5: Stop condition. If all remaining categorization errors are resolved or all possible features f of erroneous categories c are tested, go to Step 0 and start the next training epoch. Otherwise go to Step 3 and test further feature candidates and cLVQ nodes.

3 Experimental Results

3.1 Object Ensemble

For evaluating our cLVQ architecture we use an image ensemble composed of 56 different training objects and 56 distinct objects for testing (see Fig. 2), which were never used during the training phase. For each object 300 views are taken in front of a black background while rotating it around the vertical axis.

Overall our object ensemble contains ten different shape categories and five different color categories (see Fig. 2). It should be mentioned that several objects



Fig. 2. Examples of training (left) and test objects (right) used for our categorization task, where 15 different categories are trained. As color categories red, green, blue, yellow and white are trained. The shape categories are animal, bottle, box, brush, can, car, cup, duck, phone, tool. Each object was presented in front of a black background while rotating around the vertical axis (bottom), resulting in 300 images per object.

are multi-colored (e.g. the cans) where not only the base color should be detected, but also all other prominent colors. This multi detection constraint complicates the categorization task compared to the case where only the best matching category or the best matching category of a specified group of visual attributes (e.g. one for color and one for shape) must be detected.

3.2 Categorization Results

We compare our proposed life-long learning architecture cLVQ with a single layer perceptron (SLP) and a simplified version cLVQ*. The SLP output for each category is given as $o_c^{slp}(\mathbf{x}^i) = \tanh(w_f^c * \mathbf{x}^i - \theta_c)$, where \mathbf{w}^c is a single linearly separating weight vector with threshold θ_c for each c . Training of the SLP consists of standard stochastic gradient descent in the sum of quadratic difference errors between training target and model output. In contrast to the more common used ROC curves we estimate the rejection thresholds during the learning process, based on the average activation strength. This is required to allow categorization of new object views at any time which is an essential requirement for interactive learning tasks. We refer to cLVQ* as a modified version of cLVQ, where all features without dynamic feature weighting are used, allowing conclusions about the effect of feature selection and weighting. Additionally the node performance gain threshold ϵ was set to zero. For all experiments summarized in Fig. 3 we trained the different network architectures with a limited and changing training ensemble composed of a visible “window” of only three objects, to test the life-long learning ability of our cLVQ architecture. For each epoch only three objects are visible to the learning algorithm, where at the beginning of each epoch a randomly selected object is added, while the oldest one is removed. This scheme is repeated until all 56 training objects are presented once to the network architectures.

Although no prior information is given with respect to the kind of categories, we distinguish for the performance measurement between color and shape categories to discuss the different quality of extracted features. The overall performance of the cLVQ architecture as shown in Fig. 3 is high for both feature

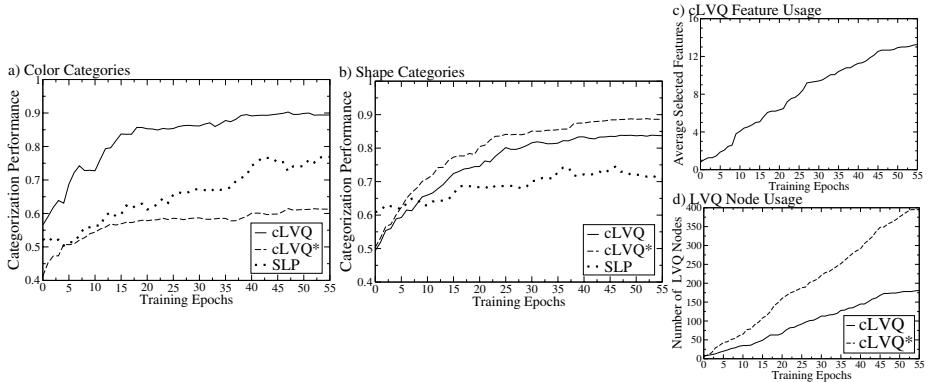


Fig. 3. Comparison of categorization performance for SLP, cLVQ* and cLVQ, averaged over five runs. The categorization performance is calculated after each training epoch, based on all test objects. This means that the performance is calculated based on the representation of the so far seen objects, simulating an interactive learning session. a) For the color categories the cLVQ algorithm performed better compared to the cLVQ*. The SLP performance is in between both LVQ approaches. b) Compared to the color categories the cLVQ* is slightly better than cLVQ for shape categories, while the SLP performed worst. c) Shows the number of selected features, averaged over all categories, while d) shows the allocation of cLVQ nodes during training.

modalities. For the color categories it performs much better than the simpler cLVQ*, while for the shape categories the cLVQ* performs slightly better. In general it can be said, that the feature selection method, typically selecting less than 5% of all available feature dimensions $F = 716$ (see Fig. 3c), is able to capture the category information and also strongly reduces the necessary amount of required resources, which is crucial for interactive learning. Despite the fact that cLVQ typically requires only half of the nodes compared to cLVQ* as shown in Fig. 3d it is even about 100 times faster. The cLVQ algorithm is even a few times faster than the SLP network, making it well suited for interactive learning.

The SLP performance is surprisingly high for all trained categories. This is somehow contradictory to classification tasks, with a one-out-of-n class selection, where the SLP approach is known for the “catastrophic forgetting effect” [1]. For our categorization task this effect is only slightly visible for the shape categories. Although the forgetting effect is less present in our categorization task the SLP approach is still considerably worse than cLVQ for both feature modalities.

4 Discussion

We propose a architecture for life-long learning of visual categories, able to perform an automatic feature selection, feature weighting and incremental allocation of cLVQ nodes, which is suitable for interactive learning. The wrapper method for automatic feature selection is mainly used because the representation of categories should use as few feature dimensions as possible, which can not be achieved

with simple filter methods, where typically only a small amount of redundant or noisy features are eliminated. There are metric adaptation methods for LVQ networks like the GRLVQ proposed by [4] which was used by [6] as embedded method for feature selection. Those metric adaptation methods are designed for feature weighting but lack the ability to separate cooccurring visual attributes and are therefore not applicable to categorization tasks. Especially this separation capability of cooccurring categories combined with fast interactive and incremental learning makes CLVQ beneficial for categorization tasks.

In comparison to many other categorization approaches we are able to learn multiple categories at once, while commonly the categories are trained individually. Additionally many computer vision approaches dealing with categorization like the implicit shape models (ISM) proposed in [9] only uses the canonical views of the category (e.g. only side views of cars), while we rotate the objects around the vertical axis, which causes much higher appearance changes. This illustrates that our exemplar-based method can deal with a larger within-category variation, which we consider crucial for complex categories.

References

- French, R.M.: Catastrophic Forgetting in Connectionist Networks: Causes, Consequences and Solutions. *Trends in Cognitive Sciences* 3(4), 128–135 (1999)
- Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
- Hamker, F.H.: Life-long learning Cell Structures - Continuously Learning without Catastrophic Interference. *Neural Networks* 14, 551–573 (2001)
- Hammer, B., Villmann, T.: Generalized Relevance Learning Vector Quantization. *Neural Networks* 15, 1059–1068 (2002)
- Hasler, H., Wersing, H., Körner, E.: A Comparison of Features in Parts-based Object Recognition Hierarchies. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4669, pp. 210–219. Springer, Heidelberg (2007)
- Kietzmann, T.C., Lange, S., Riedmiller, M.: Incremental GRLVQ: Learning Relevant Features for 3D Object Recognition. *Neurocomputing* (2007) (in press)
- Kirstein, S., Wersing, H., Körner, E.: A Biologically Motivated Visual Memory Architecture for Online Learning of Objects. *Neural Networks* 21, 65–77 (2008)
- Kohonen, T.: Self-Organizing and Associative Memory, 3rd edn. Springer Series in Information Sciences. Springer, Heidelberg (1989)
- Leibe, B., Leonardis, A., Schiele, B.: Combined Object Categorization and Segmentation with an Implicit Shape Model. In: ECCV 2004 Workshop on Statistical Learning in Computer Vision (2004)
- Ozawa, S., Toh, S.L., Abe, S., Pang, S., Kasabov, N.: Incremental Learning of Feature Space and Classifier for Face Recognition. *Neur. Netw.* 18, 575–584 (2005)
- Swain, M.J., Ballard, D.H.: Color Indexing. *International Journal of Computer Vision* 7(1), 11–32 (1991)

An Integrated System for Incremental Learning of Multiple Visual Categories

Stephan Kirstein^{1,2}, Heiko Wersing², Horst-Michael Gross¹, and Edgar Körner²

¹ Ilmenau University of Technology,

Neuroinformatics and Cognitive Robotics Lab,

P.O.B. 100565, 98684 Ilmenau, Germany

{stephan.kirstein,horst-michael.gross}@tu-ilmenau.de

² Honda Research Institute Europe GmbH,

Carl-Legien-Str. 30 63073 Offenbach am Main, Germany

{heiko.wersing,edgar.koerner}@honda-ri.de

Abstract. We present a biologically inspired vision system able to incrementally learn multiple visual categories by interactively presenting several hand-held objects. The overall system is composed of a foreground-background separation part, several feature extraction methods and a life-long learning approach combining incremental learning with category specific feature selection. In contrast to most visual categorization approaches where typically each view is assigned to a single category we allow labeling with an arbitrary number of shape and color categories and also impose no restrictions to the viewing angle of presented objects.

1 Introduction

An amazing capability of the human visual system is the ability to learn an enormous repertoire of visual categories. This large amount of categories is acquired incrementally during our life and requires at least at the beginning the direct interaction with a tutor. Inspired by child-like learning we propose an architecture for learning several visual categories in an incremental and interactive fashion. The architecture is composed of several building blocks including segmentation, feature extraction, a category learning module and user interaction, which allow training of categories based on natural hand-held objects.

The development of online and life-long learning systems became more and more popular in the recent years e.g. [9], [11] or [14]. The work of [9] allows online learning and detection of hand-held objects in cluttered scenes based on a combination of a background model and a tracking method but is restricted to static camera settings like security cameras. Of particular interest is the work proposed by [11], because it targets for a similar interactive category learning task as investigated in this paper but relies on a simple learning method which can only be applied to categories with little appearance changes.

The learning system proposed in this paper is related to earlier work that enables object recognition of complex shaped objects presented by hand in cluttered scenes [14]. Especially the preprocessing steps are therefore similar, but

several modifications to the feature extraction, the learning method and the user interaction were necessary to allow the same functionality for learning categories based on natural objects. Natural objects typically belong to several different categories (e.g. red-white car), therefore a decoupled representation for each category (for category red, white and car) is required, which can not be handled by typical incremental learning systems dealing with classification tasks. This decoupling leads to a more condensed representation and higher generalization performance compared to classification architectures. Additionally several modifications with respect to the figure-ground segregation were applied.

Since several years many architectures dealing with object detection and categorization tasks have been proposed in the computer vision community. Interestingly most of these approaches are only based on local parts-based features, which are extracted around some defined interest points (e.g. implicit shape models (ISM) [8]) to build up object models for categories like faces or cars. The advantages of such models are their robustness against partial occlusion and scale changes, but also the ability to deal with clutter. The main drawback of these architectures is the restriction to the canonical view of a certain category, while objects in a natural environment usually occur in many different orientations. Typically such architectures also require long training phases to generate the object models, which make them unsuitable for interactive training. A recent work of [2] tries to overcome this speed limitation and proposes a combined approach of ISM [8] and a semi-supervised clustering algorithm which enables to incrementally build up object categories based on dialog interactions with a human teacher. Although the general approach is interesting because it minimizes the necessary interaction with a tutor, it still does not allow learning of categories from arbitrary viewpoints.

The manuscript is structured as follows: in Section 2 we describe the building blocks of our category learning system and show its ability to incrementally learn several visual categories in Section 3. Finally we discuss the results and related work in Section 4.

2 Incremental Category Learning System

In the following we describe the building blocks of our learning system (see Fig. 1) composed of preprocessing, figure-ground segregation, and several feature extraction methods providing shape and color information. The core part of our category learning system is a life-long learning vector quantization method which is trained in direct interaction with a human tutor.

2.1 Preprocessing and Figure-Ground Segregation

The input to our category learning system is a stream of image pairs taken by a camera head with pan-tilt unit and parallel aligned cameras. Depth information is calculated after the rectification of both camera images to correct lens distortion. This depth information is used to generate an object hypothesis in

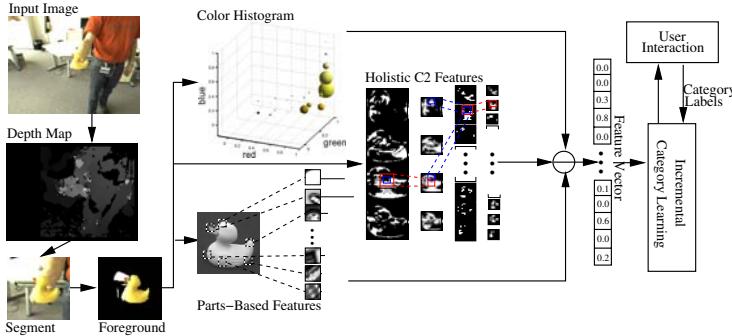


Fig. 1. Category Learning System. Based on an object hypothesis extracted from the depth map a figure-ground segregation is performed. The detected foreground is used to extract color and shape features. Color features are extracted as histogram bins in RGB space. In contrast to most other categorization approaches we combine holistic features obtained with a detection hierarchy and parts-based features based on SIFT-descriptors. All extracted features are concatenated into a single structureless vector which, together with the category labels provided by a human tutor, is the input of the incremental category learning module.

cluttered scenes, where we assume that things in the near range of the camera system are of particular interest for the system. The result of the preprocessing is a region of interest (ROI) used to localize and track segments in the scene.

This extracted segment still contains a substantial amount of background clutter, but for the incremental build up of category representations it is beneficial if such clutter is suppressed. Therefore we apply an additional figure-ground segregation as proposed by [1] to reduce this influence. The basic idea of this segregation method is to use learning vector quantization (LVQ) to learn a pre-defined number of distinct prototypes for foreground and background where the noisy depth information is used as initial hypothesis. The learning of these prototypes is based on different feature maps consisting of RGB-color features as well as the pixel positions. Instead of the standard Euclidean metrics, for the distance computation additional prototype specific relevance factors are calculated based on generalized matrix LVQ [10]. These relevance factors are adapted online and dynamically weight the different maps to maximize the margin between foreground and background. The output of this segregation step is a binary mask defining the foreground. In the following processing steps only foreground pixels are used to extract category specific features.

2.2 Feature Extraction

Color Features. For the representation of color information we use the common histogram binning method which combines robustness against view and scale changes with computational efficiency [12]. Overall $6 \times 6 \times 6 = 216$ histogram bins within the RGB space are used, where typically a small amount of features are specific for a complete color category.

Shape Features. The shape features are obtained by a hierarchical feed-forward architecture and parts-based feature detectors. The feature detectors of the hierarchical feed-forward architecture are obtained by unsupervised learning, providing a set of general but less category-specific features, while the parts-based features are trained supervised with respect to category specificity. We combine these different shape features to show the ability of the category learning method to select appropriate features out of a large amount of possible candidates. Such feature combinations are rare because most categorization methods rely on parts-features only, but in offline experiments we recognized an increase in categorization performance, when both extraction methods are combined compared to using them individually.

The hierarchical feed-forward architecture is based on weight-sharing and a succession of feature detection and pooling stages (see [13] for details). The first feature-matching layer S1 is composed of four orientation sensitive Gabor filters. Additionally a Winner-Take-Most mechanism between features at the same position and a final threshold function is applied. The following C1 layer subsamples the S1 features by pooling down to a quarter of the original resolution in both directions using a Gaussian receptive field and a sigmoidal nonlinearity. The 50 features in the intermediate layer S2 are obtained by sparse coding and are sensitive to local combinations of the features from the C1 layer. The layer C2 again performs spatial integration and reduces the resolution to a half in each direction, resulting in 50 18x18 sparsely activated feature maps with a total dimensionality of 16200, where typically less than 10% of all features are non-zero.

As parts-based feature detectors a set of preselected SIFT-descriptors is used as proposed by [4]. For each new object view the response of those detectors is calculated at each location in the image using the dot product as similarity measure. The maximum response per feature detector is kept and stored in an activity vector, which neglects all spatial information. The offline feature selection scheme follows the approach described in [4], where all SIFT-descriptors of each training image are clustered into 100 components. Out of the large number of resulting clusters an iterative scheme selects at each step a SIFT-descriptor as new detector until a given number is reached (e.g. 100 in our case). The choice of detectors is based on the highest additional gain for a certain shape category.

Combined Feature Representation. All extracted features of an image are combined into a single structureless and sparsely activated feature vector $\mathbf{x}^i = (x_1^i, \dots, x_F^i)$, with resulting feature space dimensionality of $F = 16516$. The task of the category learning method, described in the next section, is to automatically select a category specific feature subset, which best represents the category without the given knowledge which features contain color or shape information.

2.3 Incremental and Interactive Category Learning

The learning of visual categories is based on a limited and changing set of labeled training vectors, which are stored into a short term memory (STM). The category learning method must be able to extract the category informations from the STM

and conserve this information in the long term memory (LTM) representation. To achieve this transfer an incremental exemplar-based network is combined with a forward feature selection method. This allows life-long learning and enables a separation of cooccurring visual categories based on selected category-specific feature sets, which most exemplar-based networks can not handle.

Short Term Memory. This memory type is similar to the online learning vector quantization method developed earlier [5]. In contrast to the naive approach, where each view is stored in the STM a similarity calculation is performed based on all vectors with identical category label list. New views are only added to the STM if the similarity to such vectors is below a specified insertion threshold S_T . Based on this simple selection schema it could be shown [5], that the number of training views can be reduced by about 30% without losing generalization performance, and reducing the LTM training time. Additionally we assume a limited memory size of the STM, which requires a deletion heuristic of feature vectors if the capacity limit is reached. Therefore STM vectors are deleted which belong to the same category label list and for which almost no categorization errors occur. Such vectors are already successfully transferred to the LTM and can be deleted without information loss.

Long Term Memory. The general idea of the category LVQ (cLVQ) method is to iteratively make small changes to the representation of erroneous categories. After a change the performance gain is calculated, based on all available training vectors. If the increase is larger than a threshold ϵ the feature or LVQ node is permanently added to the representation and otherwise removed.

For guiding the feature testing we use a statistical feature scoring method (see [3] for an introduction to feature selection methods) as proposed in [6], where a single scalar r_{fc} for each category c and feature f is calculated to estimate the category specificity. Similar to the development of children, the feature scoring is only based on previously seen exemplars of a certain category and can strongly change if further information is encountered. Therefore we continuously update the feature scoring values to follow this change and also allow in rare cases the deletion of selected features. The feature testing itself selects predominately those features which occur almost exclusively for a certain category and also are often present in the most categorization errors. In parallel to the incremental selection of features also new LVQ nodes are inserted and tested. For the node insertion we propose to insert new LVQ nodes based on the training vectors with most categorization errors. This typically leads to an improvement for several categories and generates a compact representation, which is a major requirement for fast and interactive learning.

For the LTM representation each cLVQ node \mathbf{w}^k is assigned to a vector $\mathbf{u}^k = (u_1^k, \dots, u_C^k)$ of arbitrary color and shape categories, where each $u_c \in \{-1, 0, +1\}$ labels a \mathbf{w}^k as positive or negative example of a category. The third state $u_c = 0$ means unknown category membership and is required to incrementally resolve categorization errors, while imposing no representational changes to error free categories. For the calculation of winning node $\mathbf{w}^{k_{\min}(c)}(\mathbf{x}^i)$ for category c the

Euclidean distance computation is combined with dynamic metric adaptation, based on the obtained feature scoring values:

$$\|\mathbf{x}^i - \mathbf{w}^k\|_{\lambda_c}^2 = \sum_{f=1}^F \lambda_{cf} (x_f^i - w_f^k)^2, \quad (1)$$

where $\lambda_{cf} = r_{fc}$ for all features f selected for category c , while all other $\lambda_{cf} = 0$ and thus have no influence on the distance computation. For the adaptation of the winning node $\mathbf{w}^{k_{\min}(c)}(\mathbf{x}^i)$ of category c , the standard LVQ learning rule [7] is used, but is restricted to the selected feature dimensions of category c . Additionally the learning rate is dependent on the node age, which allows strong modification of newly inserted nodes, while the representation of well-adapted nodes is conserved.

User Interaction. For interactively providing label information to the STM and LTM we use a simple state-based user interface. This user interface is based on a list of predefined labels, including some wild card labels, to allow the labeling of categories, for which no category label is defined. All labels can be provided to the system in any arbitrary order. In general the user interaction is composed of two operation modes. The default user interaction mode is that the learning system generates a hypothesis list of currently present categories, which afterward can be corrected or confirmed by the user. The other possibility is that the user directly provides category labels, to label previously unknown categories.

3 Experimental Results

For all experiments several complex shaped objects are freely rotated in front of our camera system. Based on the extracted features and the current category representation in the LTM the category decision which categories are currently detected is communicated to the user. This hypothesis generation is repeated until the user confirms or corrects the categorization decisions of the LTM representation, which then triggers the collection of new training vectors into the STM. This means our category learning method has two operation states, where in the one state it produces category hypotheses and in the other state, when confirmed category labels are available, it collects new training vectors in the STM, which later are transferred into the LTM representation. The incremental learning of the LTM representation is performed in both states and is even continued, if currently no new object views are presented, because this knowledge transfer typically takes much longer than the STM training.

Overall our system is distributed on three different 3 GHz CPUs running at a frame rate of 8-10 Hz, which is fast enough to show the desired incremental and life-long learning ability of our system. We consider two different scenarios summarized in Fig. 2. For the first scenario, we start with a blank STM and LTM representation, where we show how efficient the training of categories can be done based on only few representatives of the corresponding category, typically requiring less than 10 min training time. At this state the learning system is able

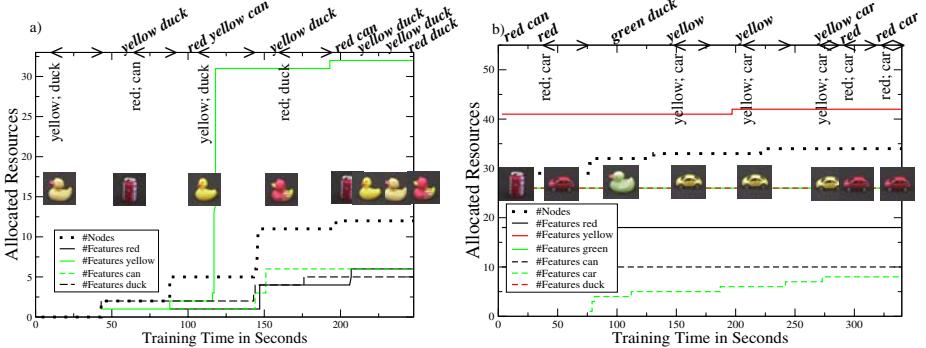


Fig. 2. Incremental learning of visual categories. The formation of selected features and allocated nodes are shown over time, while presenting different objects for starting with a) blank memory and b) at a later learning stage. It can be seen how many cLVQ nodes (dotted lines) are allocated over time, while additionally the change of selected features are shown for each known category (dashed and solid lines). We also added the categorization decisions of the learning system, communicated to the user on top of the figure with sloped text, while the confirmed category labels provided by the user are denoted underneath, separated by semicolons. Additionally the intervals where new training vectors are collected into the STM are marked with <>.

to detect the categories reliably for objects already presented to the system, but typically the generalization to completely new objects is quite poor. For the second scenario we used the representation after about one hour training time, where in the mean time more representatives of the trained categories were shown, which strongly improved the representation and also the generalization performance. Here it should be mentioned that even after more elaborate training the number of allocated features for each category is still small compared to the overall number of extracted features, which is a basic requirement for interactive training. Additionally we show that adding new categories to the representation can be done in a flexible way, without affecting the already known categories.

4 Discussion

We have presented a learning system able to interactively learn arbitrary visual categories in a life-long learning fashion. To our knowledge this is the first category learning system which allows category learning based on complex shaped objects held in the hand. Comparable architectures as proposed by [1] or [2] learn categories based on objects placed on a table, which simplifies the ROI detection and figure-ground segregation. Additionally it also strongly reduces the appearance variations of the presented objects and therefore makes the category learning task much easier. We also allow different categories for a single object, while typically the categories are trained independently.

We also could show that our learning system can efficiently perform all necessary processing steps including figure-ground segregation, feature extraction

and incremental learning. Especially the ability to handle high dimensional but sparse feature vectors is necessary to allow interactive and incremental learning, where often additional dimension reduction techniques like the PCA are required to allow online learning. This high feature dimensionality is also challenging for the used feature selection method, because of the large amount of possible feature candidates, but still the learning system is able to extract small sets of category specific features out of many possible feature candidates.

Acknowledgments. We thank S. Hasler and A. Denecke for providing the parts-based feature extraction and figure-ground segregation method. Additionally we thank both for the assistance of integrating their work into the system.

References

1. Denecke, A., Wersing, H., Steil, J.J., Körner, E.: Robust Object Segmentation by Adaptive Metrics in Generalized LVQ. In: Proc. ESANN, pp. 319–324 (2008)
2. Fritz, M., Kruijff, G.-J.M., Schiele, B.: Cross-Modal Learning of Visual Categories using Different Levels of Supervision. In: Proc. ICVS Conference (2007)
3. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
4. Hasler, H., Wersing, H., Körner, E.: A Comparison of Features in Parts-based Object Recognition Hierarchies. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4669, pp. 210–219. Springer, Heidelberg (2007)
5. Kirstein, S., Wersing, H., Körner, E.: A Biologically Motivated Visual Memory Architecture for Online Learning of Objects. *Neural Networks* 21, 65–77 (2008)
6. Kirstein, S., Wersing, H., Gross, H.M., Körner, E.: A Vector Quantization Approach for Life-Long Learning of Categories. In: ICONIP 2008 (2008) (submitted)
7. Kohonen, T.: Self-Organizing and Associative Memory. Springer Series in Information Sciences, 3rd edn. Springer, Heidelberg (1989)
8. Leibe, B., Leonardis, A., Schiele, B.: Combined Object Categorization and Segmentation with an Implicit Shape Model. In: ECCV 2004 Workshop on Statistical Learning in Computer Vision (2004)
9. Roth, P.M., Donoser, M., Bischof, H.: On-line Learning of Unknown Hand Held Objects via Tracking. In: Proc. Int. Conf. on Computer Vision Systems (2006)
10. Schneider, P., Biehl, M., Hammer, B.: Relevance Matrices in LVQ. In: Similarity-based Clustering and its Applications to Medicine and Biology (2007)
11. Skočaj, D., Berginc, G., Ridge, B., Štímec, A., Jogan, M., Vanek, O., Leonardis, A., Hutter, M., Hawes, N.: A System for Continuous Learning of Visual Concepts. In: Proc. ICVS (2007)
12. Swain, M.J., Ballard, D.H.: Color Indexing. *Int. J. of Computer Vision* 7(1), 11–32 (1991)
13. Wersing, H., Körner, E.: Learning Optimized Features for Hierarchical Models of Invariant Object Recognition. *Neural Computation* 15(7), 1559–1588 (2003)
14. Wersing, H., Kirstein, S., Goetting, M., Brandl, H., Dunn, M., Mikhailova, I., Goerick, C., Steil, J.J., Ritter, H., Körner, E.: Online Learning of Objects in a Biologically Motivated Architecture. *Int. J. of Neural Systems* 17, 219–230 (2007)

A Neural Network Model for Sequential Multitask Pattern Recognition Problems*

Hitoshi Nishikawa¹, Seiichi Ozawa¹, and Asim Roy²

¹ Graduate School of Engineering, Kobe University, Kobe 657-8501, Japan

075t247t@stu.kobe-u.ac.jp, ozawasei@kobe-u.ac.jp

² Arizona State University, Tempe, Arizona 85048, USA

ASIM.ROY@asu.edu

Abstract. In this paper, we propose a new multitask learning (MTL) model which can learn a series of multi-class pattern recognition problems stably. The knowledge transfer in the proposed MTL model is implemented by the following mechanisms: (1) transfer by sharing the internal representation of RBFs and (2) transfer of the information on class subregions from the related tasks. The proposed model can detect task changes on its own based on the output errors even though no task information is given by the environment. It also learns training samples of different tasks that are given one after another. In the experiments, the recognition performance is evaluated for the eight MTPR problems which are defined from the four UCI data sets. The experimental results demonstrate that the proposed MTL model outperforms a single-task learning model in terms of the final classification accuracy. Furthermore, we show that the transfer of class subregion contributes to enhancing the generalization performance of a new task with less training samples.

1 Introduction

Most machine learning researches have focused on the learning of a single task. Thinking about our daily life, however, humans are performing and learning multiple tasks in a parallel or sequential way. For example, when we meet a human and see the face, we may first know the age, sex, and other features of the person from the face image. Then, we may try to identify the name, health condition, etc. Therefore, an object in the real-world can have multiple descriptions, and we have to select a right description *pro re nata* in the recognition process. Learning multiple descriptions from a single object gives a new direction to pattern recognition problems and provides a key technology in building a highly intelligent system like smart robots. This type of pattern recognition problems is called *multitask pattern recognition* (MTPR) problems [1].

* The authors would like to thank Professor Shigeo Abe for his helpful comments and discussions. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C) 205002205.

Caruana [2] has proposed a multitask learning (MTL) model using a feedforward neural network which learns multiple tasks in parallel. Silver and Mercer [3] proposed another MTL learning model in which multiple tasks are learned sequentially and the knowledge of the previous tasks is selectively utilized for learning a new task. However, these models assume that training samples of a task are given in a batch and the system is informed which task is given to learn.

On the other hand, the concept of autonomous mental development (AMD) [4] attracts great attention as a new learning paradigm recently. AMD is discussed under more general learning conditions. For example, it requires a system to learn the knowledge of multiple tasks through realtime interactions with the environment. This implies that tasks and training samples are sequentially given to a system. It also requires a system to learn not only from known tasks but also from unknown tasks.

This paper presents a novel MTL model which satisfies the above requirements. In the proposed model, multiple pattern recognition tasks with the same input domain are learned sequentially by a single *Resource Allocating Network with Long-term Memory* (RAN-LTM) [5]. The proposed model has the following distinctive functions: *one-pass incremental learning*, *detection of task changes*, and *selective knowledge transfer*. The first function is already realized by RAN-LTM [5], and the second function can be implemented by checking the conflict between the prediction of RAN-LTM and the class label of a training sample [1]. The third function is newly introduced in the proposed MTL model, and it is realized not only by sharing the internal representation of RAN-LTM among different tasks but also by selectively transferring the information on class sub-regions of related task based on the task relatedness.

2 A Neural Network Model for MTPR

Training samples of a specific task are given sequentially for a certain period of time called *epoch* and discarded after training. Recognition tasks are switched one after another and could appear repeatedly. Therefore, we assume *one-pass incremental learning* and *sequential task learning* as a learning environment. We further assume that training samples have class information, but they have no information on the task category; that is, a learning system is not informed about which task the training sample comes from. Thus, a system has not only to detect task changes on its own, but also to recognize the task category.

2.1 One-Pass Incremental Learning Model

To learn an RBF network stably under the one-pass incremental learning environment, we have proposed *Resource Allocating Network with Long-Term Memory* (RAN-LTM) [5]. In RAN-LTM, several representative input-output data are selected and stored in the long-term memory (LTM). These representative data are called *memory items*, and they are retrieved from LTM and learned with a training sample to suppress the catastrophic interference.

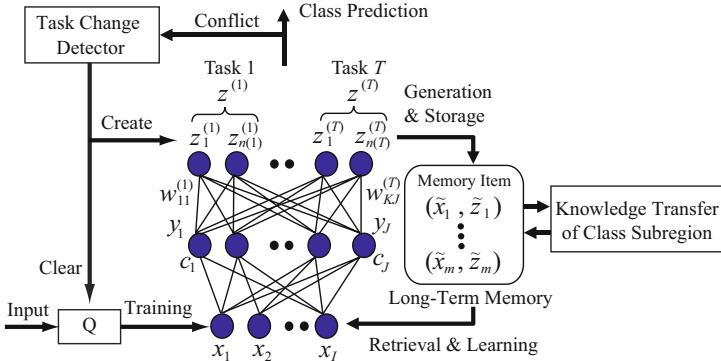


Fig. 1. Architecture of RAN-MTPR

In this section, we present an extended RAN-LTM model that can learn the assumed MTPR problems efficiently by transferring the knowledge of the previous tasks. We call this model *Resource Allocating Network for Multitask Pattern Recognition* (RAN-MTPR). Figure 1 illustrates the structure of RAN-MTPR. As seen from Fig. 1, RAN-MTPR has a single network whose inputs and RBF units are shared among different tasks. On the other hand, the outputs are defined for individual tasks. Let $\mathbf{x} = \{x_1, \dots, x_I\}'$ and $\mathbf{y} = \{y_1, \dots, y_J\}'$ be inputs and RBF outputs. Assume that T tasks have already been trained by RAN-MTPR. Then, the output layer is composed of T sections, each of which is responsible for the outputs for an individual task. Let $\mathbf{z}^{(t)} = \{z_1^{(t)}, \dots, z_{n(t)}^{(t)}\}$ be the outputs for the t th task; then, the whole outputs of RAN-MTPR is represented by $\mathbf{z} = \{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(T)}\}'$. Here, $z_k^{(t)}$ and $n(t)$ mean the label of the k th class and the number of classes for the t th task, respectively.

When a system recognizes that a training sample \mathbf{x} belongs to task t , the t th output section is activated. Then, the RBF outputs \mathbf{y} and the network output $\mathbf{z}^{(t)}$ is calculated as follows:

$$y_j = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{2\sigma^2}\right) \quad (j = 1, \dots, J) \quad (1)$$

$$z_k^{(t)} = \sum_{j=1}^J w_{kj}^{(t)} y_j + \xi_k^{(t)} \quad (k = 1, \dots, n(t)) \quad (2)$$

where $\mathbf{c}_j = \{c_{j1}, \dots, c_{jI}\}'$ is the center of the j th RBF, σ is the RBF width, $w_{kj}^{(t)}$ is the connection weight from the j th RBF to the k th output in the t th section; $\xi_k^{(t)}$ is the bias of the k th output.

Assume that the t th task is given. The learning of RAN-MTPR is conducted such that the error between the output $\mathbf{z}^{(t)}$ and the target $\mathbf{d}^{(t)}$ is minimized. However, in the assumed learning environment, minimizing only the output error may cause the catastrophic forgetting. As described above, RAN-MTPR learns the memory items $(\tilde{\mathbf{x}}_m, \tilde{\mathbf{z}}_m)$ ($m = 1, \dots, M$) with the given training sample

to suppress the unexpected forgetting. Thus, the training of RAN-MTPR is conducted such that the following squared error is minimized:

$$E^{(t)}(\mathbf{x}) = \|\mathbf{d}^{(t)} - \mathbf{z}^{(t)}(\mathbf{x})\|^2 + \sum_{m=1}^M \|\tilde{\mathbf{z}}_m^{(t)} - \mathbf{z}^{(t)}(\tilde{\mathbf{x}}_m)\|^2 \quad (3)$$

where M is the number of memory items and $\tilde{\mathbf{z}}_m^{(t)}$ corresponds to the t th section of $\tilde{\mathbf{z}}_m$. The learning algorithm of RAN-MTPR is basically the same as that of RAN-LTM (see [5] for details).

2.2 Task Change Detection

When an unknown class sample is given, the learning system has to check if the task is changed or not. If the unknown class actually belongs to the current task, the system only adds a new output node for the class and sets the connection weights from all the RBFs to the corresponding output. However, the unknown class may belong to an unknown task that has never appeared before. Under the assumed learning environment, only the class information on a single training sample is not enough to judge a task change; however, the system could use the class information on several training examples for the judgment because a certain number of samples are assumed to be given for a single task learning.

We introduce the following simple heuristics on the task change detection [1]: if q training samples with unknown classes are successively given, then the system confirms that a task change happens. To implement this heuristics, we introduce a queue Q in RAN-MTPR in order to keep the unknown class samples temporarily (see Fig. 1). When the queued samples reach the maximum size q , the system confirms that a task change happens. Then, a new output vector $\mathbf{z}^{(T+1)}$ is defined for the new task and the $(T+1)$ th output section in RAN-MTPR is trained with all the samples in Q .

2.3 Knowledge Transfer

The knowledge transfer in RAN-MTPR is implemented by the following mechanisms: (1) transfer by sharing the internal representation of RBFs and (2) transfer of the information on class subregions from related tasks. The first mechanism is very common and the internal representation of RBFs are utilized for learning the current task as inductive bias.

In the second mechanism, the task relatedness between the current task and a previous task is estimated, and the class subregion information in the related tasks is selectively transferred to learn the current task. The class subregion is represented by a set of memory items within a local input region. The knowledge transfer is carried out by setting the class label of the current training sample to these memory items.

Before describing the details of the knowledge transfer, let us define the following relatedness $R^{(T,t')}(i,j)$ between the i th class of task T and the j th class of task t' [6]:

$$R^{(T,t')}(i,j) = \frac{\mathbf{w}_i^{(T)} \mathbf{w}_j^{(t')\prime}}{\|\mathbf{w}_i^{(T)}\| \|\mathbf{w}_j^{(t')\prime}\|} \quad (4)$$

where $\mathbf{w}_i^{(T)}$ corresponds to the connection weights to the i th output of the T th section. Since the connection weights from the RBFs correspond to the coefficients of a linear surface, the task relatedness $R^{(T,t')}(i,j)$ evaluates the similarity of the class boundaries of the two tasks T and t' . Figure 2 illustrates an example of two-task problems. As seen from Fig. 2, Class 1 of Task 1 and Class 4 of Task 2 have similar boundaries; hence, the relatedness $R^{(2,1)}(4,1)$ would be high in this case.

When a training sample (\mathbf{x}, \mathbf{d}) is given, RAN-MTPR first find out the memory item $(\tilde{\mathbf{x}}^*, \tilde{\mathbf{z}}^*)$ where $\tilde{\mathbf{x}}^*$ is the closest input to \mathbf{x} . Then, a set of memory items $(\tilde{\mathbf{x}}_m, \tilde{\mathbf{z}}_m)$ satisfying $\|\tilde{\mathbf{x}}_m - \tilde{\mathbf{x}}^*\| < \eta$ are selected. Let \mathcal{S} be the index set of the closest memory item $(\tilde{\mathbf{x}}^*, \tilde{\mathbf{z}}^*)$ and the selected memory items $(\tilde{\mathbf{x}}_m, \tilde{\mathbf{z}}_m)$.

For $m' \in \mathcal{S}$, the relatedness $R^{(T,t')}(C(d^{(T)}), C(\tilde{\mathbf{z}}_{m'}^{(t')}))$ is calculated. Here, $C(d^{(T)})$ means the class of the training sample (\mathbf{x}, \mathbf{d}) when task is T . Similarly, $C(\tilde{\mathbf{z}}_{m'}^{(t')})$ means the class of the memory item $(\tilde{\mathbf{x}}_{m'}, \tilde{\mathbf{z}}_{m'})$ ($m' \in \mathcal{S}$) when task is t' . If $R^{(T,t')}(C(d^{(T)}), C(\tilde{\mathbf{z}}_{m'}^{(t')}))$ is larger than a certain value β , it means that the two class regions of Task T and t' are very similar. Then, the class information of the training sample is transferred to the selected memory item. This knowledge transfer is carried out through the following class label modification: $\tilde{\mathbf{z}}_{m'}^{(T)} \leftarrow d^{(T)}$. Then, these memory items are trained with the training sample by RAN-MTPR.

Since the class subregions of two tasks are not always matched, a wrong class label would be set to memory items by this knowledge transfer mechanism. Such memory items may work as a *negative inductive bias*. To avoid serious effects by the negative bias, the transferred incorrect class label should be modified when the system finds a conflict between the class label of a training sample and that of the closely located memory item. In the proposed method, the label modification is conducted by deleting the class label $\tilde{\mathbf{z}}^{(T)*}$ of the memory item which is closest to a training sample (\mathbf{x}, \mathbf{d}) .

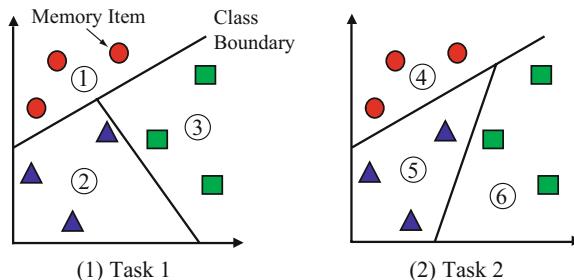


Fig. 2. An example of multitask pattern recognition problem

3 Performance Evaluation

3.1 Experimental Setup

The performance of RAN-MTPR is evaluated for the eight MTPR problems which are defined from the four UCI data sets. The data information is summarized in Table 1. We define two MTPR problems per UCI data, each of which consists of three tasks. Table 2 shows the correspondence between the labels of the original data and the defined tasks. Let us explain how to generate the training samples of Task 2 in Problem 1. Table 2(a) shows that Class 5 corresponds to Classes 1 and 2 in the original data set and Class 6 corresponds to Classes 3 and 4. Hence, the training samples whose original classes are either Class 1 or 2 are set to Class 5 in Task 2, and the same rule is applied to Class 6. We define MTPR problems for the other data sets in Table 1. However, due to space limitations, we omit the label correspondence tables.

Here, we assume that the three tasks are presented in turn several times and a certain number of training samples are given sequentially during every learning period of one task. This period is called *epoch*. In our experiments, the number of training samples per epoch is fixed at 40. For each MTPR problem, 720 training samples of the three tasks are given one by one in total (i.e., 240 training samples for each task). Therefore, the number of epoches is 18. Since the performance is generally influenced by the presentation order of tasks and training samples. We generate ten different sequences of training samples for each MTPR problem, and the average performance for the twenty sequences is estimated for each UCI data set. The parameters of RAN-MTPR are set as follows: $\sigma^2 = 2$, $q = 5$, $\eta = 0.1$, and $\beta = 0.2$.

3.2 Experimental Results

The performance of the proposed RAN-MTPR is compared with the two models. One is the conventional single-task learning model in which multiple RAN-LTMs

Table 1. Evaluated data sets

	#attrib.	#classes	#training	#test
Segmentation	19	7	210	2100
Thyroid	21	3	3772	3428
Vehicle	18	4	188	658
Vowel-context	10	11	528	462

Table 2. Two MTPR problems defined from Vehicle data

(a) Problem 1				(b) Problem 2			
original	Task 1	Task 2	Task 3	original	Task 1	Task 2	Task 3
1	1	5	7	1	1	3	5
2	2	5	8	2	2	3	6
3	3	6	7	3	2	4	5
4	4	6	8	4	1	4	6

Table 3. Final classification accuracies [%]. The two values in each cell are the average accuracy and the standard deviation. The accuracy with an asterisk means that the difference from the accuracy of RAN-MTPR is statistically significant.

	RAN-LTMs	RAN-MTPR(IR)	RAN-MTPR
Segmentation	84.2 ± 1.7	84.1 ± 1.9	84.4 ± 2.2
Thyroid	$80.6 \pm 0.5^*$	87.1 ± 1.7	87.2 ± 1.9
Vehicle	77.3 ± 0.9	77.4 ± 1.3	77.2 ± 1.5
Vowel-context	$57.4 \pm 2.8^*$	59.3 ± 2.6	60.7 ± 0.8

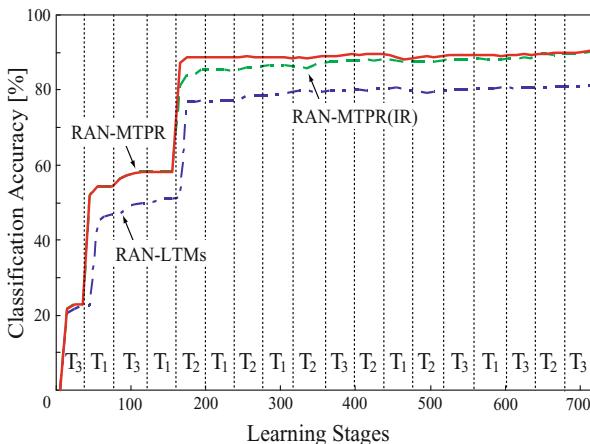


Fig. 3. Time evolution of the classification accuracies using Thyroid data

are individually trained for multiple tasks. The other is RAN-MTPR in which the knowledge transfer is only implemented by sharing the internal representation of RBFs. For notational simplicity, the two models are noted as RAN-LTMs and RAN-MTPR(IR), respectively.

Table 3 shows the final classification accuracies of the three models. If the knowledge of the previous tasks is successfully transferred, it is expected that the generalization performance is enhanced with less training samples and the final classification accuracy would be better by accumulating the performance advantage over the learning stages. As seen from Table 3, the final accuracies of the two RAN-MTPR models are higher than that of RAN-LTMs for ‘Thyroid’ and ‘Vowel-context.’ This result demonstrates that the knowledge transfer works well in RAN-MTPR even if it is conducted only by sharing the internal representation of RBFs. On the other hand, we can see little difference between the two RAN-MTPR models; thus, it seems that the transfer of class subregions does not affect the final classification accuracy very much.

To study the effectiveness of the knowledge transfer of class subregions, we examine how the classification accuracy is evolved during the learning. Figure 3 illustrates time evolutions of the classification accuracies of RAN-LTMs,

RAN-MTPR(IR), and RAN-MTPR. As seen from Fig. 3, the performance of RAN-MTPR quickly rises up compared with RAN-MTPR(IR) when Task 2 first appears at the 160th learning stage. From this result, we can say that the transfer of class subregions mainly contributes to enhancing the generalization performance until training samples are sufficiently given.

4 Conclusions and Future Work

In this paper, we proposed a new MTL model for pattern recognition called RAN-MTPR which could learn multiple multi-class recognition problems in a sequential way. The proposed model has the following distinctive functions: *one-pass incremental learning*, *task change detection*, and *selective knowledge transfer*. Although the first two functions have already proposed in our previous works [15], the third function is new. In RAN-MTPR, the following two types of the knowledge transfer were implemented: (1) transfer by sharing the internal representation of RBFs and (2) transfer of class subregions from the related tasks. The selective knowledge transfer is mainly realized by the latter.

In the experiments, the proposed RAN-MTPR was evaluated for the eight MTPR problems. The experimental results demonstrated that the classification accuracy of RAN-MTPR was higher than that of a single-task learning model. To study the effectiveness of the knowledge transfer of class subregions, we tested the RAN-MTPR model without this knowledge transfer. As a result, although there was little difference in the final classification, we verified that the transfer of class subregions contributes to enhancing the generalization performance while a small number of training samples are given.

In general, the performance of RAN-MTPR is influenced by how tasks are related each other because the transferred knowledge depends on the size of overlapped class subregions between the two tasks. Therefore, further experiments are needed to study the influence of task relatedness to the performance.

References

1. Ozawa, S., Roy, A.: Incremental Learning for Multitask Pattern Recognition Problems. In: Proc. of 17th Int. Conf. on Machine Learning and Applications (in press)
2. Caruana, R.: Multitask Learning. *Machine Learning* 28, 41–75 (1997)
3. Silver, D.L., Mercer, R.E.: The Task Rehearsal Method of Life-Long Learning: Overcoming Impoverished Data. In: Cohen, R., Spencer, B. (eds.) Canadian AI 2002. LNCS, vol. 2338, pp. 90–101. Springer, Heidelberg (2002)
4. Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., Thelen, E.: Autonomous Mental Development by Robots and Animals. *Science* 291(5504), 599–600 (2001)
5. Okamoto, K., Ozawa, S., Abe, S.: A Fast Incremental Learning Algorithm of RBF Networks with Long-Term Memory. In: Proc. Int. Joint Conf. on Neural Networks, pp. 102–107 (2003)
6. Silver, D., Mercer, R.: Selective Functional Transfer: Inductive Bias from Related Tasks. In: Proc. of IASTED Int. Conf. on AI and Soft Computing, pp. 182–189 (2001)

Automatic Discovery of Subgoals in Reinforcement Learning Using Strongly Connected Components

Seyed Jalal Kazemita^{bar} and Hamid Beigy

Computer Engineering Department,
Sharif University of Technology, Tehran, Iran
`{jkazemitabar, beigy}@ce.sharif.edu`

Abstract. The hierarchical structure of real-world problems has resulted in a focus on hierarchical frameworks in the reinforcement learning paradigm. Preparing mechanisms for automatic discovery of macro-actions has mainly concentrated on subgoal discovery methods. Among the proposed algorithms, those based on graph partitioning have achieved precise results. However, few methods have been shown to be successful both in performance and also efficiency in terms of time complexity of the algorithm. In this paper, we present a SCC-based subgoal discovery algorithm; a graph theoretic approach for automatic detection of subgoals in linear time. Meanwhile a parameter tuning method is proposed to find the only parameter of the method.

1 Introduction

Introducing abstraction to reinforcement learning has been mostly obtained by deploying hierarchical structures such as the *options* framework [9]. In early applications, the solution incorporated the use of hand-tailored macro-actions that lead the agent to important milestones or *subgoals* in the way to goal.

There after, the major research has been suggesting methods for automatically finding subgoal states. Using these subgoals, the agent can build options and thereby increase the speed of learning in future tasks without the aid of programmer. Besides enabling the agent to solve large-scale problems, this can also be of great help in real domains where enough human knowledge is not available to support the agent with useful macro-actions.

Among recent methods, a considerable number try to partition the transition graph and choose as subgoals the connecting states between densely-connected areas [8]. Among the proposed methods, some search for states that are situated between densely-connected areas of the state space. Most of such approaches use graph partitioning techniques and select the connecting states as subgoals [8][14]. These methods mainly suffer from the intrinsic NP-hardness of the partitioning problem which by itself has imposed the use of approximation techniques which result in either loosing precision of the initially suggested concept or incurring a computational cost to the online acting agent. In this paper, we present a graph

theoretic approach for automatic detection of subgoals in linear time. This is done by finding strongly connected components of the environment. Meanwhile a parameter tuning method is proposed to find the only parameter of the method.

In the following sections, we first describe the SCC-based subgoal discovery algorithm in section 2. The algorithm is then evaluated in section 3 and finally the paper concludes in section 4 with a glance at future works.

2 Proposed Method

In this section, we present a method based on graph theory to discover subgoal states in an environment. First, subgoal state are characterized and after that, a graph algorithm is introduced to find such states.

In our method, subgoals are states that

- Connect dense regions of the state space [6][7].
- The transition probability from one region to the other is low [8].

In this definition, subgoals are the border states of regions that are highly intra-connected. However, there are few transitions that connect these dense regions. If an agent can find such regions automatically, it will easily be able to identify the inter-connecting states as subgoal.

The algorithm for finding dense regions of the state space works as follows. After running Q-learning for a few episodes, the transition graph is constructed from accumulated trajectories. In this directed graph (DG), vertices stand for states and edges demonstrate the transitions with weights showing the frequency of transitions. In order to take into account the low probability transitions between outer states of two regions, we used a threshold, t_t , as the minimum number of transitions between two states needed to stand for an edge in graph. After that, algorithm [1] is used to find *strongly connected components* (SCCs) of the transition graph. An SCC of a directed graph $G = (V, E)$ is a maximal set of vertices $C \subseteq V$ such that for every pair of vertices u and v in C , we have both a path from node u to node v and vice versa; that is, vertices u and v are reachable from each other.

Algorithm 1. SCC-Inspector(G)

- 1: Call DFS(G) to compute finishing times $f(u)$ for each vertex u
 - 2: Computer G^T
 - 3: Call DFS(G^T), but in the main loop of DFS, consider the vertices in order of decreasing $f(u)$ (as computed in line 1)
 - 4: Output the vertices of each tree in the depth-first forest formed in line 3 as a separate strongly connected component
-

First the graph is traversed using depth first search (DFS) in step 1. This traversal method also assigns a timestamp to every vertex v . The timestamp $f(v)$ records when the search *finishes* examining the adjacent nodes of v . In

step 2 the transpose graph, G^T , is constructed, which has the same nodes as G and the direction of all edges are reversed. After that in step 3 the transpose graph is searched with DFS starting from an unmarked node which has the largest finishing time. This is repeated until all nodes have been marked. Finally in step 4 the vertices of each tree in the depth first forest generated in step 3 are output as a separate strongly connected component [3].

Due to the sparseness of transition graphs in most RL problems, the adjacency list representation was used instead of the adjacency matrix. This resulted in an order of magnitude decrease from the initial cost of $\theta(N^2)$ for storing the edges among vertices of the graph. Thus, the DFS algorithm creates the depth first forest of a DG in a running time of $\Theta(V+E)$ and creation of G^T takes $O(V+E)$ time. Hence, SCC-Inspector(G) is a linear-time algorithm.

3 Experimental Results

In this section, we present some experiments to assess our SCC-based subgoal discovery method. Like many other methods, we used the two-room gridworld environment as the test bench [56][78][2]. Figure 1 shows a two-room gridworld with a middle hallway. The agent starts randomly from a square in the left room and tries to get to the bottom right corner of the right room. Four primitive actions are available to the agent in all states: up, right, down, left. The agent remains in the same location in case the direction of movement is blocked. Getting in the goal square results in a reward value of 1 and a small punishment of 10^{-6} is received for all other states. To take into account the stochastic nature of the environment, choosing an action moves the agent in the intended direction with probability 0.9 and in a uniform random direction with probability 0.1.

The trajectories were collected in the first 10 episodes of learning task and edges with a weight of $t_t = 14$ or higher were included in the SCC-Inspector algorithm. The algorithm was evaluated in 50 runs. As bleached in figure 2, in %96 of runs, the hallway or one of its neighbors at a distance of two were identified as subgoal.

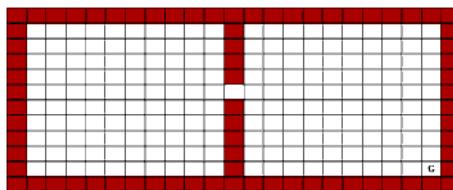


Fig. 1. Hallway position in a simple two-room gridworld

Figure 3 depicts a two-room gridworld with two hallways. The obtained results after 50 executions are shown in table 1. For both subgoals, in %94 of times the subgoal states were the hallways or one of their three neighboring squares.

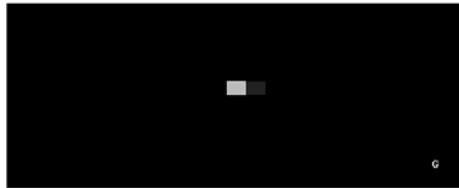


Fig. 2. Identified subgoals in a simple two-room gridworld in 50 runs

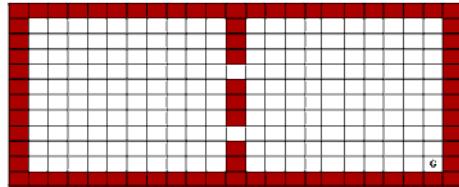


Fig. 3. Hallway positions in a two-room gridworld

Table 1. Identified subgoals in the two-room gridworld in 50 runs

Square	Northern Hallway	Southern Hallway
Hallway	19	18
Left Neighbor	19	23
Right Neighbor	9	4
2Right Neighbor	0	2
Others	3	3

According to the results, the identified subgoals are quite satisfactory. Unlike other methods, these results are gained in an efficient linear time without any alteration to the base concept of finding SCCs.

In order to set the t_t threshold parameter, edge weight histograms were assessed. According to the simulations in diverse gridworld domains, local minima points in the histogram indicate potential values for t_t threshold. As an example, figure 4 shows an asymmetric four-room gridworld with five hallways. Assigning values 12, 21, 31 to the threshold guided us to find the five subgoals connecting the four rooms. It can also be concluded from figure 5 that using a value of $t_t = 9$ can also result in a precise detection of subgoals in the room containing the goal state. This fact was confirmed in our experiments.

The proposed method is related to some other algorithms in the definition of subgoal states. Menache's *Q-Cut* method, partitions the transition graph by running a max-flow/min-cut algorithm and evaluates the cuts by RatioCut metric. This method has a complexity of $O(N^3)$ in terms of states of the environment [6]. Due to the point that both Q-Cut and SCC-based algorithms consider the transition graphs constructed in all previous episodes of the learning task, it can be observed that the proposal algorithm has a less time complexity of two orders of magnitude in finding subgoal states.

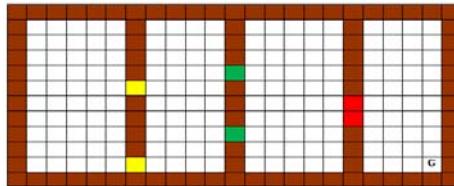


Fig. 4. Hallway positions in a four-room gridworld

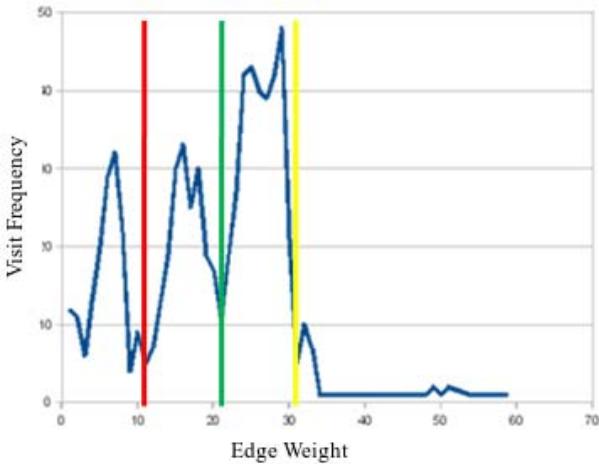


Fig. 5. Histogram and selected parameters for the four-room gridworld

Şimşek in her *L-Cut* method, approximates the NCut metric by spectral clustering. This method incorporates running an $O(h^3)$ time algorithm where h is the number of states observed in an acceptable number of transitions in the last episode [8]. This method has a superiority in that it finds cuts of the local transition graph and not the whole accumulated trajectories in previous episodes. However, assessing the simulations show that the tuned parameter h in gridworld domains is high enough compared to the small number of states in the tested environments. Considering the parameters given in the L-Cut paper, we simulated the agent's interaction with the environment and observed that only in the 10th episode of the learning task, the agent visits %64 of the state space in an average of 50 runs. Moreover, it should be noted that the $O(h^3)$ time spectral clustering algorithm should be called at the end of each episode to result in proper results [8]. This is while the SCC-based method calls the linear time algorithm only once whole through the experiment.

The proposed method has a special dominance in setting the parameter(s). While other precise methods such as L-Cut and Relative Novelty [7] include many programmer-tuned parameters, t_t as the only parameter in this method has shown to be a non-sensitive parameter which can also be easily set in gridworld domains by observing the histogram of edge weights.

4 Conclusions and Future Works

The SCC-based subgoal discovery algorithm is a novel graph-based approach for solving complex MDPs where enough reinforcement is not available. Used in early episodes of a learning task, the method results in identifying useful subgoals that improve the agent's performance in attaining other goals in similar tasks.

The presented algorithm finds strongly connected components of the environment. By using adjacency list representation, the algorithm discovers subgoals in linear time. Meanwhile, a parameter tuning method was proposed to find the only parameter of the method.

Evaluating the algorithm in more complex domains and preparing mechanisms for automatic tuning of the threshold parameter in such environments is one of our works in progress. After introducing automatically identified subgoals, it is the time to assess the effect of utilizing subgoals both in coming episodes of a learning task and also similar future tasks. Other than helping the agent for more efficient exploration in similar tasks, subgoals of achievement can also be transferred to other agents in cooperative multi-agent tasks. An important direction for our future research is considering the effect of autonomously gained and transferred skills in multi-agent domains.

References

1. Asadi, M., Huber, M.: Reinforcement learning acceleration through autonomous subgoal discovery. In: Proceedings of The 2005 International Conference on Machine Learning, Models, Technologies and Applications, pp. 69–74 (2005)
2. Chen, F., Gao, Y., Chen, S., Ma, Z.: Connect-based subgoal discovery for options in hierarchical reinforcement learning. In: Proceedings of the 3rd International Conference on Natural Computation, pp. 698–702 (2007)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2001)
4. Mannor, S., Menache, I., Hoze, A., Klein, U.: Dynamic abstraction in reinforcement learning via clustering. In: Proceedings of the 21st International Conference on Machine Learning, pp. 560–567 (2004)
5. McGovern, A., Barto, A.G.: Automatic discovery of subgoals in reinforcement learning using diverse density. In: Proceedings of the 18th International Conference on Machine Learning, pp. 361–368 (2001)
6. Menache, I., Mannor, S., Shimkin, N.: Q-cut - dynamic discovery of sub-goals in reinforcement learning. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS, vol. 2430, pp. 295–306. Springer, Heidelberg (2002)
7. Simsek, Ö., Barto, A.G.: Using relative novelty to identify useful temporal abstractions in reinforcement learning. In: Proceedings of the 21st International Conference on Machine Learning, pp. 751–758 (2004)
8. Simsek, Ö., Wolfe, A.P., Barto, A.G.: Identifying useful subgoals in reinforcement learning by local graph partitioning. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 816–823 (2005)
9. Sutton, R.S., Precup, D., Singh, S.P.: Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Journal of Artificial Intelligence 112, 181–211 (1999)

Part IX

Special Session: Dynamics of Neural Networks

Bifurcation and Windows in a Simple Piecewise Linear Chaotic Spiking Neuron

Tomonari Hasegawa and Toshimichi Saito

Hosei University, Koganei, Tokyo, 184-0002 Japan

Abstract. This paper studies a piecewise linear chaotic spiking oscillator relating to neuron models. Repeating vibrate-and-fire dynamics, the system can exhibit chaotic/periodic spike-trains and related bifurcation phenomena. Deriving the return map of a state variable, we can analyze typical phenomena precisely and have confirmed an interesting bifurcation phenomena of chaotic spike-trains and window structure of period-doubling route.

1 Introduction

Spiking neuron models are important objects to study nonlinear phenomena and brain function [1] - [5]. Repeating integrate-and-fire switchings, the models can exhibit rich periodic/chaotic spike-trains and related bifurcation phenomena. Classification of the spike-trains is basic to consider signal processing function in the brain and analysis of typical bifurcation phenomena can contribute nonlinear dynamical systems theory. The spiking neuron can be a building block of pulse-coupled neural networks that exhibit rich synchronous phenomena [6]-[8]. Potential applications of such systems are many, including image/signal processing, associative memories and ultra-wide-band communications [7]-[11].

This paper studies a piecewise linear (PWL) chaotic spiking oscillator (CSO) relating to the spiking neuron models [8] [12]. The CSO has two state variables which can vibrate below the threshold. When a state variable reaches a threshold, it is reset to the base level. Repeating the vibrate-and-fire behavior, the CSO can output rich chaotic/periodic spike-trains and related bifurcation phenomena. We then pay special attention to (a) chaotic chattering or intrinsically bursting behavior, and (b) period-doubling bifurcation to chaos with line-like inter-spike interval (ISI) spectrum. In a parameter subspace, the CSO mainly exhibits (a) and existence region of (b) constructs infinite window structure. Using the exact piecewise solutions, we can derive return map of a state variable at the firing moment and can analyze the typical phenomena precisely. These results provide basic information for spike-based bifurcation theory and spike-based engineering applications.

For novelty of this paper, we note that Ref. [12] studies a generalized CSO in different parameter range and that Ref. [8] studies synchronization of the CSO with restricted parameters for chaos.

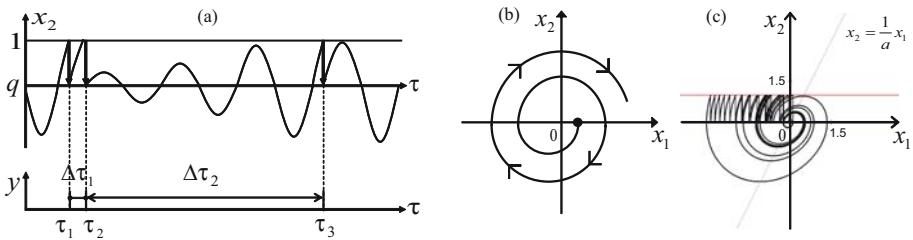


Fig. 1. Dynamics of the PWL-CSO. (a) Vibrate-and-fire switching. (b) Unstable spiral. (c) Chaotic attractor for $a = 0.5$ and $q = 0$.

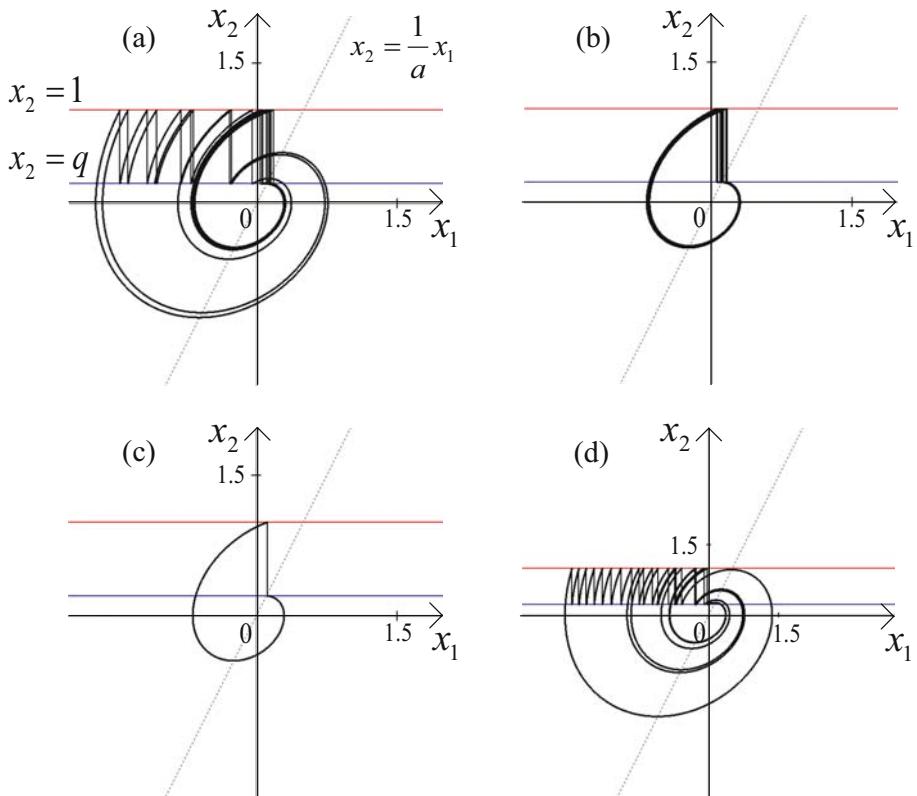


Fig. 2. Typical attractors for $a = 0.5$. (a) Chaos for $q = 0.208$, (b) Chaos for $q = 0.209$, (c) Periodic attractor for $q = 0.212$, (d) Chaos for $q = 0.218$.

2 Piecewise Linear Spiking Oscillator

The dynamics of the CSO is described by

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = ax_2 - x_1 \end{cases} \quad \text{for } x_2(\tau) < 1 \quad (1)$$

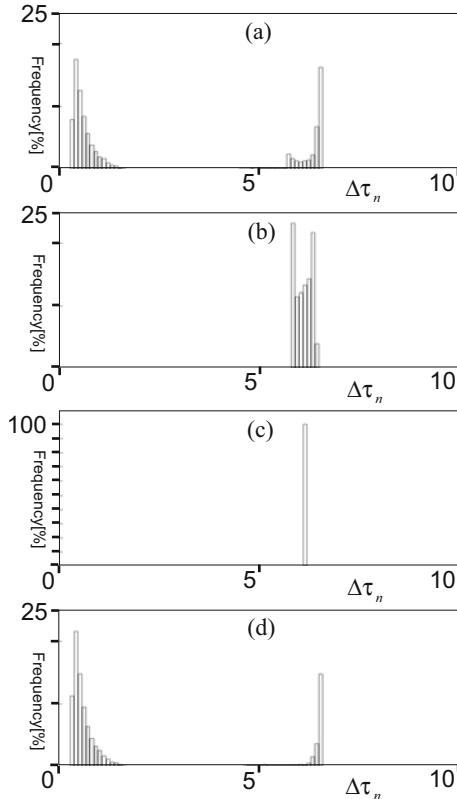


Fig. 3. ISI spectrums for $a = 0.5$. (a) Chaos for $q = 0.208$, (b) Chaos for $q = 0.209$, (c) Periodic attractor for $q = 0.212$, (d) Chaos for $q = 0.218$.

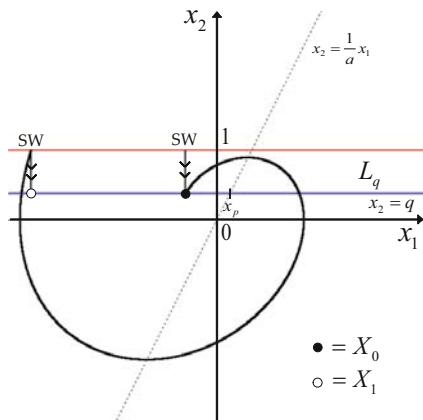
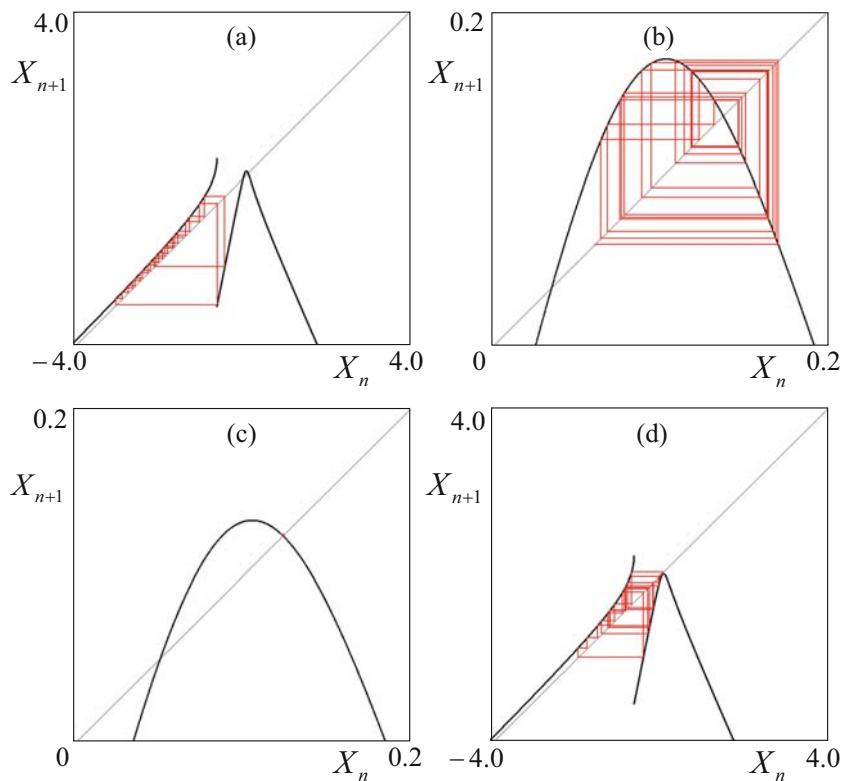
where “.” denotes differentiation by dimensionless time τ . x_1 and x_2 are dimensionless state variables. If $0 < a < 2$ then the system has unstable complex characteristic roots. x_1 and x_2 can vibrate divergently as shown in Fig. 1. If x_2 reaches the threshold 1, x_2 is reset to the base $q < 1$ instantaneously holding continuity of x_1 that is proportional to capacitor voltage in a circuit model [13]:

$$(x_1(\tau_+), x_2(\tau_+)) = (x_1(\tau), q) \text{ if } x_2(\tau) = 1 \quad (2)$$

This dimensionless equation has two parameters a and q . Repeating such vibrate-and-fire behavior, the CSO can output a variety of spike-trains:

$$z(\tau_+) = \begin{cases} 1 & \text{if } x_2(\tau) = 1 \\ 0 & \text{otherwise} \end{cases}$$

The CSO can exhibits a variety of attractors as suggested in Fig. 1 (c) and Fig. 2. For $q < 0$, simple periodic attractor is changed into chaotic attractor via period doubling bifurcation [12]. This paper considers bifurcation for $0 < q < 1$

**Fig. 4.** Key objects and definition of return map**Fig. 5.** Some shapes of return map f for $a = 0.5$. (a) Chaos for $q = 0.208$, (b) Chaos for $q = 0.209$. (c) Periodic attractor for $q = 0.212$, (d) Chaos for $q = 0.218$.

whose analysis is not sufficient in our previous works [12]. In the parameter range the CSO exhibits a variety of chaotic/periodic attractors.

The spike-trains are characterized by firing time τ_n and ISI $\Delta\tau_n$ as shown in Fig. 1. That is, let τ_n be the n -th firing time and let $\Delta\tau_n = \tau_{n+1} - \tau_n$ be the n -th ISI. Fig. 3 shows the ISI spectrums corresponding to Fig. 2.

3 Return Map and Analysis

In order to analyze the chaotic/periodic attractors precisely, we derive the return map. Fig. 4 shows key object. Let $L_q = \{(x_1, x_2) \mid x_2 = q\}$ and let a point on L_q be represented by its x_1 -coordinate. Let us consider the trajectory that is reset onto L_q at the first firing time τ_1 . The trajectory vibrates below the threshold and resets to L_q at the second firing time τ_2 . Let X_1 and X_2 be the first and second reset points, respectively. Since X_1 determines X_2 , we can define 1D return map:

$$F : L_q \rightarrow L_q, \quad X_1 \mapsto X_2 \quad (3)$$

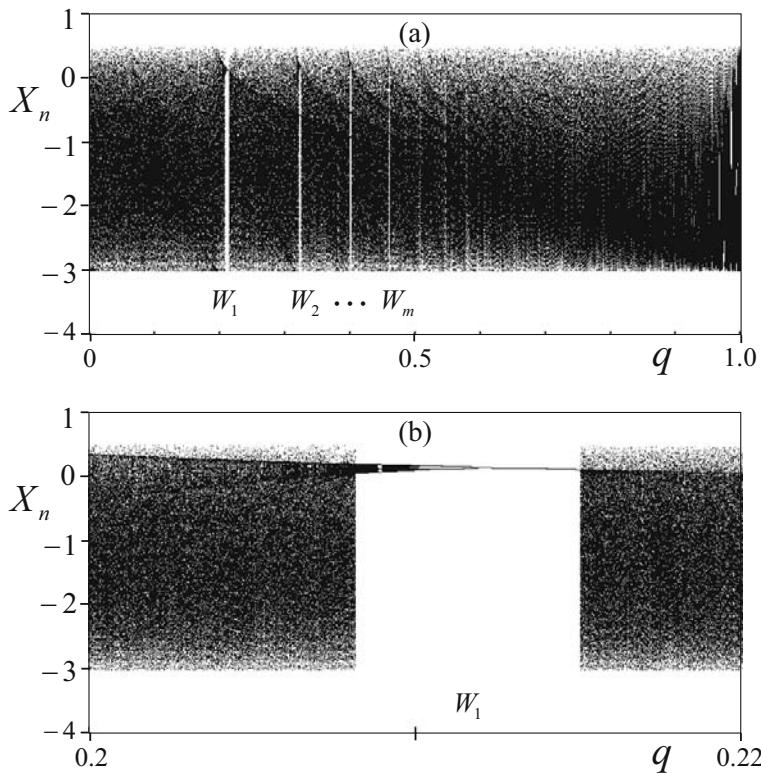


Fig. 6. Bifurcation diagrams for $a = 0.5$. (a) Return map, (b) Enlargement of window W_1 .

Fig. 5 depicts several maps corresponding to Fig. 2. In the parameter space, chaotic attractors (Fig. 5(a) and (d)) are dominant, however, the small attractors (Fig. 5(b) and (c)) can exist. Fig. 6 shows bifurcation diagram for increasing q where we can see that many windows exist. The window W_1 corresponds to the small attractors (Fig. 5(b) and (c)). In each window, a small periodic attractor is born by tangent bifurcation and is changed into small chaotic attractor via period-doubling bifurcation. The chaotic attractor is not stable for initial state but has narrow-band spectrum as shown in Fig. 3(b).

4 Conclusions

Basic dynamics of CSO is studied in this paper. Using the return map, bifurcation of chaotic spike-trains and window of period doubling chaos is analyzed precisely. Future problems include detailed analysis of bifurcation phenomena, classification of spike-trains, comparison with other spiking neurons models [3] including the piecewise constant neuron [13].

References

1. Keener, J.P., Hoppensteadt, F.C., Rinzel, J.: Integrate-and-fire models of nerve membrane response to oscillatory input. SIAM J. Appl. Math. 41, 503–517 (1981)
2. Miroollo, R.E., Strogatz, S.H.: Synchronization of pulse-coupled biological oscillators. SIAM J. Appl. Math. 50, 1645–1662 (1990)
3. Izhikevich, E.M.: Simple Model of Spiking Neurons. IEEE Trans. Neural Networks 14, 1569–1572 (2003)
4. Kon’no, Y., Saito, T., Torikai, H.: Rich dynamics of pulse-coupled spiking neurons with a triangular base signal. Neural Networks 18, 523–531 (2005)
5. Lindner, B., Chacron, M.J., Longtin, A.: Integrate-and-fire neurons with threshold noise: A tractable model of how interspike interval correlations affect neuronal signal transmission. Physical Review E 72, 021911 (2005)
6. Hopfield, J.J., Herz, A.V.M.: Rapid local synchronization of action potentials: toward computation with coupled integrate-and-fire neurons. Proc. Natl. Acad. Sci., USA 92, 6655–6662 (1995)
7. Campbell, S.R., Wang, D., Jayaprakash, C.: Synchrony and desynchrony in integrate-and-fire oscillators. Neural computation 11, 1595–1619 (1999)
8. Nakano, H., Saito, T.: Grouping synchronization in a pulse-coupled network of chaotic spiking oscillators. IEEE Trans. Neural Networks 15(5), 1018–1026 (2004)
9. Izhikevich, E.M.: Weakly pulse-coupled oscillators, FM interactions, synchronization, and oscillatory associative memory. IEEE Trans. Neural Networks 10, 508–526 (1999)
10. Sushchik, M., Rulkov, N., Larson, L., Tsimring, L.: Chaotic pulse position modulation: a robust method of communicating with chaos. IEEE Comm. Lett. 4, 128–130 (2000)

11. Hamanaka, H., Torikai, H., Saito, T.: Quantized spiking neuron with A/D conversion functions. *IEEE Trans. Circuits Syst. II* 53(10), 1049–1053 (2006)
12. Nakano, H., Saito, T.: Basic dynamics from an integrate-and-fire chaotic circuits with a periodic input. *IEICE Trans. Fundamentals* E84-A, 5, 1293–1300 (2001)
13. Hasegawa, T., Matsuoka, Y., Saito, T.: Analysis of Inter-Spike Interval Characteristics of Piecewise Constant Chaotic Spiking Oscillators. In: Proc. of IEEE-INNS/IJCNN, pp. 3435–3440 (2008)

Bifurcation between Superstable Periodic Orbits and Chaos in a Simple Spiking Circuit

Yuji Kawai and Toshimichi Saito

EE Dept., Hosei University, Koganei, Tokyo, 184-0002 Japan

Abstract. This paper studies typical nonlinear dynamics of spiking circuit including two capacitors. Applying impulsive switching depending on both state and time, the circuit can exhibit rich chaotic/periodic phenomena. We pay special attention to superstable periodic orbits and related bifurcation phenomena. The circuit dynamics can be simplified into a piecewise linear one-dimensional return map that enables us to analyze basic bifurcation phenomena precisely.

1 Introduction

In spiking neurons and related systems, impulsive switch (ISW) plays important roles in generation of rich dynamics. We first note that there are two categories for the switching condition: state-controlled switching (SCSW) and time-controlled switching (TCSW). For example, in spiking neuron models [1]-[3], the SCSW can cause rich spike-trains analysis of which is basic to understand neural information processing function [4] [5]. The spiking neuron can be a building block of pulse-coupled neural networks (PCNN) having rich synchronization phenomena [6] and the TCSW can be basic to consider the phenomena. The spiking neurons and PCNN have many potential applications including associative memory, signal processing and spike-based communications [6]-[9]. Analysis of ISW-based dynamics is important in studies of spiking neurons and PCNN.

This paper considers typical nonlinear dynamics of spiking circuit (SKC) including two capacitors, one SCSW and one TCSW. The SCSW is closed when a capacitor voltage exceeds a threshold, the TCSW is closed if a spike-train input arrives, and they are connecting in series. The SCSW has been used in various spiking neuron models and the TCSW can represent effect of external input: the SKC with both SCSW and TCSW is basic to consider rich dynamics in PCNN. Noticing switching instant of the TCSW, we can define piecewise linear (PWL) one dimensional return map and can analyze basic bifurcation between superstable periodic orbits (SSPO) and chaos precisely. Although the SSPO is superstable for initial value, it can be sensitive for parameters and can be complex shape. For novelty of this paper we note that if SCSW and TCSW are connected in parallel then the SKC does not have PWL return map and the analysis is much harder [10]. We also note that bifurcation of SSPO has not been discussed in our previous works [10]-[11].

2 The Circuit Model

Fig. 1 shows the circuit model. If switches are open all the time, a 2-port VCCS and two capacitors construct a linear circuit:

$$\frac{d}{dt} \begin{pmatrix} C_1 v_1 \\ C_2 v_2 \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (1)$$

We assume that Eq. (1) has unstable complex characteristic roots $\delta\omega \pm j\omega$. The SCSW S_s is closed if v_1 exceeds the threshold V_T and the TCSW S_t is closed if impulse-train with period T arrives. Since the two switches are connected in series, the circuit switching is described by:

$$(v_1(t^+), v_2(t^+)) = (E, v_2(t)) \text{ if } v_1 > V_T \text{ and } t = mT. \quad (2)$$

where m is a nonnegative positive integer. That is, if this condition is satisfied then v_1 is reset to the base E instantaneously holding the continuity property of v_2 . Using the following dimensionless variables and parameters:

$$\begin{aligned} \tau &= \omega t, \quad x = \frac{v_1}{V_T}, \quad y = \frac{1}{V_T} \left(p v_1 + \frac{g_{12}}{\omega C_1} v_2 \right), \quad d = \omega T, \quad \delta = \frac{1}{2\omega} \left(\frac{g_{11}}{C_1} + \frac{g_{22}}{C_2} \right) > 0 \\ p &= \frac{1}{2\omega} \left(\frac{g_{11}}{C_1} - \frac{g_{22}}{C_2} \right), \quad q = \frac{E}{V_T}, \quad \omega^2 = -\frac{g_{12}g_{21}}{C_1C_2} - \frac{1}{4} \left(\frac{g_{11}}{C_1} - \frac{g_{22}}{C_2} \right)^2 > 0 \end{aligned} \quad (3)$$

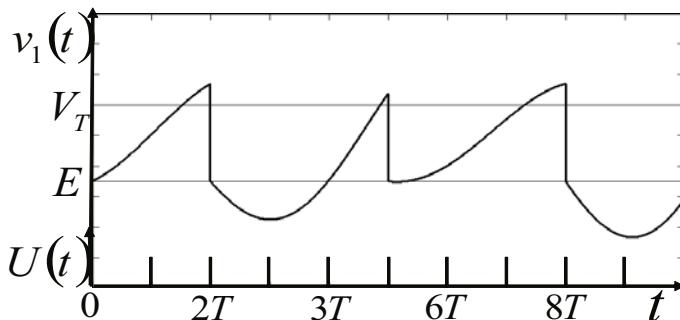
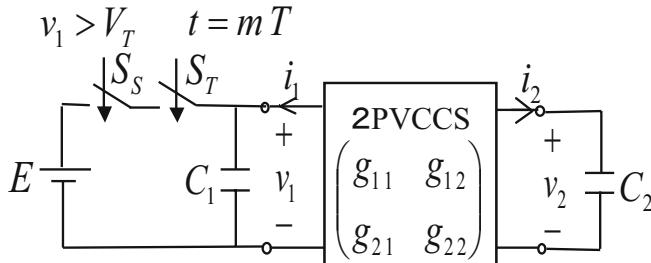


Fig. 1. Chaotic spiking circuit

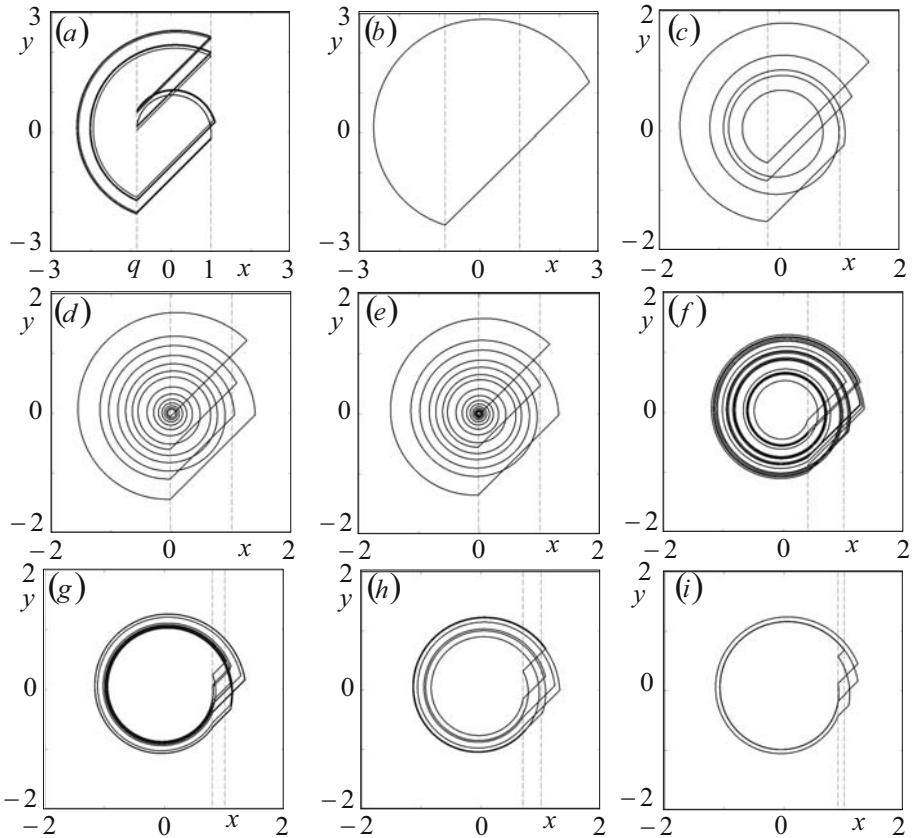


Fig. 2. Typical attractors for ($\delta = 0.05$, $p = 1$, $d = \pi/4$). (a) Chaos for $q = -0.855$, (b) SSPO for $q = -0.854$, (c) SSPO for $q = -0.2$, (d) SSPO for $q = -0.02$, (e) SSPO for $q = -0.01$, (f) Chaos for $q = 0.4$, (g) SSPO for $q = 0.7$, (h) Chaos for $q = 0.8$, (i) SSPO for $q = 0.9$.

Eqs. (1) and (2) are transformed into

$$\begin{aligned} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} &= \begin{pmatrix} \delta & 1 \\ -1 & \delta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \text{ if } x(\tau) < 1 \text{ or } \tau \neq md \\ (x(\tau^+), y(\tau^+)) &= (q, y(\tau) - p(x(\tau) - q)) \text{ if } x(\tau) > 1 \text{ and } \tau = md \end{aligned} \quad (4)$$

where “.” denote differentiation by τ . This normalized equation has four parameters: the damping δ , the jumping slope p , the base q and the basic period d . If switches are open all the time, Eq. (4) has the exact solution:

$$\begin{pmatrix} x(\tau) \\ y(\tau) \end{pmatrix} = e^{\delta\tau} \begin{pmatrix} \cos \tau & \sin \tau \\ -\sin \tau & \cos \tau \end{pmatrix} \begin{pmatrix} x(\tau_s) \\ y(\tau_s) \end{pmatrix} \quad (5)$$

Using the solution, we can calculate attractors precisely and typical ones are shown in Fig. 2. Let us overview bifurcation for parameter q . As q increases,

chaotic attractor (Fig. 2 (a)) changed into simple periodic attractor (Fig. 2 (b)) and then to a variety of periodic/chaotic attractors (Fig. 2 (c)-(i)). All the periodic attractors in this figure are superstable for initial state as shown in the next section.

3 Superstable Periodic Orbits and Bifurcation

In order to analyze the phenomena, we define the 1-D return map. As illustrated in Fig. 3, we assume that a trajectory exceeds the threshold $x = 1$ and an input arrives at $\tau = 0$. The trajectory is rest to the base $L_q \equiv \{(x, y) | x = q\}$ and let y_0 be the reset point (a point on L_q is represented by its y -coordinate). The trajectory started from y_0 rotates divergently around the origin and it is reset to L_q at some input arriving moment $\tau = m_1 d$. Let y_1 be the first reset point. In a likewise manner, the trajectory started from y_1 rotates around the origin and is reset to L_q at $\tau = m_2 d$. ($m_1 = 2$ and $m_2 = 4$ in Fig. 3). Let y_n denote the n -th reset point and let $m_n d$ be the n -th reset time ($y_n = y(m_n d)$). Since y_n determines y_{n+1} , we can define 1-D return map

$$f : y_0 \mapsto y_1. \quad (6)$$

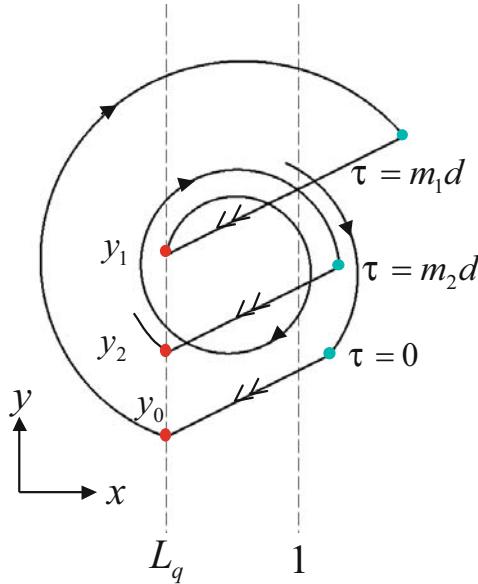


Fig. 3. Phase plane

The map can be calculated using the piecewise exact solutions:

$$y_{n+1} = f(y_n) = y'_{n+1} - p(x'_{n+1} - q) \quad (7)$$

$$\begin{pmatrix} x'_{n+1} \\ y'_{n+1} \end{pmatrix} = e^{\delta md} \begin{pmatrix} \cos md & \sin md \\ -\sin md & \cos md \end{pmatrix} \begin{pmatrix} q \\ y_n \end{pmatrix} \quad \text{for } y_n \in B_m$$

where B_m is a subset of L_q such that a trajectory started from B_m at $\tau = 0$ is reset to L_q at $\tau = md$. Note that the return map is PWL and has discontinuity point on the end points of B_m . Fig. 4 shows typical 1-D return maps corresponding to Fig. 3. In order to analyze attractors stability we use slope of each branch of the map. Using the exact piecewise solution, the slope can be calculated as follows:

$$Df(y_n) = e^{\delta md}(\cos(md) - p \sin(md)) \quad \text{for } y_n \in B_m. \quad (8)$$

Here we define periodic points and chaos for the return map. A point x_p is said to be a k -periodic point if $x_p = f^k(x_p)$ and $x_p \neq f^l(x_p)$ for $1 \leq l < k$ where $f^l(x_p) = f(f^{l-1}(x_p))$ and $f^0(x) \equiv x$. A 1-periodic point is referred to as a fixed

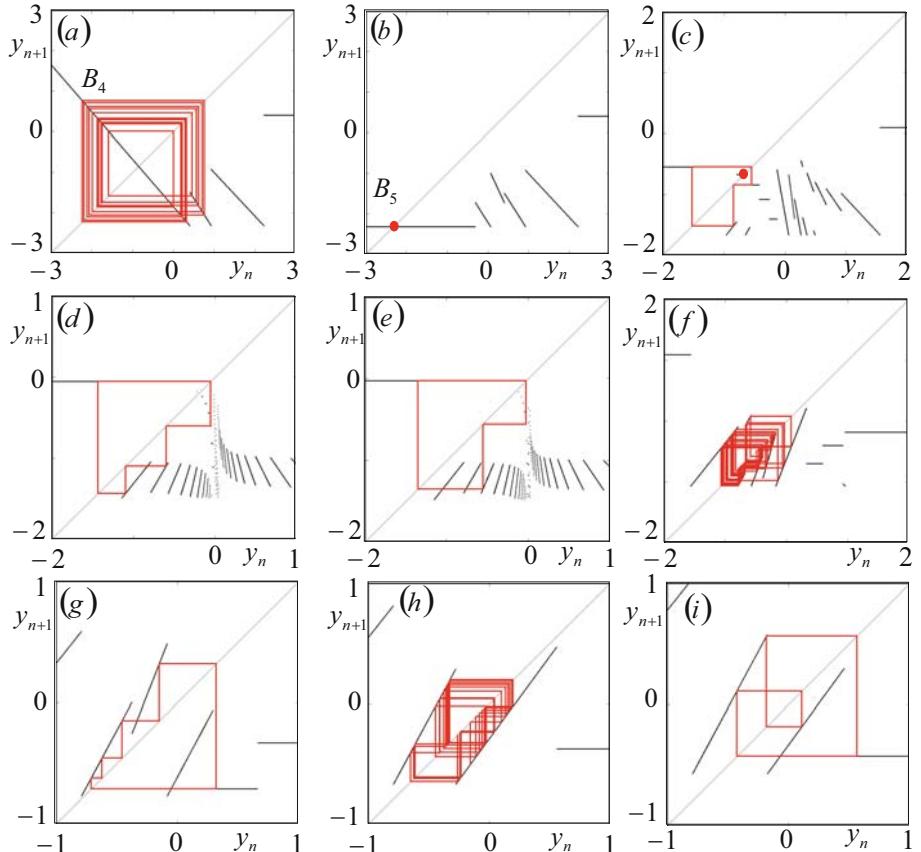


Fig. 4. Typical 1-D return maps ($\delta = 0.05$, $p = 1$, $d = \pi/4$). (a) Chaos for $q = -0.855$, (b) SSFP with for $q = -0.854$, (c) Coexisting SSPP and SSFP for $q = -0.2$, (d) SSPP for $q = -0.02$, (e) SSPP for $q = -0.01$, (f) Chaos for $q = 0.4$, (g) SSPP for $q = 0.7$, (h) Chaos for $q = 0.8$, (i) SSPP for $q = 0.9$.

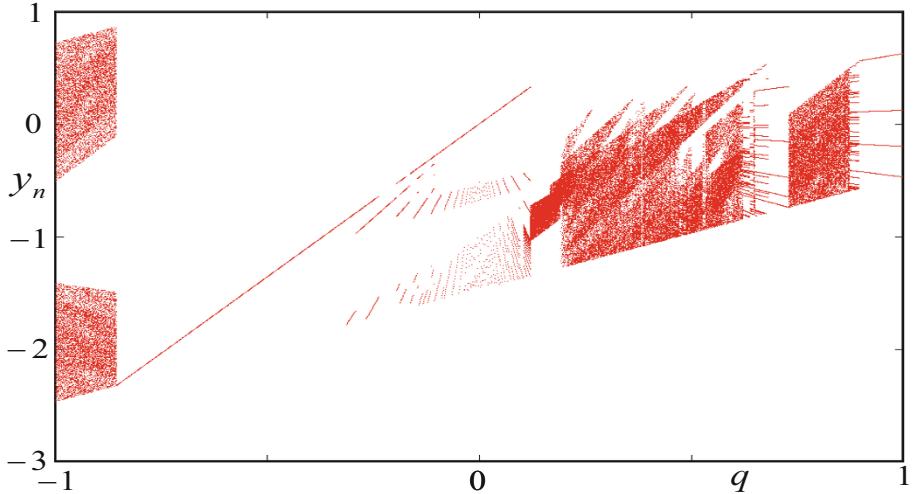


Fig. 5. Bifurcation diagram ($\delta = 0.05, p = 1, d = \pi/4$)

point. The periodic orbit is said to be unstable, stable and superstable for initial value if $|Df^k(x_p)| > 1$, $|Df^k(x_p)| < 1$ and $Df^k(x_p) = 1$, respectively, where $Df^k(x_p)$ is the slope of f^k at x_p . Superstable fixed point and periodic point are abbreviated by SSFP and SSPP, respectively. If there exists some positive integer l and $|Df^l(x_0)| > 1$ is satisfied for almost all $x_0 \in I$ then f is expanding and can not have stable periodic point. In this case f has positive Lyapunov exponent and is said to be chaotic [2].

In Fig. 4, we can see chaotic orbits that hits branches with expanding slope and SSPO that hits a branch with zero-slope. They correspond to chaotic attractor and SSPO in the phase space, respectively. Especially, the complex SSPO behaves like chaotic attractor in the phase space (e.g., Fig. 3(g)), however, it is superstable for initial point. Note that the slope on Eq. (8) can be zero for $d = \pi/4, m = 5$ and $p = 1$, (e.g., B_5 in Fig. 4(b)). Such SSPO has not been discussed in our previous papers.

Fig. 5 shows 1-parameter bifurcation diagram for increasing q that suggests a variety of complex phenomena. In this diagram, we have confirmed that the map exhibits either SSPP or chaos. We have also confirmed that the diagram is robust for small parameter perturbation. Since complete explanation is hard, we note the following points.

- Chaos is changed into SSFP at $q \simeq -0.85$. At this bifurcation point, expanding branch (B_4 is Fig. 4(a)) disappears and zero-slope branch (B_5 is Fig. 4(b)) appears.
- As q increases, the SSFP is changed into a variety of SSPPs. The SSPP can coexist with SSFP in some parameter range (Fig. 4(c)).
- The SSPP is superstable for initial state, however, it can be very sensitive for parameter (Fig. 4(d) and (e)). In such a case, the map have very complex shape with many discontinuity points and branches.

- The SSPP is changed into chaos (Fig. 4(f)) at $q \simeq 0.13$. After that, the diagram has window structure of SSPPs (Fig. 4(g)-(i)).

4 Conclusions

SKC with series connected TCSW and SCSW is analyzed in this paper. Deriving the PWL return map, we have analyzed bifurcation phenomena between chaos and SSPO. Future problems include detailed analysis of bifurcation phenomena, laboratory experiments and application to spiking neural networks.

References

1. Keener, J.P., Hoppensteadt, F.C., Rinzel, J.: Integrate-and-fire models of nerve membrane response to oscillatory input. *SIAM J. Appl. Math.* 41, 503–517 (1981)
2. Miroollo, R.E., Strogatz, S.H.: Synchronization of pulse-coupled biological oscillators. *SIAM J. Appl. Math.* 50, 1645–1662 (1990)
3. Izhikevich, E.M.: Resonate-and-fire neurons. *Neural Networks* 14, 883–894 (2001)
4. Toyoizumi, T., Aihara, K., Amari, S.: Fisher information for spike-based population coding. *Phys. Rev. Lett.* 97, 098102 (2006)
5. Lindner, B., Chacron, M.J., Longtin, A.: Integrate-and-fire neurons with threshold noise: A tractable model of how interspike interval correlations affect neuronal signal transmission. *Physical Review E* 72, 021911 (2005)
6. Nakano, H., Saito, T.: Grouping synchronization in a pulse-coupled network of chaotic spiking oscillators. *IEEE Trans. Neural Networks* 15(5), 1018–1026 (2004)
7. Izhikevich, E.M.: Weakly pulse-coupled oscillators, FM interactions, synchronization, and oscillatory associative memory. *IEEE Trans. Neural Networks* 10(3), 508–526 (1999)
8. Campbell, S.R., Wang, D., Jayaprakash, C.: Synchrony and desynchrony in integrate-and-fire oscillators. *Neural Comput* 11, 1595–1619 (1999)
9. Sushchik, M., Rulkov, N., Larson, L., Tsimring, L., Abarbanel, H., Yao, K., Volkovskii, A.: Chaotic pulse position modulation: a robust method of communicating with chaos. *IEEE Comm. Lett.* 4, 128–130 (2000)
10. Miyachi, K., Nakano, H., Saito, T.: Response of a simple dependent switched capacitor circuit to a pulse-train input. *IEEE Trans. Circuits Systs. I* 50(9), 1180–1187 (2003)
11. Kobayashi, Y., Nakano, H., Saito, T.: A Simple Chaotic Circuit with Impulsive Switch Depending on Time and State. *Nonlinear Dynamics* 44, 73–79 (2006)
12. Lasota, A., Mackey, M.C.: Chaos, Fractals, and Noise, 2nd edn. Springer, Heidelberg (1994)

Application of Higher Order Neural Network Dynamics to Distributed Radio Resource Usage Optimization of Cognitive Wireless Networks

Mikio Hasegawa^{1,2}, Taichi Takeda¹, Taro Kuroda¹, Ha Nguyen Tran¹, Goh Miyamoto¹, Yoshitoshi Murata^{3,2}, Hiroshi Harada², and Shuzo Kato²

¹ Tokyo University of Science, Tokyo 102-0073, Japan

² National Institute of Information and Communications Technology (NICT), Yokosuka 239-0847, Japan

³ Iwate Prefectural University, Takizawa 029-0193, Japan

Abstract. We propose a distributed radio access network selection method for heterogeneous wireless network environment, in which mobile terminals can adaptively and seamlessly handover among different wireless access technologies. Our algorithm optimizes fairness of radio resource usage without centralized computing on the network side. As a decentralized optimization scheme, we introduce the dynamics of the mutually connected neural network dynamics, whose energy function autonomously minimizes by distributed update of each neuron. Since the objective function of the fairness becomes a fourth-order function of the neurons' states which cannot be optimized by the conventional Hopfield neural network, we apply a neural network model extended to higher-order mutual connections and energy functions. By numerical simulation, we confirm that the proposed algorithm can optimize fairness of the throughput by distributed and autonomous computation.

1 Introduction

Various wireless networks have been developed, standardized and commercialized. For data communications, some of them have a wide service area, but low bit rate and high cost, such as the 3G cellular systems. On the other hand, some of the other systems may have small coverage area, but high bit rate and low cost, such as the wireless LAN systems. They have different characteristics and the most appropriate wireless system for the user always changes depending on the users' situation. In order to make it possible to use always the best radio access network (RAN) seamlessly, various vertical handover technologies, which enable switching of the current access network to a different type of access network without interruption of the on-going session, have been developed [12]. As a handover protocol to switch the IP address of a mobile terminal to a usable one in the network at the new location, various macro mobility technologies, such as mobile IP or SIP mobility, have been developed. To improve handover delay, various extensions of the mobile IP have also been proposed. For realizing seamless handover among different wireless access technologies, interfaces

between the upper layer handover protocols and the lower layer wireless link protocols are standardized in IEEE802.21 [3]. In order to enable selection of the most appropriate network according to the current context, a common signaling network to exchange various information between the terminal side and the network side, has been proposed as MIRAI architecture [12]. Recently, such architecture to exchange information between the network side and the terminal side to optimize radio resource usage is standardized in IEEE P1900.4 [6].

In order to optimize the total radio resource usage of entire networks, such technologies enables to utilize various kinds of radio access networks including different operators' or managers' networks by decentralized information management and decision makings [7]. We have already developed a prototype, which includes the entities and the functions of IEEE P1900.4 and showed that it adaptively switches radio access networks according to information on the network side and the terminal side by decentralized optimum decision making [8].

As a decentralized optimization algorithm, in this paper, we introduce the dynamics of the mutually connected neural network [9][10]. By autonomous and decentralized updates of each neuron, the energy function defined by fixed parameters, the connection weights between neurons and threshold of each neuron, monotonically decreases. This property of the mutually connected neural network has been applied to various combinatorial optimization problems [11]. We have already shown that such a neural network is applicable to distributed radio resource usage optimization that maximizes average throughput per user [12][13]. We have also applied it to optimize other parameters for a user-centric selection, such as communication cost and power consumption and so on [14].

In this paper, we apply the decentralized optimization dynamics of the neural network to fair radio resource selection that optimizes both at the same time, maximization of the average of available throughput per terminal and minimization of the variance of the throughput per terminal. When we design a neural network which minimizes the variance of throughput, the objective energy function becomes a fourth-order function of the neuron state. However, the energy function of the conventional Hopfield neural network is a second-order function, and cannot be applied to the optimization problem of throughput fairness. Therefore, in this paper, we introduce a more general neural network model with higher-order connections between neurons, which autonomously decreases higher-order energy function [15][16]. We show how to realize the decentralized radio resource usage optimization method based on the neural network and show the simulation results to confirm effectiveness of this approach.

2 Decentralized Optimization of Radio Resource Usage Using Mutually Connected Neural Network Dynamics

It is well known that when we mutually connect simple neurons, such as the McCulloch-Pitts neuron model and autonomously update the state of the neural network by a simple update equation as in the following equation,

$$x_{ij}(t+1) = \begin{cases} 1 & \text{for } \sum_{k=1}^{N_m} \sum_{l=1}^{N_{AP}} W_{ijkl} x_{kl}(t) > \theta_{ij}, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

the state of the neural network converges after several iterations, where $x_{ij}(t)$ is the state of the (i, j) th neuron at time t , w_{ijkl} is the connection weight between the (i, j) th and (k, l) th neurons, θ_{ij} is the threshold of the (i, j) th neuron, respectively. By updating the state of the mutually-connected neural network by the Eq. (1), the following energy function,

$$E_{NN}(t) = -\frac{1}{2} \sum_{i=1}^{N_m} \sum_{j=1}^{N_{AP}} \sum_{k=1}^{N_m} \sum_{l=1}^{N_{AP}} W_{ijkl} x_{ij}(t) x_{kl}(t) + \sum_{i=1}^{N_m} \sum_{j=1}^{N_{AP}} \theta_{ij} x_{ij}(t). \quad (2)$$

always decreases [9][10]. In Eq. (2), variable is $x_{ij}(t)$ which is updated by Eq. (1), and other parameters, w_{ijkl} and θ_{ij} , are fixed parameters. Namely, by decentralized update of each neuron by Eq. (1), this energy function can be decreased autonomously.

By using such a minimization dynamics, first, we design a neural network which optimizes the average throughput per user. For simplifying the experiments in this paper, we assume that all the terminals are communicating using a best-effort protocol such as TCP, and capacity of each access point of packet-based wireless systems is shared equally among terminals connected to the same access point. Under such an assumption, the throughput of each terminal is approximately defined by, $T_i(t) = \frac{C_{h_{\text{link}}(i)}}{N_j^{\text{AP}}(t)}$, where, $N_j^{\text{AP}}(t)$ is the number of terminals connected to the access point j , C_j is the total of the throughput which the access point j can provide, $h_{\text{link}}(i)$ is the access point which the terminal i is currently connecting. Using $T_i(t)$, the objective function $F_1(t)$ which maximizes the average throughput per user can be simply defined as follow,

$$F_1(t) = \frac{1}{N_m} \sum_{i=1}^{N_m} T_i(t) = \frac{1}{N_m} \sum_{i=1}^{N_m} \frac{C_{h_{\text{link}}(i)}}{N_{h_{\text{link}}(i)}^{\text{AP}}(t)} \quad (3)$$

where, N_m is the total number of the mobile terminals.

To solve this optimization problem by the mutually-connected neural networks, firing of each neuron is corresponded to a wireless link between a wireless access point and a mobile terminal. Fig. 1 shows the relation between the firings and the wireless links. The firing of the (i, j) th neuron corresponds to an establishment of the wireless link between the terminal i and the access point j .

By defining an objective function $F_1(t)$ by the neuron states $x_{ij}(t)$, we can obtain the connection weights and the thresholds of a neural network which autonomously optimizes the objective function. However, if we straightforwardly transform Eq. (3) to the function of the state of neurons $x_{ij}(t)$ comes to the denominator of the function, which cannot be transformed to the form of the energy function as in Eq. (2). For having suitable form of the objecting function for the optimization method based on the mutually connected neural network, the average

throughput maximization problem in Eq. (3) is replaced to an equivalent problem that is minimization of summation of inverse of the throughput. This objective energy function $E_1^{\text{OBJ}}(t)$, which autonomously minimizes by the neural network but it maximizes average throughput per user, is defined as follows,

$$E_1^{\text{OBJ}}(t) = \sum_{i=1}^{N_m} \frac{1}{T_i(t)} = \sum_{i=1}^{N_m} \frac{N_{h_{\text{link}}(i)}^{\text{AP}}(t)}{C_{h_{\text{link}}(i)}} \quad (4)$$

Using the neuron state $x_{ij}(t)$, we can obtain a following equation.

$$E_1^{\text{OBJ}}(t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_{\text{AP}}} \sum_{k=1}^{N_m} \sum_{l=1}^{N_{\text{AP}}} \frac{1}{C_j} (1 - \delta_{ij}) x_{ij}(t) x_{kj}(t) + \sum_{i=1}^{N_m} \sum_{j=1}^{N_{\text{AP}}} \frac{1}{C_j} x_{ij}(t), \quad (5)$$

where δ_{ij} is the Kronecker delta, $\delta_{ij} = 1$ for $i = j$, and $\delta_{ij} = 0$ for $i \neq j$. By comparing Eq. (5) with Eq. (2), the connection weights and the threshold can be obtained as follows,

$$W_{ijkl}^A = -2 \frac{1}{C_j} (1 - \delta_{ik}) \delta_{jl}, \quad (6)$$

$$\theta_{ij}^A = \frac{1}{C_j}. \quad (7)$$

Using these mutual connections and threshold, the neural network updated by Eq. (1) maximizes the average throughput autonomously without any centralized computation.

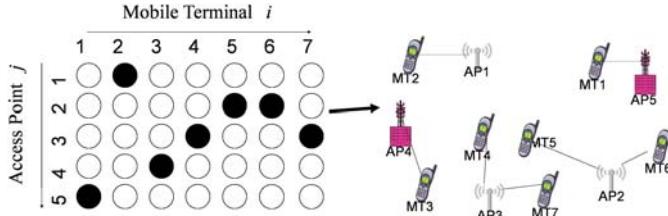


Fig. 1. Relation between firings of the neurons and establishments of the wireless links

3 Optimizing Fairness of Throughput Using Higher Order Neural Network Dynamics

Although the average throughput can be maximized by the neural network in the previous section, fairness of the throughput among the terminals is not taken into account. In this section, we extend this framework to decentralized optimization of fairness using the higher-order neural network.

First, we need to define an objective function for fairness which should be minimized. In the previous section, we introduced the inverse of the throughput

$\frac{1}{T_i(t)}$ into the objective energy function of average throughput maximization. Using , the objective energy function for optimization of fairness of the throughput can be defined by the following equation for example, $F_2(t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} (\frac{1}{T_i(t)} - \frac{1}{T_k(t)})^2$ which is minimization of difference of the throughput per user. However, if we use this equation for the objective function, big improvement of throughput degrades this objective function $F_2(t)$. As a result, it becomes difficult to perform minimization of variance and maximization of average at the same time.

In order to avoid such inconsistency, we introduce the inverse of the capacity of each access points as the following function,

$$R_i(t) = \frac{1}{C_{h_{\text{link}}(i)}} \frac{1}{N_{h_{\text{link}}(i)}^{\text{AP}}(t)}, \quad (8)$$

and define the objective function for the fairness by minimization of the differences of $R_i(t)$ as follows,

$$\begin{aligned} E_2^{\text{OBJ}}(t) &= \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} \left(\frac{1}{R_i(t)} - \frac{1}{R_k(t)} \right)^2 \\ &= \sum_{i=1}^{N_m} \sum_{k=1}^{N_m} \sum_{m=1}^{N_m} \sum_{j=1}^{N_m} \sum_{o=1}^{N_{\text{AP}}} \sum_{l=1}^{N_{\text{AP}}} C_j C_l (x_{mj} x_{ij} x_{ol} x_{il} \\ &\quad - 2x_{mj} x_{ij} x_{ol} x_{kl} + x_{mj} x_{kj} x_{ol} x_{kl}) \end{aligned} \quad (9)$$

In Eq. (9), we obtained a forth-order equation of the neuron state. It means that we cannot apply the conventional Hopfield neural network for this function, since its energy function in Eq. (2) is a second-order function.

To minimize the fourth-order energy function for optimization of fairness in Eq. (9), we define the following third-order neuronal update equation, using θ_{ij}^B , W_{ijkl}^B , V_{ijklmn}^B and $U_{ijklmnp}^B$ as the threshold and the first to the third order connection weights, respectively, as follows,

$$x_{ij}(t+1) = \begin{cases} 1 & \text{if } \frac{1}{6} \sum_{k=1}^{N_m} \sum_{l=1}^{N_{\text{AP}}} \sum_{m=1}^{N_m} \sum_{n=1}^{N_{\text{AP}}} \sum_{o=1}^{N_m} \sum_{p=1}^{N_{\text{AP}}} U_{ijklmnp}^B x_{kl}(t) x_{mn}(t) x_{op}(t) \\ & + \frac{1}{2} \sum_{k=1}^{N_m} \sum_{l=1}^{N_{\text{AP}}} \sum_{m=1}^{N_m} \sum_{n=1}^{N_{\text{AP}}} V_{ijklmn}^B x_{kl}(t) x_{mn}(t) \\ & + \sum_{k=1}^{N_m} \sum_{l=1}^{N_{\text{AP}}} W_{ijkl}^B x_{kl}(t) > \theta_{ij}^B, \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

The energy function of this neural network with the third-order update equation becomes the following fourth-order function of the neuron states,

$$E_2^{\text{OBJ}} = -\frac{1}{24} \sum_{i=1}^{N_m} \sum_{j=1}^{N_{\text{AP}}} \sum_{k=1}^{N_m} \sum_{l=1}^{N_{\text{AP}}} \sum_{m=1}^{N_m} \sum_{n=1}^{N_{\text{AP}}} \sum_{o=1}^{N_m} \sum_{p=1}^{N_{\text{AP}}} U_{ijklmnp}^B x_{ij}(t) x_{kl}(t) x_{mn}(t) x_{op}(t)$$

$$\begin{aligned}
& -\frac{1}{6} \sum_{i=1}^{N_m} \sum_{j=1}^{N_{AP}} \sum_{k=1}^{N_m} \sum_{l=1}^{N_{AP}} \sum_{m=1}^{N_m} \sum_{n=1}^{N_{AP}} V_{ijklmn}^B x_{ij}(t) x_{kl}(t) x_{mn}(t) \\
& -\frac{1}{2} \sum_{i=1}^{N_m} \sum_{j=1}^{N_{AP}} \sum_{k=1}^{N_m} \sum_{l=1}^{N_{AP}} W_{ijkl}^B x_{ij}(t) x_{kl}(t) - \sum_{i=1}^{N_m} \sum_{j=1}^{N_{AP}} \theta_{ij}^B x_{ij}(t)
\end{aligned} \tag{11}$$

By comparing Eqs. (9) and (11), we can obtain the connection weight parameters, $U_{ijklmnop}^B$, V_{ijklmn}^B , W_{ijkl}^B and θ_{ij}^B , for optimization of the fairness as follows,

$$\begin{aligned}
U_{ijklmnop}^B = & -[2(\delta_{ik}-1)C_i C_l + 2(\delta_{io}-1)C_i C_p + 2(\delta_{km}-1)C_l C_n \\
& + 2(\delta_{mo}-1)C_n C_p] \delta_{jn} \delta_{lp} - [2(\delta_{im}-1)C_j C_n + 2(\delta_{io}-1)C_j C_p \\
& + 2(\delta_{km}-1)C_l C_n + 2(\delta_{ko}-1)C_l C_p] \delta_{jl} \delta_{np} - [2(\delta_{ik}-1)C_j C_l \\
& + 2(\delta_{im}-1)C_j C_n + 2(\delta_{ko}-1)C_l C_p + 2(\delta_{mo}-1)C_n C_p] \delta_{jp} \delta_{ln}], \tag{12}
\end{aligned}$$

$$\begin{aligned}
V_{ijklmn}^B = & -[C_j C_l \{\delta_{im} \delta_{in} + \delta_{km} \delta_{jn} - 5(\delta_{jn} + \delta_{ln}) + 3(\delta_{ik} \delta_{jn} + \delta_{ik} \delta_{ln}) \\
& - 10 \delta_{jl} \delta_{jn}\} + C_j C_n \{\delta_{ik} \delta_{ln} + \delta_{km} \delta_{jl} - 5(\delta_{jl} + \delta_{ln}) + 3(\delta_{im} \delta_{jl} \\
& + \delta_{im} \delta_{ln}) - 10 \delta_{jl} \delta_{jn}\} + C_l C_n \{\delta_{ik} \delta_{jn} + \delta_{im} \delta_{jl} - 5(\delta_{jl} + \delta_{jn}) \\
& + 3(\delta_{km} \delta_{jl} + \delta_{km} \delta_{jn}) - 10 \delta_{ji} \delta_{jn}\}], \tag{13}
\end{aligned}$$

$$W_{ijkl}^B = -2C_j C_l (16 \delta_{jl} - 2 \delta_{ik} + 2), \tag{14}$$

$$\theta_{ij}^B = -10C_j. \tag{15}$$

4 Neumerical Experiment

In this paper, we have introduced two types of the energy functions, E_1^{OBJ} for maximization of the average throughput per user, and E_2^{OBJ} for optimizing fairness of the available throughput among users which minimize differences of throughput among the terminals. The total energy function to optimize both energy functions at the same time is defined as follow,

$$E_{\text{TOTAL}} = A E_1^{\text{OBJ}} + B E_2^{\text{OBJ}}, \tag{16}$$

where A and B are the weight parameters for each objective functions. Then, the connection weights and the threshold to minimize E_{TOTAL} becomes as follows,

$$U_{ijklmnop}^{\text{TOTAL}} = B U_{ijklmnop}^B, \tag{17}$$

$$V_{ijklmn}^{\text{TOTAL}} = B V_{ijklmn}^B, \tag{18}$$

$$W_{ijkl}^{\text{TOTAL}} = A W_{ijkl}^A + B W_{ijkl}^B, \tag{19}$$

$$\theta_{ij}^{\text{TOTAL}} = A \theta_{ij}^A + B \theta_{ij}^B. \tag{20}$$

For the update equation, we introduce the following winner take all neuron [17], since we assume, in this paper, each mobile terminal have a wireless link with one access point,

$$x_{ij}(t+1) = \begin{cases} 1 & \text{if } y_{ij}(t+1) = \max\{y_{i1}(t+1), \dots, y_{iN_{AP}}(t+1)\}, \\ 0 & \text{otherwise,} \end{cases} \tag{21}$$

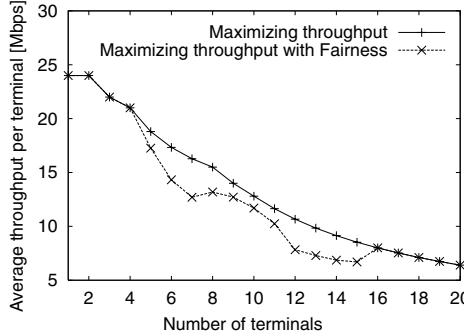


Fig. 2. The average throughput per user of the neural network-based decentralized radio resource selection with and without higher-order mutual connections which optimizes fairness

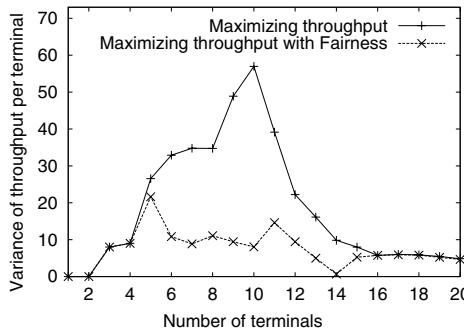


Fig. 3. Variance of the throughput per user of the neural network-based decentralized radio resource selection with and without higher-order mutual connections which optimizes fairness

where,

$$\begin{aligned}
 y_{ij}(t+1) = & \frac{1}{6} \sum_{k=1}^{N_m} \sum_{l=1}^{N_{AP}} \sum_{m=1}^{N_m} \sum_{n=1}^{N_{AP}} \sum_{o=1}^{N_m} \sum_{p=1}^{N_{AP}} U_{ijklmnp}^{\text{TOTAL}} x_{kl}(t) x_{mn}(t) x_{op}(t) \\
 & + \frac{1}{2} \sum_{k=1}^{N_m} \sum_{l=1}^{N_{AP}} \sum_{m=1}^{N_m} \sum_{n=1}^{N_{AP}} V_{ijklmn}^{\text{TOTAL}} x_{kl}(t) x_{mn}(t) + \sum_{k=1}^{N_m} \sum_{l=1}^{N_{AP}} W_{ijkl}^{\text{TOTAL}} x_{kl}(t) - \theta_{ij}^{\text{TOTAL}},
 \end{aligned} \tag{22}$$

When $A > 0$ and $B = 0$, it has only the first-order mutual connections and is the same as the algorithm in Sec. 2, that maximizes the average throughput per user.

Figs. 2 and 3 show the result of the algorithms without the higher-order mutual connections ($A = 1$ and $B = 0$, corresponding to only maximization of the throughput), and the one with the higher-order mutual connections ($A = 1$

and $B = 0.000045$, corresponding to both maximization and fairness of the throughput). From Fig. 3, it is clear that the algorithm which optimizes fairness could reduce the various of the throughput. From the both figures, we could confirm that that the variance of the throughput could be reduced in Fig. 3, but the average of the throughput was not degraded much as in Fig. 2. By tuning the parameter B , we can change the balance of effects for maximization and fairness.

5 Conclusion

In this paper, we have applied autonomous and decentralized dynamics of the higher-order neural networks to optimize radio resource selection with fairness. Although the conventional Hopfield neural network was not applicable to the problems which have higher order objective functions, the proposed approach can take into account almost any kind of objective functions, which can be minimized without centralized calculations.

In a real situation, this kind of decentralized selection can be implemented by various ways in a distributed architecture. Computations can be distributed into the access points, the servers in each RAN, terminals, and so on, for optimization of the total system.

References

1. Wu, G., Havinga, P., Mizuno, M.: IEEE Comm. Mag., 126–134 (2002)
2. Inoue, M., Mahmud, K., Murakami, H., Hasegawa, M., Morikawa, H.: IEEE Wireless Commun. 11, 56–63 (2004)
3. Lampropoulos, G., Salkintzis, A., Passas, N.: IEEE Communications Magazine 46 (2008)
4. Hasegawa, M., Inoue, M., Morikawa, H.: Proc. of Intl. Sympo. on Wireless Personal Multimedia Communications, 359–342 (2005)
5. Cordier, P., et al.: E2R, IST summit 2006 (2006)
6. Buljore, S., et al.: IEICE Trans. on Communications 91-B, 2–9 (2008)
7. Harada, H., Murakami, H., Ishizu, K., Filin, S., Saito, Y., Tran, H., Miyamoto, G., Hasegawa, M., Murata, Y., Kato, S.: Proc. IEEE Globecom (2007)
8. Ishizu, K., Murakami, H., Miyamoto, G., Tran, H., Filin, S., Hasegawa, M., Murata, Y., Harada, H., Kato, S.: SDR workshop (2008)
9. Hopfield, J.J.: Proc. Natl. Acad. Sci. USA 79, 2554–2558 (1982)
10. Hopfield, J.J.: Proc. Natl. Acad. Sci. USA 81, 3088–3092 (1984)
11. Hopfield, J.J., Tank, D.W.: Biological Cybernetics 52, 141–152 (1985)
12. Hasegawa, M., Tran, H., Miyamoto, G., Murata, Y., Harada, H., Kato, S.: IEICE Trans. of Communications, 91-B, 110–118 (2008)
13. Hasegawa, M., Tran, H., Miyamoto, G., Murata, Y., Kato, S.: Proc. IEEE Intl. Sympo. on Personal, Indoor and Mobile Radio Communications (PIMRC) (2007)
14. Hasegawa, M., Tran, H., Miyamoto, G., Murata, Y., Harada, H., Kato, S.: Proc. of IEEE Wireless Communication and Network Conference (WCNC) (2008)
15. Cooper, B.S.: Proc. of IEEE International Conference on Neural Networks, 1855–1860 (1995)
16. Matsuda, S.: IEEE Trans. on Neural Networks, 9, 1319–1330 (1998)
17. Takefuji, Y., Lee, K., Aiso, H.: Biological Cybernetics, 67 (1992)

Synchronized Rhythmic Signals Effectively Influence Ongoing Cortical Activity for Decision-Making: A Study of the Biological Plausible Neural Network Model

Hiroaki Wagatsuma and Yoko Yamaguchi

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

Abstract. The brain is capable of parallel processing of different types of information in different brain regions. However, for higher cognitive functions such as decision-making, such regionally distributed information must interact together in the proper timing. The prefrontal cortex, the region which carries out decision-making, needs to be modulated by external signals to represent the current behavioral context in the hippocampus (HP). The question remains as to how the firing activity of the cortical neural network can be modulated by external signals in corporation with the ongoing activity. We hypothesized that rhythmic signals that attempt to synchronize the cortical ongoing activity minimize the disturbance and effectively enhance the activities of selective neurons. We investigated the level of the modulation by using a mutually connected neural network that consists of a neuron model with excitatory and refractory periods. The results demonstrated that cortical ongoing activities are weakly modulated by random external signals, while synchronized rhythmic signals, given as the pseudo HP signals, selectively enhance cortical activities. This suggests that the cortical ongoing activity is effectively influenced by the synchronized signals, which carry information in the proper timing of excitation. The investigation of neural synchronization dynamics is important to understanding how the brain realizes parallel processing in different sub-regions and to update immediately the internal representation even if the previous internal processing is ongoing.

1 Introduction

The brain has the capability for multimodal sensory perceptions and multiple action controls. The information processing for receiving sensory stimuli and sending action signals is carried out separately but in parallel in different brain sub-regions. Yet the binding together of such different information in time is a crucial cognitive function of the brain for achieving consistent sensation and motion. The key question is how the brain coordinates different sub-regions for the necessary timely cross-interaction, irrespective of the fact that activities in different regions are independently ongoing.

Neural synchronization, which is observed as the repetitive synchronized activity of multiple neurons, is a remarkable candidate to realize the dynamic coupling not only between different neurons in the same region [1] but also between different regions.

Recent experimental evidence from rat electrophysiological studies has shown this capability in the relationship between the prefrontal cortex and the hippocampus [2]. The prefrontal cortex is known to govern decision-making, and the hippocampus is necessary for the encoding of spatial information in the current behavioral context. In spatial navigation tasks, such as running in a maze, the hippocampus monitors one's location in the environment, and the prefrontal cortex is required to switch the current behavior to the next behavior, depending on one's location. The difficulty is in understanding how the prefrontal cortex receives necessary spatial information from the hippocampus immediately before the decision making point.

In the case of the digital computer, CPU and peripheral devices can interact together by using the same clock rate. Some brain regions have an internal pacemaker, which is observed as the oscillation in the local field potential (LFP), but most cortical regions do not have sustained internal rhythms. The hippocampus has a sustained LFP oscillation, especially when the animal is running, and the activities of the neurons are strongly modulated by the theta rhythm. In contrast, prefrontal neurons transiently exhibit some degree of synchronized activity, such as spindle oscillations, under limited conditions [3], suggesting individual neurons have a property of being oscillator. Interestingly, the experimental data reported that the coherence between the prefrontal firing timing and the hippocampal theta rhythm significantly increases when the animal passes through the branching point of the T-shape spatial maze. This fact suggests that the synchrony between activities in the two regions is enhanced for information transfer when making a decision to go down either branch.

The question of the present paper is how the population activity of cortical neurons, accompanied by ongoing changes, is modulated by the external signal without an explicit internal rhythm. We hypothesized here that the ongoing cortical activity is effectively modulated by rhythmic signals, enabling it to synchronize the cortical transient rhythm. This indicates that the independent rhythmic signal is not sufficient to change the ongoing cortical activity; however, if the rhythmic signal is synchronized, the cortical activity smoothly changes. We simply assume that the prefrontal cortex is a mutually connected neural network that is arranged in an anatomical space. According to the distribution of synaptic connections in the space, a spontaneous activity is easily sustained through a chain reaction by stimulating inactive neurons via other active neurons. We investigated how such ongoing spontaneous activity is controlled by giving various external signals. Section 2 describes the fundamental property of the cortical neuron model. Section 3 describes the results of computer experiments with various types of the external signals, showing the level of the modulation and the behavior of the population activity when receiving the external signal. Finally, section 4 presents the conclusions of this work.

2 Cortical Neural Network

In this section, we first define the dynamics of the neuron model and the connectivity that we use in the present paper. We simply assume a cortical network model to be a neural network with properties to sustain a spontaneous activity, showing a chain reaction by stimulating inactive neurons via other active neurons (Fig. 1A), typically known a synfire chain [4]. In experimental studies, cortical slices exhibit repeatedly sequential activities after minutes, so-called cortical songs [5], and have a long lasting

population activity in hours, known as neural avalanches [6]. For the sake of a simple reproduction of the propagating population activity, we assume virtual-anatomical positions of neurons in the two dimensional plane, $[x] \times [y] = [-0.5, 0.5] \times [0, 1]$, the neurons are randomly set in the plane, and the connectivity is given by spreading distributions of excitatory and inhibitory connections toward the propagating direction. The dynamics of the internal variable ϕ_i of a cortical neuron indexed with i , which provides the membrane potential as $\cos(\phi_i)$, is described by a differential equation,

$$\frac{d\phi_i}{dt}(t) = \omega_0 + \left\{ \beta_0 - I_i(t) - \sum_j (w_{ij}^+ + w_{ij}^-) f_j(t) \right\} \sin \phi_i(t), \quad (1)$$

$$f_i(t) = 0.5 \tanh(50(\cos(\phi_i) - 0.9)) + 0.5, \quad (2)$$

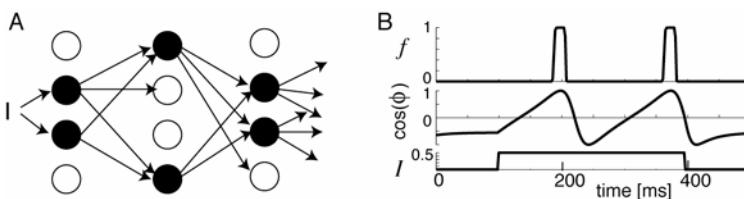


Fig. 1. (A) Schematic illustration of a chain reaction in the neural network and (B) the excitability of the neuron with parameters: $\omega_0 = 1$, $\beta_0 = 1.2$, $I = 0.5$ in $[100, 400]$ [ms]

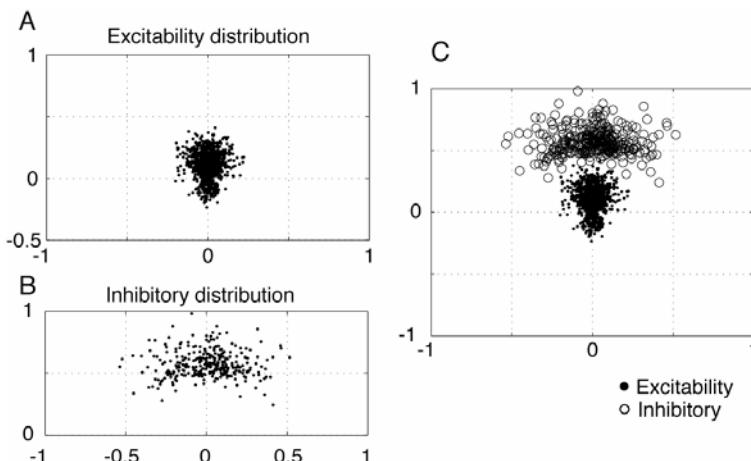


Fig. 2. Example of distributions of excitatory (A) and inhibitory (B) connections of a neuron. The origin represents the position of the pre-synaptic neuron and dots denote candidate of post-synaptic (target) neurons' position. From one neuron, ten target neurons are randomly selected in the distribution A for excitatory connections and five target neurons are randomly selected in the distribution B for inhibitory connections. (C) The positional relationship between A and B is shown as a superimposed image. Neighboring excitatory connections provide a concentrated population activity and surrounding inhibitory connections maintains the isolation, called active packet typically derived from Mexican-hat-type synaptic connections.

where ω_0 is the intrinsic time constant, which determines the refractory period (Fig. 1B), β_0 is the stabilization coefficient to satisfy $\beta_0 > \omega_0$ for staying the resting state if input I_i is zero. The spike activity f_i is defined by a sigmoid function. We assume that excitatory and inhibitory synaptic weights, w_{ij}^+ and w_{ij}^- , so that positions of post-synaptic (target) neurons are spatially and directionally distributed from the position of the pre-synaptic neuron, as shown in Fig. 2. The target neurons are selected by random variables: the distance of the pre-synaptic neuron r^D and the angle θ^D where $D = \{+, -\}$. Probability density functions (PDF) of r^+ , θ^+ , θ^- are respectively given as normal distributions with means $\mu = 0.1, 0, 0$ and variances $\sigma = 0.1, 0.1\pi, 0.1\pi$. The PDF of r^- is given as the exponential distribution with mean $\mu = 0.1$, and the positional shift $s = 0.5$ is added in all the y-coordinate. This neural network is expected to reproduce a traveling population activity, which starts around the origin in the virtual-anatomical plane, by giving a trigger input.

Output for Decision-making (action selection). For the evaluation how much activity is modulated by the external inputs, we define two types of output in this network: the Right output and the Left output. In this evaluation we reduced decision-making output to a binary selection of active neurons. The two dimensional plane is separated in the center, and the Right and Left outputs are given activity summations in the area $[0, 0.5] \times [0.2, 1]$ and in the area $[-0.5, 0] \times [0.2, 1]$ respectively (Fig. 3). Since the virtual-anatomical positions and the synaptic connections are uniformly distributed in the horizontal direction, spontaneous activity tends to propagate in the straight forward without external inputs. The propagation represents an ongoing cortical activity corresponding to the internal sub-process in the cortex. If external signals come into the either side of neurons, the coming input triggers the neurons in the corresponding side and then the population activity in the side is enhanced, rather than the opposite side, if it is given in the proper timing in relation to the current ongoing activity. In addition, for the preservation of a long lasting activity without external inputs, we added supplemental excitatory connections to connect from neurons near the upper edge to neurons around the origin, and inhibitory connections mutually connected between neurons near the right and the left edges for competitive dynamics.

Method of Computer Experiments. The neural network model is described by a set of differential equations, which are numerically integrated using the Runge-Kutta method. In this experiment, we simply assume that the intrinsic cycle of cortical neurons is 125ms, e.g. 8Hz. Note that we used the excitability state of the neuron.

3 Experimental Results

Firstly, we investigated the capability of spontaneous activity in the present network. By giving a trigger input in a short period, for example in 30ms, the network can reproduce a propagating population activity. The reproduction of the spontaneous activity propagating from the bottom to the top in the virtual-anatomical plane is successfully demonstrated as shown in Fig. 3.

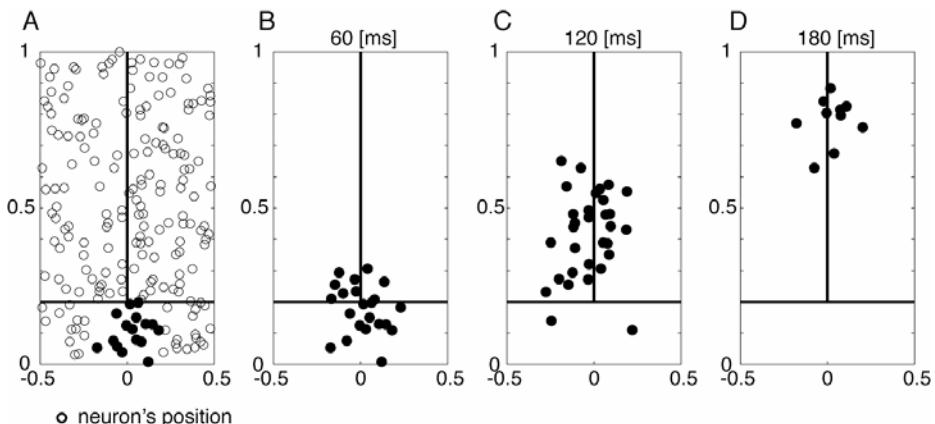


Fig. 3. The time evolution of the population activity. (A) The arrangement of neurons that defined by uniform random variables. Filled circles denote neurons receiving initial trigger inputs located around the origin. (B-D) The time evolution of active neurons in every 60ms. A cluster of activity, active packet, is moving toward to the upper area, along the central line. Solid lines represent the separation between the Right-output the Left-output areas.

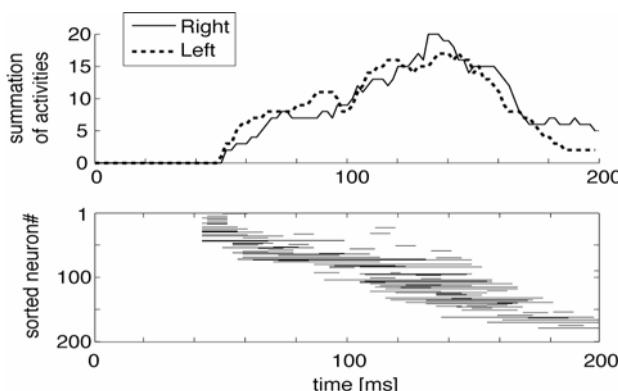


Fig. 4. (Top) Time evolutions of summation of activities of the Right-output area and the Left output area. Neurons that are uniformly arranged in both sides provide a similar change of activities in both sides. (Bottom) The time evolution of spike activity f_i of all neurons in period $[0, 200\text{ms}]$ ($i = 1, \dots, N = 200$). The propagation is observed in the shift of active neurons in accordance with the axis of the neuron index that is sorted out according to the y-coordinate in the virtual-anatomical plane.

Secondary, before the investigation, we prepare five types of external signals to put the cortical network: 1) Poisson random input (Fig. 5, top; Fig. 6A), 2) Rhythmic random input (Fig. 5, bottom; Fig. 6B), 3) Pseudo HP input with the LFP oscillation close to the refractory period of cortical neurons and with random projection from the HP network (Fig. 6C), 4) Pseudo HP input with the LFP oscillation close to the

refractory period of cortical neurons and with structurally connected projection from the HP network (Fig. 6D), 5) Pseudo HP input with the LFP oscillation twice faster than the refractory period of cortical neurons and with random connections (Fig. 6E).

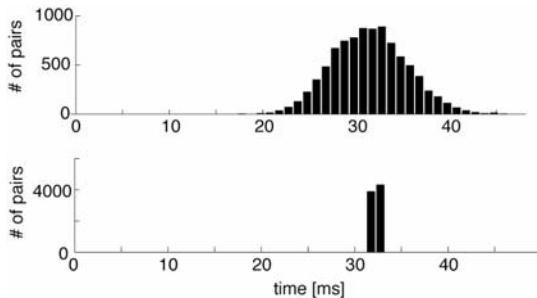


Fig. 5. Distributions of inter-spike-intervals (ISI) in random inputs, in the condition of the poisson random input (Top) and the rhythmic random input (Bottom)

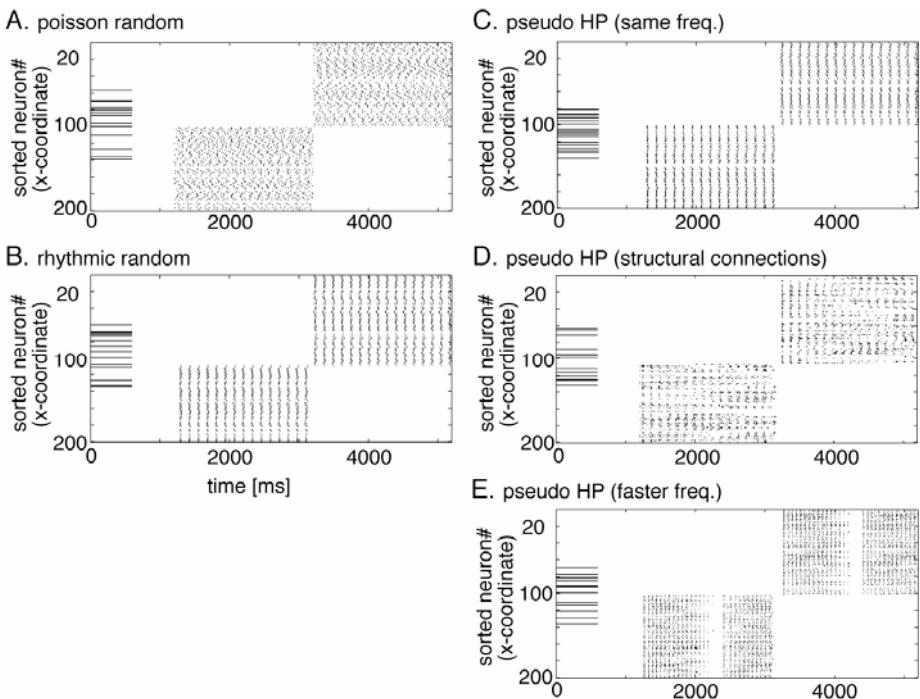


Fig. 6. Example of input patterns in the whole experimental period [0, 5200] [ms]. (A) Poisson random input and (B) rhythmic random input are given to the cortical network independent of the ongoing activity (passive mode). (C-E) pseudo HP inputs are given to the cortical network accompanied with the modulation of the membrane potential of cortical neurons by the LFP oscillation of the HP signals (synchronize mode). Trigger inputs are given in [0, 600]. Right- and Left-inputs are separately given in [1200, 3200] and in [3200, 5200] [ms].

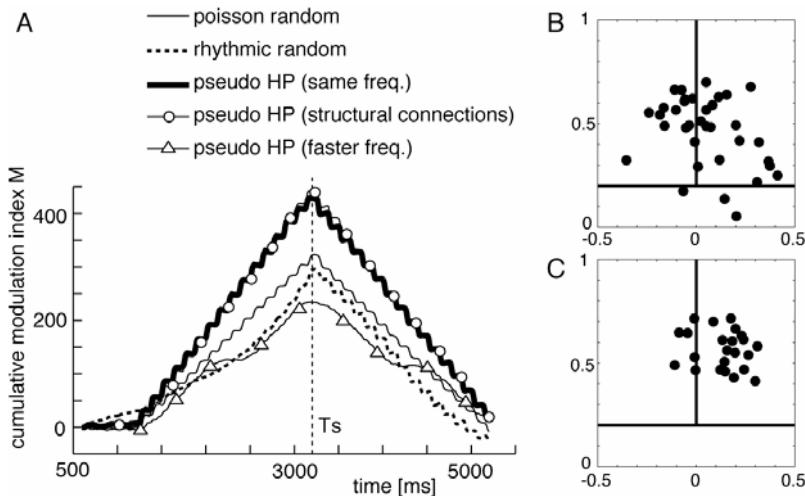


Fig. 7. (A) Statistical analyses of the cumulative modulation index $M(t)$. Ascending of the index represents that the network decided to go to the right direction continuously, while descending of the index represents that the network decided to go to the left direction continuously. T_s denotes the switching time-point of external signals: neurons in the Right area → neurons in the Left area (see, Fig. 6). Each plot is the average of 50 data sets. In third and fourth conditions, plotting curves have the highest hill, which means they are sensitive to external signals even though a propagating activity is ongoing. (B, C) Typical firing patterns in the poisson random condition (B) and in the pseudo HP with the same frequency (B) in 1729[ms]. In B, external signals tend to add firings in the right, while the cluster is keeping in C.

For comparison of the degree of the external-signal-modulation, we used the cumulative modulation index $M(t)$, which is defined as $M(t) = \int_{t=500}^{5200} P(t) \cdot \sin(\bar{\theta}(t))$, $\bar{\theta}(t) = \pi(L_{out} - R_{out}) / (L_{out} + R_{out})$ where $P(t) = R_{out} + L_{out}$, $R_{out} = \sum_{j \in R} f_j$, $L_{out} = \sum_{j \in L} f_j$ and showed the experimental result in Fig. 7A. This index indicates the amount of the difference between activities of neurons in the R-output area R_{out} and the Left-output area L_{out} , in the form of the turning-direction $\bar{\theta}$. In conditions 3)-5), investigating the efficacy of the rhythmic synchronization, the HP network are simulated by another neural network model with the mechanism of the theta phase precession described in Wagatsuma and Yamaguchi [7]. In the fourth condition, the structural connections is defined so that the temporal firing pattern in the HP is directionally enhances cortical neurons according to the upward in the y-coordinate of the virtual-anatomical plane. The effectiveness of these conditions was evaluated in Fig. 7A. Interestingly, the rhythmic synchronization seems to have the advantage of keeping the original propagation cluster during the external modulation, rather than others.

4 Concluding Remarks

In the computer experiments, our results demonstrated that cortical ongoing activities are weakly modulated by the random external signals and independent rhythmic

signals, while synchronized rhythmic signals, given as the pseudo HP signals, effectively modulate the current cortical activity. In the condition of the pseudo HP signals with a two-fold frequency's rhythm, which means that the temporal structure is the same as the original HP signals but it is too much faster for the synchronization, the level of the modulation is weaker than that in the random signals. It indicates that the successful coupling between two regions is given by the synchronization, so that activities in the selective neuron group are effectively enhanced in the proper timing of the intra-cellular cycle, which has excitatory and refractory periods. We simplified the cortical region as a neural network that consists of the simple neuron model with the same time constant. Indeed, anatomically, the region has excitatory and inhibitory neurons with a variety of time constants. The mechanism for generating the transient cortical oscillation is still unclear, but the complexity not only in connectivity but also in time constant may give us a key to understand the mechanism. The nature of cortical intrinsic propagations that supported by the synaptic weight distribution could be a further analysis. Thus the relationship between the synaptic property and the degree of synchronization connecting different regions is important in comparing with other proposed models, which relevant to the synfire chain and the bird song generation mechanism. The argument of how the decision-making process is carried out in the prefrontal cortex is further considerable, which might be beyond the selective dynamics of active neurons. It might be clarified when the proposed neural network model is testified in the specific task interacting with the environment.

Acknowledgments. This work was supported by JSPS KAKENHI (19650073).

References

1. Fries, P., Roelfsema, P.R., Engel, A.K., König, P., Singer, W.: Synchronization of oscillatory responses in visual cortex correlates with perception in interocular rivalry. *Proc. Natl. Acad. Sci. USA.* 94(23), 12699–12704 (1997)
2. Jones, M.W., Wilson, M.A.: Phase precession of medial prefrontal cortical activity relative to the hippocampal theta rhythm. *Hippocampus* 15(7), 867–873 (2005)
3. Siapas, A.G., Lubenov, E.V., Wilson, M.A.: Prefrontal phase locking to hippocampal theta oscillations. *Neuron* 46(1), 141–151 (2005)
4. Abeles, M.: *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, Cambridge (1991)
5. Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D., Yuste, R.: Synfire chains and cortical songs: temporal modules of cortical activity. *Science* 304(5670), 559–564 (2004)
6. Beggs, J.M., Plenz, D.: Neuronal avalanches in neocortical circuits. *J. Neurosci.* 23(35), 11167–11177 (2003)
7. Wagatsuma, H., Yamaguchi, Y.: Cognitive map formation through sequence encoding by theta phase precession. *Neural Computation* 16, 2665–2697 (2004)

Synchronization Transition in a Pair of Coupled Non-identical Oscillators

Yasuomi D. Sato¹, Yuji Tanaka², and Masatoshi Shiino²

- ¹ Frankfurt Institute for Advanced Studies (FIAS), Johann Wolfgang Goethe University, Ruth-Moufang-Str. 1, 60438, Frankfurt am Main, Germany
² Department of Physics, Faculty of Science, Tokyo Institute of Technology, 2-12-1 Oh-okayama, Meguro-ku, Tokyo, 152-0033, Japan

Abstract. We study synchronization phenomena in a pair of integrate-and-fire (IF) oscillators with the width of an action potential. They have slightly different periodic firings each other. Such non-identical IF oscillators typically interact with each other via excitatory or inhibitory synaptic stimuli. A return map analysis gives us a systematic analysis of stabilities of the 1 : 1 phase locking states in the pair, in terms of a synaptic decaying parameter. We demonstrate transitions of the phase lockings with a small difference in their firing frequencies and different width of an action potential. Spike timing histograms are also observed in the pair with dynamical noise. The slight shifts of a synchronous state effectively induces distribution shifts in the spike timing histogram. This agrees with results observed in physiological experiments.

1 Introduction

Rhythmic phenomena in neuronal assemblies such as gamma oscillations are very important factors to understand mechanisms how visual information processing can be achieved in the brain [1]. It would be necessary for us to understand mechanisms to generate rhythmic oscillations. The classic approach to understand them is a mathematical analysis on synchronization phenomena in a pair of coupled neural oscillator systems. Many researchers have given synchronization scheme of the pair system in terms of synaptic response, taking into careful consideration about spike emission properties such as the width of an action potential [2][3][4][5]. One of the most interesting observation is that a stability of 1 : 1 synchronization in the pair gradually breaks down with synaptic stimuli. Physicists demonstrated such breaking process in two *identical* oscillators via synaptic propagation delays [6][7]. The process was also accomplished in two *non-identical* oscillators coupled by exponentially relaxing excitatory stimuli [8].

Experimental studies on a two-neuron system have been employed to investigate observable voltage dynamics [9][10][11]. The neuronal and synaptic properties underlying synchronous behavior are being illuminated. Synchronous behavior of coupled pairs of real neurons have been observed under dynamic patch clamp techniques [12][13]. Mutual excitation and mutual inhibition respectively produce synchrony (0.0 or ± 1.0 phase lags on the unit cycle) and antisynchrony (phase

lag of ± 0.5). More interestingly, spike timing histograms between two neurons were reported as follows: For synchrony, we could not observe the precise synchrony, in spite of using an qualitatively *identical* pair. The peaks around ± 1.0 are not equal, and the additional histogram peak is not precisely centred at 0.0, but deviated slightly from there. For antisynchrony, the histogram peaks are not distributed exactly at ± 0.5 , and are also shifted.

Some researchers suggested significant effects of current noise on histogram peak shifts [13]. However, there is still ample room for analytical discussions about the phase difference distribution shifts. In this paper, we attempt to focus on mechanisms of how the histogram peak can be shifted for firing dynamics with (or involving) two types of noise for firing dynamics: firing frequency difference and current noise. In particular, the first type of noise implies infinitesimal different firing period among the pair, and will explain transition of the 1 : 1 synchronization to another state in a pair without current noise. Showing responsibilities of an action potential width for occurrence of synchronous behavior, we have found the different firing periods of the pair play an essential role in the mechanisms on histogram peak shifts.

2 A Return Map Analysis for a Pair of Coupled Oscillators

We shall begin by describing two interconnected *non-identical* oscillators of the double integrate-and-fire (DIF) type [5,14] with excitatory or inhibitory synaptic couplings and current noise ξ_i with the standard deviation σ^2 . The DIF model represented here is useful to investigate influence of the different width of an action potential on synchronous behavior. The equation for the membrane potential x_i is:

$$\frac{dx_i}{dt} = R_i(x_i) + I_{syn}^{(i)} + \xi_i(t) \quad (i = 1, 2), \quad (1)$$

where

$$R_i(x_i) = \begin{cases} X_1 - \gamma_1 x_i & \text{for } x_i \in [\theta_2, \infty] \\ X_0^i - \gamma_0 x_i & \text{for } x_i \in [-\infty, \theta_1] \end{cases}. \quad (2)$$

$I_{syn}^{(i)}$ represents the synaptic coupling. θ_1 and θ_2 ($\theta_1 \leq \theta_2$) are the thresholds for the start and finish of a firing. γ_0 and γ_1 are the relaxation rates for the dynamics of inactive and active phases respectively. X_0^i and X_1 control respectively the periods of the inactive and active phases of the DIF oscillator. x_i is the membrane potential variable of the i th neuron of the DIF type where we assume that $X_1 < \gamma_1 \theta_2$ and $X_0^i > \gamma_0 \theta_1$ so that the DIF model represents an oscillatory system exhibiting spontaneous periodic firing. If x_i arrives at the threshold θ_1 , it jumps up to $x_i = X_u$ ($> \theta_2$) to start firing (*i.e.*, X_u is the onset value of the active phase, to which a trajectory jumps from the threshold θ_1). The trajectory then behaves as an exponentially decaying path over time until $x_i = \theta_2$. Then, it jumps down to the reset value X_r ($< \theta_1$) [Fig. 1].

We investigate the synchronous behavior of a pair of *non-identical* oscillators by setting,

$$X_0^1 = X_0, \quad X_0^2 = X_0 + \Delta, \quad (3)$$

where Δ is responsible for a small time period difference between the inactive phases of the DIF oscillators. It may be regarded as additive firing frequency noise, because neuron 2's frequency is slightly higher than neuron 1's one as $\Delta > 0$.

For the synaptic coupling $I_{syn}^{(i)}$, we use a specific version of the first order kinetics

$$I_{syn}^{(i)} = G_{syn} s_{\bar{i}} \quad (4)$$

where \bar{i} represents the counterpart of neuron i and G_{syn} shows the synaptic strength. It represents excitatory couplings when $G_{syn} > 0$, and inhibitory ones when $G_{syn} < 0$. We consider the synaptic conductance variable $s_{\bar{i}}$ ($1 \geq s_{\bar{i}} \geq 0$) to obey the following equation:

$$\frac{ds_{\bar{i}}}{dt} = \begin{cases} 0 & (x_{\bar{i}} \geq \theta_1) \\ -\beta s_{\bar{i}} & (x_{\bar{i}} < \theta_1) \end{cases} \quad (\bar{i} = 1, 2), \quad (5)$$

where $s_{\bar{i}} = 1$ whenever $x_{\bar{i}} \geq \theta_1$, in particular, $x_{\bar{i}}$ being in $[\theta_2, X_u]$. It exponentially decays with a rate constant β as $x_{\bar{i}}$ is in the inactive phase. We notice that $\gamma_0 \theta_1 - X_0^i < G_{syn} < \gamma_1 \theta_2 - X_1$ is necessary to see periodic firing behavior in neuron i . This is because if we set $\gamma_0 \theta_1 - X_0^i \geq G_{syn}$, one of the two neurons never arrive at θ_1 to start firing. For $\gamma_1 \theta_2 - X_1 \leq G_{syn}$ it cannot allow the neurons to stop firing.

In order to give a systematic explanation of how a pair of *non-identical* neurons behave, we need to analytically build up a return map corresponding to spike timings of the pair. The return map analysis is generally a beneficial method to study simple or complex synchronous behavior in coupled oscillators like Eqs.(1)-(5).

Constructing return maps for the solutions of Eqs.(1) to (5), we are concerned with the initial state of the two-neurons system that is just after neuron 1 has crossed the threshold θ_1 and fired. We then suppose that if neuron 2 is firing at the initial state, we set a phase difference between two neurons: $x^{(n)} = x_1(t_n^{(1)}) - x_2(t_n^{(1)}) = X_u - x_2^{(n)} (\geq 0)$, $s_1(t_n^{(1)}) = 1$ and $s_2(t_n^{(1)}) = 1$ where the times at which neuron 1 jumps to X_u are represented as $t_n^{(1)}$ ($n = 0, 1, 2, \dots$).

Specifying so-called temporal firing pattern diagrams (TFPDs), which represents how each firing dynamics switches between the active and inactive phases, we thus obtain mainly three different types of one-dimensional return maps that explicitly expresses $(n+1)$ th iterate as a function of n th iterate:

$$x^{(n+1)} = F^{(k)}(x^{(n)}), \quad (6)$$

where $k = \text{I}, \text{II}$ and III of the three different types of TFPDs. These return maps then determine if any stationary differences x_0 between two neurons as $t \rightarrow \infty$ (namely, $n \rightarrow \infty$). Subtracting gives the condition

$$F^{(k)}(x_0) - x_0 = 0. \quad (7)$$

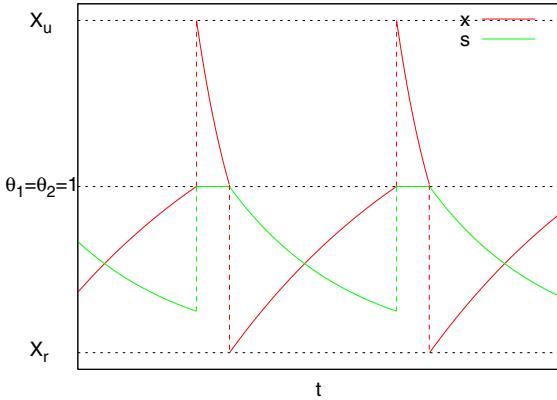


Fig. 1. A double integrate-and-fire model with synaptic coupling. The x -variable represents a membrane potential [red line]. The s -variable in I_{syn} (Eq.(4)) shows a synaptic response [light green line]. We set $X_0 = X_u = 2.0$, $X_1 = X_r = 0.0$, $\theta_1 = \theta_2 = 1.0$, $\gamma_0 = 1.0$ and $\gamma_1 = 5.0$.

For each k , the solution also expresses one of stationary synchronous states represented as the time lag rate between spikes of the two neurons. Letting $\phi = 0.0$ or 1.0 define exact inphase states, another synchronous states $0.0 < \phi < 1.0$ can be defined as the relevant 1:1 phase-locking states. The stabilities of such synchronous states are determined by

$$\frac{dF^{(k)}(x)}{dx} - 1 \gtrless 0. \quad (8)$$

$dF^{(k)}(x_0)/dx > 1$ means the fixed point x_0 is unstable (as drawn by black lines in Fig.2) while for $dF^{(k)}(x_0)/dx < 1$, the fixed point is stable and is represented by red lines in Fig.2

3 Simulation Results

Using a whole return map constructed in previous section, we simulate time evolutions of two coupled oscillators without current noise being attracted into stationary states. The stabilities of stationary states ϕ of the coupled oscillators are then shown by a linear stability analysis of the return map method[5][14]. We thus demonstrate how the detailed behavior of the 1:1 phase locking states of two coupled oscillators can be changed within the small time period difference Δ , $[0, 10^{-2}]$ (also called an additive firing frequency noise). It will be helpful to understand the additive firing frequency noise influence on synchronous behavior in the coupled oscillators without current noise. We set following parameters: $\gamma_0 = 1.0$, $\gamma_1 = 5.0$, $X_0 = 2.0$, $X_1 = 0.0$, $\theta_1 = \theta_2 = 1.0$, $X_u = 2.0$, $X_r = 0.0$ and $|G_{syn}| = 0.1$. Synchronous behavior in our system are numerically simulated with the Runge-Kutta method.

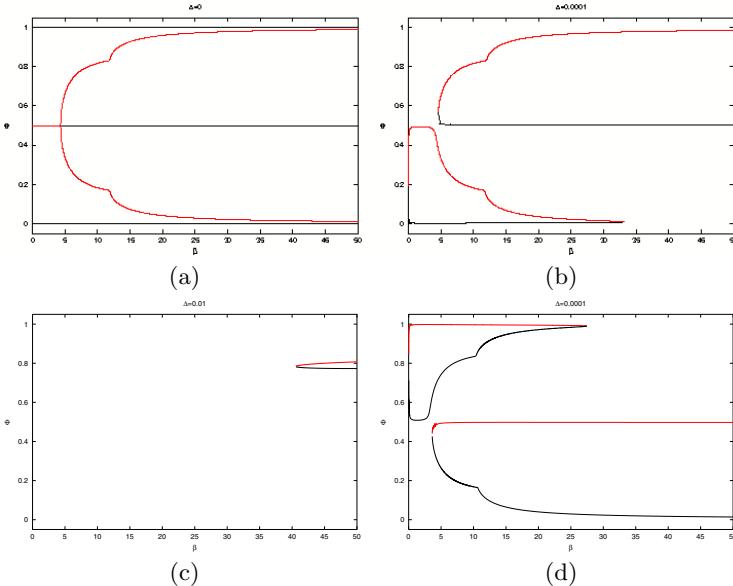


Fig. 2. ϕ - β bifurcation diagrams of (a) $\Delta = 0$, (b) $\Delta = 10^{-4}$ and (c) 10^{-2} for excitatory couplings and (d) $\Delta = 10^{-4}$ for inhibitory couplings. Stable fixed points are represented by red lines while unstable fixed points are shown by black lines.

3.1 Synchronization Shift by Different Firing Frequencies

We shall start with showing a bifurcation diagram for excitatory couplings when $\Delta = 0$. As shown in Fig.2(a), there are solutions $\phi = 0$, 0.5 and 1, but $\phi = 0.5$ is only stable when (or for) $\beta \lesssim 4.32$. The $\phi = 0.5$ becomes unstable and two additional stable solutions arise from it beyond the critical value of β (~ 4.32).

As shown in Figs.2(b), increasing Δ only by $\Delta = 10^{-4}$, the bifurcation diagram demonstrated in $\Delta = 0$ cannot be observed anymore. Interestingly enough, immediately after occurrence of one saddle-node bifurcation around $\phi = 0.25$ and $\beta \simeq 0$, the stable and unstable branches reach to $\phi \simeq 0.5$ and up to $\simeq 0.0$ as increasing β . A pitch-fork bifurcation at a critical value of β for $\Delta = 0.0$ has disappeared. We find another bifurcation point of the saddle-node type with stable and unstable out-of-phase solutions over $\phi = 0.5$. The stable branch generated at $\Delta = 0$ still remains in existence up to $\beta \gtrsim 33.0$ [Fig.2(b)]. For $\Delta \gtrsim 0.009$, the saddle-node bifurcation point around $\beta = 40.0$ is still remaining with a pair of the stable and unstable solutions [Fig.2(c)]. For the case of inhibitory couplings, the same series are obtained. The similar set of solutions exist but their stabilities are reversed as shown in Fig.2(d).

Regarding effects of the pulse width γ_1 on synchronous behavior, we increase the value of γ_1 with the case for $\Delta = 0$. As shown in an example of $\gamma_1 = 100$ [5], a pitchfork bifurcation point β shifts to a much smaller β to then disappear. Instead, synchronous behavior with almost or complete inphase appear in a whole β range. When $\Delta = 0.0005$ (though the result is not shown here), we have

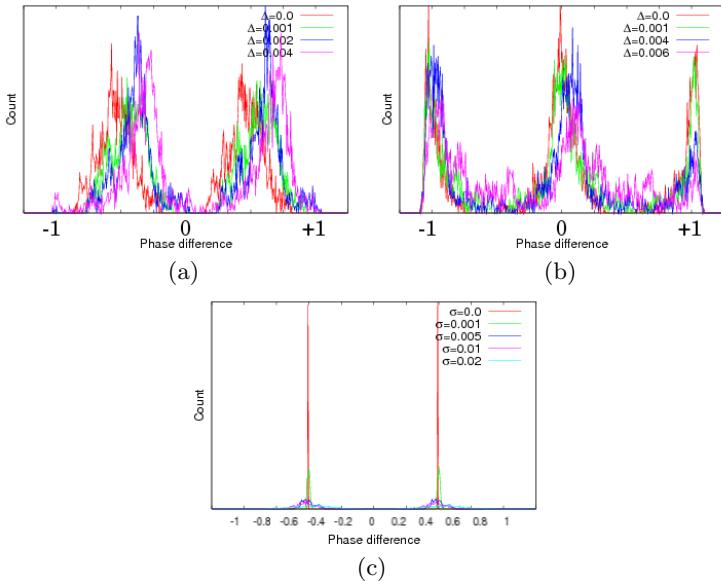


Fig. 3. Spike time difference distribution for $\beta = 2.0$ and $|G_{syn}| = 0.1$. (a) For $\sigma = 0.015$ and $G_{syn} = 0.1$, the change of the distribution by different values of Δ (b) For $\sigma = 0.015$ and $G_{syn} = -0.1$, the phase difference distribution makes asymmetric according to Δ . (c) For $\Delta = 0$ (namely, *identical* neural oscillators) and $G_{syn} = 0.1$, the heights of the distribution are varied according to a noise strength.

found two saddle-node bifurcations at $(\phi, \beta) \simeq (0.75, 0.0)$ and $\simeq (0.0, 0.0)$. The first bifurcation point for $\Delta = 0.0005$ is gradually moved toward larger β as Δ is further increased. Also, the additional bifurcation happens at much larger β when $\Delta = 0.01$. The second bifurcation for $\Delta = 0.0005$ have another bifurcation at $\beta \simeq 42$. It vanishes for larger Δ .

3.2 Effects of Noise: Distributions of a Phase Difference between Two Oscillators

We have so far found that the effect of frequency difference Δ gives rise to the mutual spike phase difference for 1 : 1 synchronization between the neurons without current noise (with $\sigma = 0$). We now investigate the case with current noise ($\sigma = 0.15$). The phase difference distribution, which may also be called the spike timing histogram, is obtained by detecting the firing times of neural oscillator 1 with respect to the firing times in neural oscillator 2 and then measuring the difference between the two firing times. Here the firing times are the times when neural oscillators 1 and 2 arrive at θ_1 .

Let us set system parameters $\beta = 2.0$ when $\gamma_1 = 5.0$, or a larger β for $\gamma_1 = 100.0$, involving current noise $\sigma = 0.015$. For the case with excitatory couplings, two peaks of phase difference distribution are staying around $\phi = \pm 0.5$ when $\Delta = 0.0$. As Δ slightly increases 0.0 to 0.004, the positions of the peaks

apparently shift bigger values than $\phi = \pm 0.5$ [Fig.3(a)]. This is due to that the $\phi = 0.5$ branch for $\Delta = 0.0$ in Fig.2(a) must have been transitted with the small Δ and at least no 1 : 1 phase locking exist in such a small β region when $\Delta = 0.004$.

When we consider the case with inhibitory couplings, we have shown that a phase difference distribution peaked at $\phi = 0$ also shift to the small bigger value of ϕ with Δ [Fig.3(b)]. Finally, we consider the case of $\Delta = 0$ to understand whether the current noise σ is very important for distribution shifts. We have found that even though σ is increased, the phase difference distribution keeps peaks at $\phi = 0.5$ without any effective shifts of the distribution[Fig.3(c)].

4 Discussion and Conclusion

In this paper, using a return map analysis, we have studied how two mutually interconnected neurons behave with their different firing time periods Δ . We have thus shown that complicated transitions of synchronization in the two coupled DIF oscillators appear, due to the preference of non-zero Δ .

We are aware that the coupled DIF oscillators can be reduced to the coupled IF types if γ_1 becomes large. When $\beta = 10.0$, $\Delta = 0.01$ and $\gamma_1 = 100.0$, we have obtained the 1:1 synchronization with an arbitrary phase difference by means of computer simulations. The 1:1 synchronization result is in agreement with a phase diagram proposed by Daido and Matsumoto [8]. So, the phase transition found by using DIF model is a more generalized result, compared to the Daido and Matsumoto's results, because of the preference of the width of an action potential.

Ermentrout and Saunders [4] suggested that the peak of the distribution for a phase difference should deviate due to noisy phase response curves (PRCs) of the neural oscillator model. The PRC can often be regarded as measuring one of firing properties in neural dynamics and play an effective role in a stability analysis by using an oscillator model being reduced to the phase equation [2][5]. However in general, the neural firing dynamics will intrinsically have various types of noise including current noise σ and firing frequency difference Δ presented here. They did not still investigate a dynamic effect of such noise on the histogram peak shifts.

In our analytical investigations of synchronization in the pair neurons without current noise, we have shown that synchronous behavior in the pair with slowly decaying synaptic responses and $\gamma_1 = 5$ is drastically changed as Δ is changed [Fig.2]. The synchronization transition observed with a small change of firing frequency difference, rather than by changing current noise, causes that phase difference distributions peaked at $\phi = 0$ and $= \pm 0.5$ shift to the small higher ϕ values. This qualitatively agrees with experimental results reported by Netoff et. al [12].

However, we will have to keep discussing more whether Δ essentially effects shifts of the distribution peaks. Remaining subjects are, for example, whether synaptic decaying rate constant β causes more substantially phase difference

distribution peak shifts, compared to qualitative results presented here, and also to check again random generators for creating current noise on the numerical simulation. These will be done elsewhere.

Acknowledgements

During the completion of this work, the authors would like to thank some helpful discussions of many individuals, in particular, Y. Tsuchiya, K. Kobayashi D. Takeshita and S. Bahar and Neuroscience Group at FIAS. This work has been generously funded by Hertie Foundation as well as another research programs. YDS was financially supported by the German Federal Ministry of Education and Research (BMBF) within the “Bernstein Focus: Neurotechnology” through research grant 01GQ0840. YT was engaging in the 21st Century COE Program at Tokyo Tech “Nanometer-Scale Quantum Physics” by the Ministry of Education, Culture, Sports, Science and Technology.

References

1. Fries, P., Roelfsema, P.R., Engel, A.K., König, P., Singer, W.: Proc. Natl. Acad. Sci. USA 94, 12699 (1997)
2. Ermentrout, G.B.: Neural Computation 8, 979 (1996)
3. Galan, R.F., Ermentrout, G.B., Urban, N.N.: Phys. Rev. Lett. 94, 158101 (2005)
4. Ermentrout, B., Saunders, D.: J. Comput. Neurosci. 20, 179 (2006)
5. Sato, Y.D., Shiino, M.: Phys. Rev. E 75, 011909 (2007)
6. Coombes, S., Lord, G.J.: Phys. Rev. E 55, R2104 (1997)
7. Bressloff, P.C., Coombes, S.: Phys. Rev. Lett. 80, 2168 (1998)
8. Daido, H., Matsumoto, S.: Europhys. Lett. 64(5), 613 (2003)
9. Elson, R.C., Selverston, A.I., Huetra, R., Rulkov, N.F., Rabinovich, M.I., Abarbanel, H.D.I.: Phys. Rev. Lett. 81, 5692 (1998)
10. Gibson, J.R., Beierlein, M., Connors, B.W.: J. Neurophysiol. 93, 467 (2005)
11. Long, M.A., Cruikshank, S.J., Jutras, M.J., Connors, B.W.: J. Neurosci. 25, 7309 (2005)
12. Netoff, T.I., Banks, M.I., Dorval, A.D., Acker, C.D., Haas, J.S., Kopell, N., White, J.A.: J. Neurophysiol. 93, 1197 (2005)
13. Merriam, E.B., Netoff, T.I., Banks, M.I.: J. Neurosci. 25, 6175 (2005)
14. Sato, Y.D.: Phys. Lett. A 319, 486 (2003)
15. Kuramoto, Y.: Chemical Oscillations, Waves, and Turbulence. Springer, Heidelberg (1984)

Parameter Analysis for Removing the Local Minima of Combinatorial Optimization Problems by Using the Inverse Function Delayed Neural Network

Yoshihiro Hayakawa and Koji Nakajima

Laboratory for Brainware Systems/Laboratory for Nanoelectronics and Spintronics,
Research Institute of Electrical Communication
Tohoku University, Sendai 980-8577, Japan
{asaka,hello}@nakajima.rie.c.tohoku.ac.jp

Abstract. The Inverse function Delayed (ID) model is a novel neuron model derived from a macroscopic model which is attached to conventional network action. The special characteristic of the ID model is to have the negative resistance effect. Such a negative resistance can actively destabilize undesirable states, and we expect that the ID model can avoid the local minimum problems for solving the combinatorial optimization problem. In computer simulations, we have shown that the ID network can avoid the local minimum problem with a particular combinatorial optimization problem, and we have also shown the existence of an appropriate parameter for finding an optimal solution with high success rate experimentally. In this paper, we theoretically estimate appropriate network parameters to remove all local minimum states.

1 Introduction

The most of existing neural network models interpret the biological behaviours in terms of time averaging techniques, rather than directly emulating them. It is a reasonable simplification to handle macroscopically the collective computational properties of the networks.[\[1\]](#)[\[2\]](#) On the other hand, the direct emulations are main concerns to the single or microscopic neuron models.[\[3\]](#)[\[4\]](#) Therefore, it is necessary that the novel model which connects the microscopic models and the macroscopic models. We have already proposed Inverse function Delayed Model(ID model)[\[5\]](#) to represent networks with neurons in the critical state, and we are studying about the fundamental characteristics of the ID model. The most important feature of the ID model is to include negative resistance dynamics.

The combinatorial optimization problems require to find the combination that has a minimum or maximum cost from several combinations, these problems can be found in actual life. Neural networks have been used for solving combinatorial optimization problems[\[6\]](#), and they compute in parallel and hence fast. In addition, it has high versatility. However, the existence of local minima is still a serious problem which interferes with searching global minima as the optimal solutions. In order to overcome this problem, various neural network models were proposed[\[7\]](#)[\[8\]](#)[\[9\]](#).

Our proposed ID model can destabilize undesirable stable states actively by using the negative resistance. Hence, we expect that the ID model is the powerful tool to find

an optimal solution of combinatorial optimization problem. In computer simulations, the ID network has found the optimal solutions with high probability[10] and we have shown that we can destabilizes only local minima in some kinds of problems typified by the N-Queen Problem by experiment[11][12]. In a previous paper[12], we have also pointed out the existence of particular permanent oscillations which interfered with reaching global minima, even if we destabilize all local minima. Additionally, we have also shown the preliminary result that these oscillations are removed by appropriate network parameters. However, these theoretical analyses are still left.

In this paper, we analyse the condition of removing all local minimum perfectly. First we define the combinatorial optimization problems which are treated by the ID model well, and then we clarify that we can obtain appropriate parameters to remove all local minima.

2 Basic Equation of the ID Neuron Model

The ID model is described as follows

$$\tau \frac{du_i}{dt} + u_i = \sum_j w_{ij} x_j + h_i, \quad (1)$$

$$\tau_x \frac{dx_i}{dt} = u_i - g(x_i), \quad (2)$$

where u_i , x_i and h_i are the internal state, the output and the bias of the neuron i , respectively, w_{ij} is the synaptic weight from neuron j to i , τ is the time constant of the internal state, τ_x is the time constant of the output, and $g(x) = f^{-1}$ is the inverse function of the output function $f(u)$. In addition, we assume $\tau_x \ll \tau$. Equation (2) is equal to $x = f(u)$ when $dx/dt = 0$, and the ID model are equivalent to the Hopfield model in the limit $\tau_x \simeq 0$.

Now, for simplicity let inputs of neuron i be $\theta_i = \sum_{j \neq i} w_{ij} x_j + h_i$ except for self-connection. We reduce variables of the equations to only x from both Eqs.(1) and (2) as follows,

$$\tau_x \frac{d^2 x_i}{dt^2} + \eta(x_i) \frac{dx_i}{dt} = -\frac{\partial U_i}{\partial x_i}, \quad (3)$$

$$\eta(x_i) = \frac{dg(x)}{dx}|_{x=x_i} + \frac{\tau_{x_i}}{\tau}, \quad (4)$$

$$\frac{\partial U_i}{\partial x} = (g(x_i) - w_{ii} x_i - \theta_i)/\tau. \quad (5)$$

Equation (3) represents a particle motion on the potential U_i which is obtained by the integration of Eq. (5). The first and the second terms on the left hand side of Eq.(3) correspond to a inertia and a friction terms, respectively. $\eta(x)$ is a friction factor and a nonlinear function of the output x .

$\eta(x)$ is always positive when $g(x)$ is a monotonic increasing function. However, it becomes negative if $g(x)$ has a N shape which gives a S shape to the output function f . Consequently, the ID model can have the negative resistance in its dynamics.

Next, let us consider the symmetric connections. If $w_{ij} = w_{ji}$, we can define the energy function similar to the Hopfield model as follows.

$$E = -\frac{1}{2} \sum_i w_{ij} x_i x_j - \sum_i h_i x_i + \sum_i \int_{\frac{1}{2}}^{x_i} g(x) dx + \frac{\tau_x \tau}{2} \sum_i \left(\frac{dx_i}{dt} \right)^2. \quad (6)$$

Its time derivative is derived by using Eqs. (1), (2) and (4),

$$\frac{dE}{dt} = -\tau \sum_i \eta(x_i) \left(\frac{dx_i}{dt} \right)^2. \quad (7)$$

Therefore, the time evolution of the system is a motion in state space, and it seeks out minima in E and comes to a stop at such points, if η is positive. However, in the negative η region, E turns to increase with time, and the motion climbs the potential hill. If we can cover over undesirable minima with the negative resistance region, they are destabilized.

In this paper, we use $g(x) = f^{-1}(x) - \alpha(x - 1/2)$ as an inverse function by using $f(u) = 1/(1 + \exp(-\beta u))$, because it is convenient to limit $x \in [0, 1]$ for the neural network to solve combinatorial optimization problems. Moreover, the linear term $-\alpha(x - 1/2)$ is used to introduce the negative resistance region. β and α correspond to a gain parameter of activation function and a control parameter of negative resistance, respectively. $g(x)$ is symmetric about $(x, u) = (0.5, 0)$, so that the negative resistance region is also symmetric. We define the distance L between the center of $x = 0.5$ and the edge of the negative resistance region,

$$L = \sqrt{1/4 - 1/(\beta(\alpha - \tau_x/\tau))}. \quad (8)$$

L is a function of α , β and τ_x/τ . Furthermore, we define ϵ to make easier as follows,

$$\epsilon = 1/2 - L. \quad (9)$$

3 Definition of the Combinatorial Optimization Problem

In this paper, we define the combination of three kinds of constraints in terms of the combinatorial optimization problems. A typical problem is the N-Queen Problems. This is expanded definitions from ones used by a CRANN[13]. We define "0-region" as $x \in [0, \epsilon]$ and "1-region" as $x \in [1 - \epsilon, 1]$, respectively. We suppose that the stable output in "0-region" is zero which is negligible from the empirical fact.

I. Number constraint

We consider a unit-group to make the number constraint. In such a group, only one unit must stay in "1-region" and all other units must be in "0-region". Additionally, all units in the network have to belong to one number constraint group at least.

II. Exclusive constraint

We consider a unit-group to make the exclusive constraint. In such a group, the number of units to stay in "1-region" are 0 or 1 and other units must be in "0-region". In sum, plural units cannot stay in "1-region".

III. Output constraint

Each unit must be in "0-region" or "1-region".

We define the notations as shown in Table I. The energy functions E_p , E_q and E_o are written by

$$E_p = 1/2 \sum_{l \in \{p\}} (1 - \sum_{i \in G_l^p} x_i)^2, \quad (10)$$

$$E_q = 1/2 \sum_{l \in \{q\}} \sum_{i \in G_l^q} \sum_{j \in G_l^q, j \neq i} x_i x_j, \quad (11)$$

$$E_o = 1/2 \sum_i x_i (1 - x_i), \quad (12)$$

Table 1. Definition of notations

$\{p\}$	a set of the number constraint groups	$\{q\}$	a set of the exclusive constraint groups
C_i^p	a subset of $\{p\}$ including unit i	C_i^q	a subset of $\{q\}$ including unit i
N_i^p	the number of element of a subset C_i^p	N_i^q	the number of element of a subset C_i^q
G_l^p	a set of the units in the number constraint group l	G_l^q	a set of the units in the exclusive constraint group l
n_l^p	the number of element of a set G_l^p	n_l^q	the number of element of a set G_l^q
N_{max}^p	the maximum number of N_i^p for all i	N_{max}^q	the maximum number of N_i^q for all i
n_{max}^p	the maximum number of n_l^p for all l	n_{max}^q	the maximum number of n_l^q for all l

respectively. Each energy should be 0, which is minimum value, to satisfy each constraint condition. Consequently, we obtain the total energy function for the combinatorial optimization problem as follows,

$$E_{problem} = AE_p + BE_q + CE_o, \quad (13)$$

where the coefficient A, B and C are non-negative value. An optimal solution satisfies the all constraints and it should correspond to $E_{problem} = 0$ as a global minimum.

According to the Hopfield model, we can also consider that the third term of Eq.(6) is negligible if β is large enough, and the fifth term is zero because the solutions are always stationary. From reference [12], we can correct both self connections and biases to erase the position dependency of minimum in the negative resistance region. The self-connection w_{ii}' and the input θ_i' after correcting are given as follows,

$$w_{ii}' = w_{ii} - \alpha = -A \sum_{l \in C_i^p} +C - \alpha, \quad (14)$$

$$\begin{aligned} \theta_i' = \theta_i + \frac{\alpha}{2} &= A \sum_{l \in C_i^p} (1 - \sum_{j \in G_l^p, j \neq i} x_j) \\ &- B \sum_{l \in C_i^q} \sum_{j \in G_l^q, j \neq i} x_j - \frac{C - \alpha}{2}. \end{aligned} \quad (15)$$

4 Requirement to Remove All Local Minimum States

We define ϵ_u as the critical ϵ for destabilization of all local minima. We can guarantee that all local minimum states are unstable if we set ϵ less than ϵ_u to satisfy three following inequalities.

$$A - \frac{C}{2} > f^{-1}(\epsilon_u) + (AN_{max}^p - C)\epsilon_u + A\epsilon_u, \quad (16)$$

$$B - \frac{C}{2} > f^{-1}(\epsilon_u) + (AN_{max}^p - C)\epsilon_u + B\epsilon_u, \quad (17)$$

$$A - BN_{max}^q - \frac{C}{2} > (AN_{max}^p - C)\epsilon_u + f^{-1}(\epsilon_u). \quad (18)$$

Now, let us consider the equilibrium points of the ID network. The equilibria have to satisfy $\frac{du_i}{dt} = \frac{dx_i}{dt} = 0$ in both Eqs. (11) and (12). Furthermore, we can derive a following equation in the case of corrected self-connections and biases.

$$f^{-1}(x_i) = \theta_i + w_{ii}x_i \quad (19)$$

Figure II shows the relation between θ_i and the equilibrium point. θ_i must be less than θ_i^- to stay the unit in 0-region. On the other hand, θ_i must be greater than θ_i^+ to stay the unit in 1-region. θ_i^- and θ_i^+ are given by the following equations,

$$\theta_i^- = f^{-1}(\epsilon) + (AN_i^p - C)\epsilon, \quad (20)$$

$$\theta_i^+ = (AN_i^p - C) - f^{-1}(\epsilon) - (AN_i^p - C)\epsilon. \quad (21)$$

Both of them are indispensable equations for the conditions destabilizing local minima.

Destabilization under supernumerary 1-region units in the number constraint group: Here we assume that the network state is stationary with supernumerary 1-region units in the number restraint group m , and let us consider the 1-region unit i in the group m . Hence, we can estimate θ_i from Eq.(15) as the following equation.

$$\theta_i = A \sum_{l \in C_i^p} -\frac{C}{2} - A \underbrace{\sum_{j \in G_m^p, j \neq i} x_j}_{(A1)} - A \underbrace{\sum_{l \in C_i^p, l \neq m} \sum_{j \in G_l^p, j \neq i} x_j}_{(A2)} - B \underbrace{\sum_{l \in C_i^q} \sum_{j \in G_m^q, j \neq i} x_j}_{(A3)} \quad (22)$$

The term shown by (A1) in Eq. (22) corresponds to the number constraint group m with the supernumerary 1-region units, so that the number of 1-region units except for itself should be one at least. Therefore, the minimum value of this term is $1 - \epsilon$. On the other hand, the terms indicated by both (A2) and (A3) are zero as minimum value. Consequently, we can derive the following equation,

$$\theta_i \leq AN_i^p - \frac{C}{2} - A(1 - \epsilon). \quad (23)$$

We can derive $\theta_i^+ > \theta_i$ from Eqs. (16), (19) and (23). Hence, our assumption is rejected. Under the condition of Eq. (16), the number of 1-region units in the number constraint group must be 0 or 1 for the stationary state.

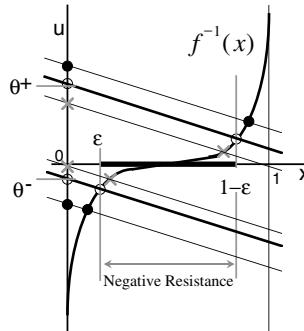


Fig. 1. The relation between θ_i and the equilibrium point

In the same way, we can derive $\theta_i^+ > \theta_i$ in the 1-region or $\theta_i^- < \theta_i$ in the 0-region from Eqs. (16), (17) and (18), easily. Consequently, under the condition that Eqs. (16), (17) and (18) are satisfied, the network can become a stationary state if and only if all constraint conditions are satisfied.

Additionally, we have pointed out that we have to erase oscillations to reach global minima with high probability [12] and that we can discuss the possibility to escape from these oscillations by comparing the staying times, while the competed units are unstable. Though we cannot express the derived process due to space limitation, the conditional equation is obtained as follows,

$$\frac{\tau_x}{\tau} + \frac{2}{4\beta\epsilon(1-\epsilon_p)} \geq \frac{(AN_{max}^p - C)C - 2(2B(1-\epsilon_p) - AN_{max}^p + C)f^{-1}(\epsilon_p)}{(2B - C) + 2(AN_{max}^p - 3B - C)\epsilon_p + 4\epsilon_p^2B - 2f^{-1}(\epsilon_p)}. \quad (24)$$

We also expect to be able to remove the oscillation states if $\epsilon < \epsilon_p$ is satisfied.

5 Computer Simulations

In this section, we confirm the validity of estimated parameters of ID model. We use the 4-coloring problem, which is well-known as "NP-complete". It is to find the solutions which assign a different color from adjacent areas with only 4 colors. Figures 2 and 3 show the maps and the solutions used in this section and Fig. 4 shows a part of the network structure with a focus on the region A in the case of sample 1. We have to arrange 4 units, which indicate color, to each region as shown in Fig. 4. The decision of color means that the number of 1-region units in the region should be one, and therefore, the number restraint conditions are applied to groups surrounded with the gray rectangle line. In this case, $N_{max}^p = 1$ because each unit belongs to only one group of the number restraint condition and $n_i^p = n_{max}^p = 4$.

Additionally, the units assigned to the same color of two adjacent regions can not stay in 1-region at the same time, hence, they construct a pair for the exclusive restraint as shown by grey linked lines in Fig 4. In case of Fig 2, $n_i^q = n_{max}^q = 2$ and $N_{max}^q = 6$ because the maximum adjacent regions are 6. Meanwhile, in case of Fig 3, $n_i^q = n_{max}^q = 2$ and $N_{max}^q = 9$.

Figures 5 and 6 show the dependences of final states on parameter ϵ and β . In both samples, we set $A = 2, B = 0.2$ and $C = 0.5$. The triangle, cross and circle marks show the parameter points investigated in computer simulations. The triangle marks indicate the existence of local minima, and both the cross marks and circle marks never have local minima. However, at the cross marks, the network does not always reach a global minimum and it has oscillation states. At the circle marks, all appeared final states are optimal solutions. The upper lines in both Figs. 5 and 6 show the lowest ϵ_u given by Eqs. (16), (17) and (18). We could confirm that all local minimum states are destabilized by the negative resistance if we set ϵ less than the lowest ϵ_u . Furthermore, the lower lines in both Figs. 5 and 6 show the ϵ_p given by Eq.(24). From these results, we could perfectly remove the oscillating states, and we achieved 100% success rate. Hence, ϵ should be less than $\min[\epsilon_p, \epsilon_u]$.

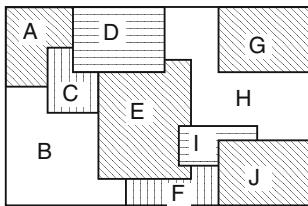
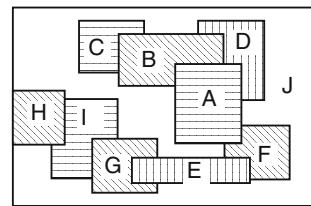
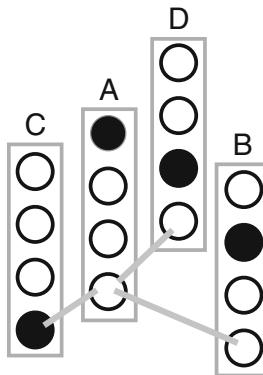
**Fig. 2.** Sample1 of 4-coloring problem**Fig.3.** Sample2 of 4-coloring problem

Fig.4. A part of the network structure for sample1 of 4-coloring problem. Each unit in the gray square is assigned to a specific color. The gray lines are examples of inhibition connections based on an exclusive constraint condition.

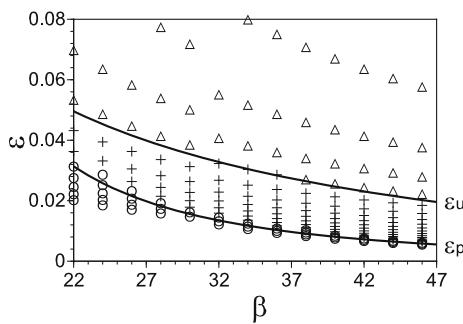


Fig.5. Dependence of final states on parameter ϵ and β in case of Sample 1. $A = 2$, $B = 0.2$ and $C = 0.5$.

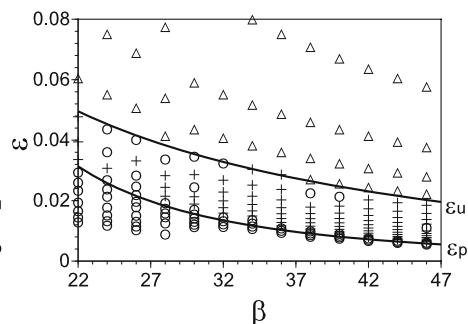


Fig.6. Dependence of final states on parameter ϵ and β in case of Sample 2. $A = 2$, $B = 0.2$ and $C = 0.5$.

6 Conclusion

Our proposed ID neural network has the negative resistance dynamics. The local minima, which are undesirable states, can be destabilized by this negative resistance actively. Hence, we believe that the ID neural network is very useful to find the optimal solutions of combinatorial optimization problems. In this paper, we expressed the problems by three constraints conditions to solve specific problems as typified by the N-Queen problem, then we estimated the relationship between the stabilization region and the positions of local and global minima analytically, and we clarified the safe network parameters to destabilize only local minima. Furthermore, the validity of estimated parameters was investigated by using the 4-coloring problems, and the results of computer simulation supported the validation of our estimations.

References

1. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophysics* 5, 115–133 (1943)
2. Caianiello, E.R.: Outline of a theory of thought processes and thinking machine. *J. Theor. Biol.* 1, 204–235 (1961)
3. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544 (1952)
4. FitzHugh, R.: Impulses and physiological states in theoretical models of nerve membrane. *Biophysical J.* 2, 445–466 (1961)
5. Nakajima, K., Hayakawa, Y.: Characteristics of Inverse Delayed Model for Neural Computation. In: Proceedings 2002 International Symposium on Nonlinear Theory and Its Applications, pp. 861–864 (2002)
6. Hopfield, J.J., Tank, D.W.: Neural computation of decisions in optimization problems. *Biological Cybernetics* 52, 141–152 (1985)
7. Chen, L., Aihara, K.: Chaotic Simulated Annealing by a Neural Network Model with Transient Chaos. *Neural Networks* 8(6), 915–930 (1995)
8. Hasegawa, M., Ikeguchi, T., Aihara, K., Itoh, K.: A novel chaotic search for quadratic assignment problems. *European Journal of Operational Research* 139, 543–556 (2002)
9. Nakaguchi, T., Jin'no, K., Tanaka, M.: Hysteresis Neural Networks for N-Queen Problems. *IEICE Trans. Fundamentals* E82-A(9), 1851–1859 (1999)
10. Hayakawa, Y., Denda, T., Nakajima, K.: Inverse function delayed model for optimization problems. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) *KES 2004. LNCS*, vol. 3213, pp. 981–987. Springer, Heidelberg (2004)
11. Sato, A., Hayakawa, Y., Nakajima, K.: The parameter dependence of the inverse function delayed model on the success rate of combinatorial optimization problems. *IEICE Trans. Fundamentals* (Japanese edn.) J89-A(11), 960–972 (2006)
12. Sato, A., Hayakawa, Y., Nakajima, K.: Avoidance of the Permanent Oscillating State in the Inverse Function Delayed Neural Network. *IEICE Trans. Fundamentals* E90-A(10), 2101–2107 (2007)
13. Nakajima, K., Hayakawa, Y.: Correct Reaction Neural Network. *Neural Networks* 6(2), 217–222 (1993)

Fractional-Order Hopfield Neural Networks

Arefeh Boroomand and Mohammad B. Menhaj

Electrical Engineering Department, Amirkabir University of Technology,
Tehran, Iran 15914
boroomand@aut.ac.ir, tmenhaj@ieee.org

Abstract. This paper proposes Fractional-order Hopfield Neural Networks (FHNN). This network is mainly based on the classic well-known Hopfield net in which fractance components with fractional order derivatives, replace capacitors. Stability of FHNN is fully investigated through energy-like function analysis. To show how effective the FHNN network is, an illustrative example for parameter estimation problem of the second-order system is finally considered in the paper. The results of simulation are very promising.

Keywords: Fractional-order, neural networks, parameter estimation.

1 Introduction

The study of the human brain is over thousands years old. With the advent of modern electronics, it was only natural to try to harness this thinking process. The first step toward artificial neural networks came in 1943 when Warren McCulloch, a neurophysiologist, and a young mathematician, Walter Pitts, wrote a paper on how neurons might work [1]. Recent work in neural networks includes Boltzmann machines, Hopfield nets, competitive learning models, multilayer networks, and adaptive resonance theory models.

In the beginning of the 1980s, J. Hopfield designed a new associative memory neural network, named Hopfield. He understood that some models of physical systems could be used to solve computational problems [2]. Such systems could be implemented in hardware by combining common and standard components such as capacitors and resistors. This feature is so significant especially in hardware implementation point of view.

About 300 years ago, calculus was first presented to the fractional world. In the fractional calculus, differential equations have non-integer order. The engineers could understand the importance of the fractional-order equations, especially when they observed that the description of some systems is more accurate, when the fractional derivative is used [3]. A typical example of a fractional order system is the voltage-current relation of a semi-infinite lossy transmission line (Wang, 1987).

In the fractional calculus, the main operator is the fractance. The fractance is a generalized capacitor. Actually it is an electrical circuit in which its voltage and current are related by the fractional-order differential equation [4].

From this idea, we use this generalized capacitor in the continuous Hopfield neural network instead of common capacitor and propose a new continuous network, called

as Fractional-order Hopfield Neural Network (FHNN), in which fractional-order equations can describe its behaviour.

The rest of the paper is organized as follows. In section 2, we review the fractional order systems and their implementation algorithms as well. In section 3, continuous Hopfield neural networks are reviewed. Section 4 fully derives fractional-order Hopfield neural networks and presents it's stability through energy-like function analysis. Section 5 shows capability of FHNN through an illustrative example. Finally, section 6 concludes the paper.

2 Fractional-Order Systems and Implementation Algorithms

We can compactly model, many of physical phenomena, material properties and processes, using uncommon differential equations in which they have fractional differential and integral parts. We name this class of equations, in brief, FDEs. In some researches [5-7], it is noted that many dynamical systems such as electrochemical processes, membranes of cells of biological organism, certain types of electrical noise, and chaos are more adequately described by FODE equations.

The main operator to state FDEs is a fractional differential operator. There are many ways to define this operator. A commonly used definition is the Caputo differintegral operator D_*^α as follows [8],

$$D^\alpha y(x) = J^{m-\alpha} y^{(m)}(x) . \quad (1)$$

where $m =: [\alpha]$ is the value α rounded up to the nearest integer, $y^{(m)}$ is the ordinary m th derivative of y and

$$J^\beta z(x) = \frac{1}{\Gamma(\beta)} \int_0^x (x-t)^{\beta-1} z(t) dt , \beta > 0 . \quad (2)$$

is the Riemann – Liouville integral of order β .

The Caputo operator has nice properties. For example it is seen in (3) that the Laplace domain representation of the Caputo derivative uses initial conditions $f^{(k)}(0)$ only for integer k . These initial values typically have a well understood physical meaning and can be measured directly.

$$L\{D^\alpha f(t)\} = S^\alpha F(S) - \sum_{k=0}^{m-1} f^{(k)}(0) s^{\alpha-k-1} . \quad (3)$$

The next stage toward FDEs is to present their solutions. There are two ways to tackle this problem. One way is to use an approximation method to represent fractional integrator operator $\frac{1}{s^\alpha}$ in the Laplace domain, by some pole-zero pairs as a transfer function. This approach uses both singularity functions in the frequency space and Bode diagram [9]. Although this method is good for a specific range of frequencies, as shown in [10], this approximation method may fail in detecting chaos phenomena in fractional-order systems.

Another way is to employ numerical techniques to find FDE solution [11-13]. These numerical methods are based on the approximation integral in (2), by different ways such as two-point trapezoidal quadrature formula. One of the common and reliable numerical methods is the Predictor-Corrector (PC) algorithm based on fractional Adams-Bashforth method. This algorithm is a generalization of the classical Adams-Bashforth-Moulton integrator that is well known for the numerical solution of first-order problems [12]. Knowing the fact that the PC algorithm has been constructed and analysed for the fully general set of equations $D^\alpha y(x) = f(x, y(x))$ without any special assumptions with ease of implementation on a computer, this paper employs it as fully discussed below.

3 Continuous Hopfield Neural Networks

The continuous Hopfield model of size N is a fully interconnected neural network with N continuous valued units. It contains linear and non-linear circuit elements, which typically are capacitor, resistor, op-amp, and sources. The topological structure of Hopfield net is shown in Fig.1. The input and output of the net are analog signals. The resistance R_{i0} and capacitor C_i are parallel to simulate the time-delay characteristics of biologic neurons. The resistance R_{ij} ($i, j = 1, 2, \dots, N$) and the op-amps are used to simulate the synapse and the non-linear characteristic of biologic neurons, respectively.

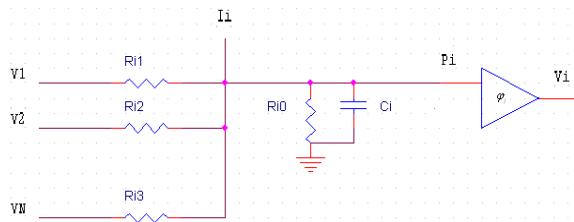


Fig. 1. The dynamic neuron of continuous hopfield

The state and output equations of continuous Hopfield with N neurons are given as follows:

$$C_i \frac{dP_i}{dt} = \sum_{j=1}^N W_{ij} V_j - \frac{P_i}{R_i} + I_i \quad , P_i = \left(\frac{1}{\lambda} \right) \varphi^{-1}_i (V_i) . \quad (4)$$

where $P_i(t)$ and $V_i(t)$ are the input and output of op-amp, respectively, for the i th neuron at time t . λ is the learning rate and W_{ij} is the conductance between the i th and j th neuron, and the following equations hold.

$$W_{ij} = \frac{1}{R_{ij}} , \quad \frac{1}{R_i} = \frac{1}{R_{i0}} + \sum_{j=1}^N W_{ij} . \quad (5)$$

Consider the following energy function as a Lyapunov function,

$$E = -\left(\frac{1}{2}\right) \sum_i \sum_j W_{ij} V_j V_i - \sum_i I_i V_i + \frac{1}{\lambda} \sum_i \left(\frac{1}{R_i}\right) \int_0^{V_i} \varphi^{-1}(v) dv . \quad (6)$$

Hopfield showed that, if the weights are symmetric $W_{ij} = W_{ji}$, then this energy function has a negative time gradient. This means that the evolution of dynamic system (4) in state space always seek the minima of the energy surface E.

Because of easy structure of Continuous Hopfield neural networks for implementation, they are often used for solving optimization problems.

4 Fractional-Order Hopfield Neural Networks

A new physical component, related to the kind of dielectric, which can be described by Caputo operator, is a fractance [4]. Actually a fractance is a generalized capacitor by fractional-order relationship between the voltage's terminal and the current passing through it. In 1994, Westerlund proposed a new linear capacitor model based on Curie's empirical law that for a general input voltage $u(t)$, the current is [14],

$$i(t) = F \frac{d^\alpha u(t)}{dt^\alpha} = F.D^\alpha u(t) . \quad (7)$$

From this idea, we use this generalized capacitor in the continuous Hopfield neural network instead of common capacitor and propose a new continuous network, named Fractional-order Hopfield Neural Network (FHNN), which can be described by fractional-order equations. The state and output equations of FHNN are as follows:

$$F_i.D^\alpha P_i = F_i P_i^{(\alpha)} = \sum_{j=1}^N W_{ij} V_j - \frac{P_i}{R_i} + I_i , \quad P_i = \left(\frac{1}{\lambda}\right) \varphi^{-1}(V_i) . \quad (8)$$

where $0 < \alpha < 1$ and $D^\alpha(\cdot)$ is defined in (1). From the FHNN structure in (8), the energy function for this network is as follows:

$$E = -\left(\frac{1}{2}\right) \sum_i \sum_j W_{ij} V_j V_i - \sum_i I_i V_i + \frac{1}{\lambda} \sum_i \left(\frac{1}{R_i}\right) \int_0^{V_i} \varphi^{-1}(v) dv . \quad (9)$$

To show the stability of proposed neural network, we need somehow to generalize the idea of Gradient operator to the fractional one.

Definition 1. The fractional gradient (or fractional gradient vector field) of a scalar function $f(x)$ with respect to a vector variable $x = (x_1, x_2, \dots, x_n)$ is denoted by $\nabla^{(\alpha)} f$, where $\nabla^{(\alpha)}$ shows the fractional gradient operator. The fractional gradient of f contains the vector field whose components are partial fractional differential of function f as follows:

$$\nabla^{(\alpha)} f = \left(\frac{\partial^\alpha f}{\partial x_1} \quad \frac{\partial^\alpha f}{\partial x_2} \quad \dots \quad \frac{\partial^\alpha f}{\partial x_n} \right) . \quad (10)$$

When a function depends also on a parameter such as time, the fractional gradient often refers simply to the vector of its spatial fractional derivatives only.

Theorem 1. For the FHNN described by (8) and (9), if $\varphi_i^{-1}(V_i)$ is a monotone increasing continuous function, and $F_i > 0$, then the FHNN energy function has a negative-definite fractional gradient and the following relationship exists as the net state changes.

$$\nabla^{(\alpha)} E \leq 0, \text{ and } \nabla^{(\alpha)} E = 0 \text{ iff } \nabla^{(\alpha)} V_i = 0 \quad i = (1, \dots, N) . \quad (11)$$

Proof: To calculate $\nabla^{(\alpha)} E$, we need some simplifications as follows, (from (1), because $0 < \alpha < 1$, then $m = 1$):

$$\varphi^{-1}(v) := h(v)$$

$$\begin{aligned} D^{(\alpha)} h(v) &= J^{1-\alpha} \left(\frac{\partial h}{\partial v} v'(t) \right) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} \cdot v'(\tau) \cdot \frac{\partial h}{\partial v} d\tau \\ &= \frac{\partial h}{\partial v} \cdot \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} \cdot v'(\tau) \cdot \frac{\partial h}{\partial v} d\tau . \end{aligned} \quad (12)$$

From definition (2), we have

$$D^{(\alpha)} v = J^{1-\alpha} (v'(t)) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} \cdot v'(\tau) d\tau . \quad (13)$$

From (12) and (13), we can extract the following relationship:

$$D^{(\alpha)} h(v) = D_t^{(\alpha)} v \cdot \frac{\partial}{\partial v} (h(v)) . \quad (14)$$

where $D_t^{(\alpha)}(.)$ shows the Caputo differintegral operator with respect to time. Furthermore, partial fractional derivatives of the first and second term in (9) are calculated as:

$$\begin{aligned} \frac{\partial^\alpha}{\partial t^\alpha} \left(-\frac{1}{2} \sum_i \sum_j W_{ij} V_j V_i \right) &= \frac{\partial^\alpha}{\partial t^\alpha} \left(-\frac{1}{2} V_- W V_- \right) = -\frac{1}{2} \nabla_v (V_- W V_-) \cdot D_t^{(\alpha)} V_- \\ &= -\sum_{i=1}^N D_t^{(\alpha)} V_i \sum_{j=1}^N W_{ij} V_j . \end{aligned} \quad (15)$$

$$\frac{\partial^\alpha}{\partial t^\alpha} \left(-\sum_{i=1}^N I_i V_i \right) = -\sum_{i=1}^N I_i D_t^{(\alpha)} V_i . \quad (16)$$

Now from (14) to (16), the fractional gradient of the energy function of FHNN will appear as:

$$\begin{aligned}
 \nabla_t^{(\alpha)} E &= \sum_{i=1}^N \frac{\partial E}{\partial V_i} \cdot \frac{\partial^{\alpha} V_i}{\partial t^{\alpha}} = -\sum_{i=1}^N D_t^{(\alpha)} V_i \sum_{j=1}^N W_{ij} V_j - \sum_{i=1}^N I_i D_t^{(\alpha)} V_i \\
 &\quad + \frac{1}{\lambda} \sum_{i=1}^N \left(\frac{1}{R_i} \right) \left(D_t^{(\alpha)} V_i \right) \left(\frac{\partial}{\partial v} (h(v)) \right) \\
 &= -\sum_{i=1}^N D_t^{(\alpha)} V_i \cdot \left(\sum_{j=1}^N W_{ij} V_j + I_i - \frac{P_i}{R_i} \right) = -\sum_{i=1}^N D_t^{(\alpha)} V_i \cdot F_i \cdot D^{\alpha} P_i \\
 &= -\sum_{i=1}^N (D_t^{(\alpha)} V_i)^2 F_i \cdot \frac{\partial}{\partial v} (\varphi^{-1}(v)) .
 \end{aligned} \tag{17}$$

From (17), if $\varphi_i^{-1}(V_i)$ is a monotonically increasing continuous function and $F_i > 0$, then equation (11) will be satisfied.

5 Illustrative Example

For software implementation of FHNN, the numerical PC algorithm is used assuming that the input impedances R_i is high enough so that the second term in (8) can be neglected and $F_i = 1$, ($i = 1, 2, \dots, N$). The PC algorithm is constructed to analyze the following equation.

$$D^{(\alpha)} P_i(t) = \sum_{j=1}^N W_{ij} \cdot \frac{1}{2} \left(1 + \tanh \left(\lambda \frac{P_i(t)}{P_0} \right) \right) + I_i . \tag{18}$$

To show application of the preceding network, we want to solve a parameter estimation problem based on optimization using FHNN network.

A second order system with two states and a single input is considered here,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0.3 \\ -0.2 & 0.9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -1 \\ 2 \end{bmatrix} u . \tag{19}$$

The objective function that would be optimized by the FHNN network is defined based on the parameter estimation error as follows:

$$E = \left(\frac{1}{k} \right) \int_0^k \left(\frac{1}{2} \right) (\dot{e}_q)^T (\dot{e}_q) dt = \left(\frac{1}{k} \right) \int_0^k \left(\frac{1}{2} \right) (\dot{x}(t) - \dot{y}(t))^T (\dot{x}(t) - \dot{y}(t)) dt . \tag{20}$$

where $x(t)$ and $y(t)$ are state variables of the system and estimator, respectively. For this problem, there are 6 parameters (four state matrix entries, and two input matrix

entries) to estimate, therefore the FHNN should have 6 neurons. By comparing (20) and (9), adjusting laws for W_{ij} and I_i will follow as [15],

$$\begin{aligned} [W_{ij}] &= -\left(\frac{1}{k}\right)^k \int_0^k \begin{bmatrix} XX^T & 0 & XU^T & 0 \\ 0 & XX^T & 0 & XU^T \\ UX^T & 0 & UU^T & 0 \\ 0 & UX^T & 0 & UU^T \end{bmatrix} dt \quad . \\ [I_i] &= \frac{1}{k} \int_0^k [X^T \dot{x}_1, X^T \dot{x}_2, u \dot{X}^T] dt \quad . \end{aligned}$$

To show the capability of the FHNN, the index $I_t = t^{-1} \int_{x=0}^t \|e(t)\|^2 d\tau$, can be used for the performance index of the parameter estimation error,

Fig.2 shows the simulation results of the parameter estimation by the FHNN network with fractional-order $\alpha = 0.85$, step size of the PC algorithm equal to 0.5 msec and learning rate $\lambda = 1$. Note that this net converges for any arbitrary λ and α , however the step size of the PC algorithm is indeed critical. Therefore, for a given α , it should be chosen by trial and error so that convergence of FHNN is guaranteed.

In Fig.3 the performance index evolving in time is depicted. The performance index will arrive at 0.0073 in 0.5 sec and the FHNN error becomes 2.12×10^{-12} . This value can completely show the convergence of the FHNN that has been proved before.

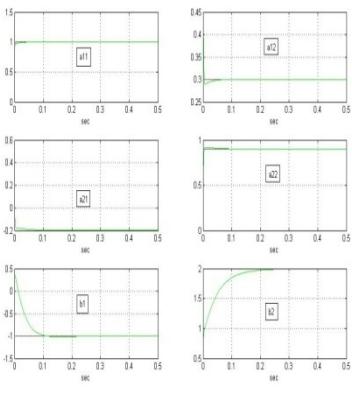


Fig. 2. Parameter estimation by FHNN

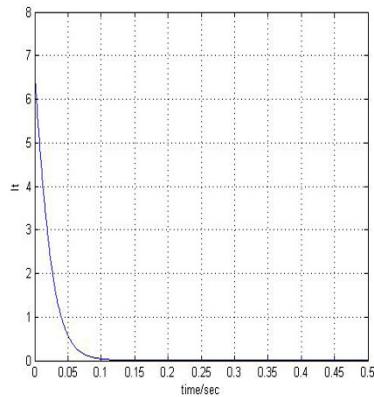


Fig. 3. Performance index of FHNN

6 Conclusion

In this paper, we have proposed fractional-order Hopfield neural networks. Stability of FHNN has been proved via energy-like function analysis. We showed the existence

of FHNN and its usage, by an illustrative example in parameter estimation problem. In the further work, it is good to extract significations of FHNN network comparing with the common Hopfield in various fields such as optimization problems.

As fractional order systems are systems that have infinite memory, probably common methods and algorithms by finite memory, may fail. In the future, it seems necessary to extend other methods and algorithms like our work to the fractional-order ones. Surely this approach will increase amount of computations using numerical algorithms for solving FDEs.

References

1. Hayman, S.: The McCulloch-Pitts model. In: International Joint Conference on Neural Networks, IJCNN apos 1999, July 1999, vol. 6, pp. 443–4439 (1999)
2. Hopfield, J.J.: Neural Network and Physical Systems with Emergent Collective Computational Abilities. Proc. Natl. Acad. Sci. 79, 2554–2558 (1982)
3. Jenson, V.G., Jeffreys, G.V.: Mathematical methods in chemical engineering, 2nd edn. Academic Press, New York (1977)
4. Nakagava, M., Sorimachi, K.: Basic characteristic of a fractance device. IEICE Trans. fundamentals E75 -A(12), 1814–1818 (1992)
5. Li, C., Chen, G.: Chaos in fractional order Chen system and its control. In: Chaos, Solutions & Fractals, pp. 305–311 (2004)
6. Cole, K.S.: Electric conductance of biological system. In: Proc. Cold Spring Harbor Symp. Quant. Biol., New York, pp. 107–116 (1993)
7. Mandelbrot, B.B.: Some noises with $1/f$ spectrum, a bridge between direct current and white noise. IEEE Trans. Info. Theory IT-13(2) (1967)
8. Caputo, M.: Linear models of dissipation whose Q is almost frequency independent. The Geophysical Journal of the Royal Astronomical Society 13, 529–539 (1967)
9. Charef, A., Sun, H.M., Tsao, Y.Y., Onaral, B.: Fractional systems as represented by singularity function. IEEE Transact. On Automatic control 37(9), 1465–1470 (1992)
10. Tavazoei, M.S., Haeri, M.: Chaotic Attractors in Incommensurate Fractional Order systems. Physica D (2008)
11. Lynch, V.E., Carreras, B.A., del-Castillo-Negrete, D., Ferreira-Mejias, K.M., Hicks, H.R.: Numerical methods for the solution of partial differential equations of fractional order. Journal of Computational Physics 192, 406–421 (2003)
12. Diethelm, K., Neville, J.F., Alan, D.F.: A Predictor-Corrector Approach for the Numerical Solution of Fractional Differential Equations. Nonlinear Dynamics 29, 3–22 (2002)
13. Kumar, P., Agrawal, O.P.: An approximate method for numerical solution of fractional differential equations. Signal Processing 86, 2602–2610 (2006)
14. Westerland, S.: Capacitor theory. IEEE Transaction on Dielectrics and Electrical Insulation 1(5), 826–839 (1994)
15. Reynold Chu, S., Shoureshi, R., Tenorio, M.: Neural networks for system identification. IEEE Control System Magazine (1990)

Part X

Special Session: Applications of Intelligent Methods in Ecological Informatics

Classification and Prediction of Lower Troposphere Layers Influence on RF Propagation Using Artificial Neural Networks

Martin Mudroch¹, Pavel Pecháč¹, Martin Grábner², and Václav Kvičera²

¹ Czech Technical University in Prague, Faculty of Electrical Engineering,
Department of Electromagnetic Field

Technická 2, 16627 Prague, Czech Republic

{mudrom1, pechac}@fel.cvut.cz

² Czech Metrology Institute, Hvozdanska 3,
148 01 Praha 8, Czech Republic

{mgrabner, vkvicera}@cmi.cz

Abstract. This paper describes the basic steps of a novel approach to weather classification by remote and local atmosphere sensing. Atmospheric data on the troposphere are gathered by 10 GHz radio links and a number of meteorological sensors. Classification is performed by artificial neural networks (ANN) and is crucial for further processing, because of the different RF propagation influences under a variety of weather conditions. Reasons for using ANN compared to other means of classification are discussed. Differences in the size and number of hidden layers of back-propagation networks used are discussed. Different learning sets of measured data and their construction are also evaluated.

1 Introduction

The Earth's atmosphere is an inhomogeneous layered medium, whose physical properties are continually changing in time and place. The troposphere, as the lowest atmospheric layer, is one of the most important environments from the point of view of short terrestrial radio link planning and management. It is strongly influenced by winds, rains and also by air pollution. The atmosphere acts as a dielectric environment described by means of its electrical properties and the permittivity and refraction index derived thereof. It follows from the classic laws of Snell that the electromagnetic wave trajectory bends when crossing an environment with an inhomogeneous distribution of the refraction index. In the atmosphere, areas having a considerably inhomogeneous distribution of the refraction index spring up spontaneously as described in [1]. These changes in the atmosphere causes electromagnetic wave bends (markedly affect waves propagating horizontally); therefore radio signal outages may come up in extreme cases.

The knowledge of the refraction index and its changes in place and time is important for a deeper understanding of electromagnetic wave propagation. The refraction index is investigated using a wide range of methods. Most experiments have been

conducted in coastal areas and were mainly directed at the waveguide layer effects on electromagnetic wave propagation in the troposphere, with the evaluation performed by statistical means (e.g. [2], [3], [4]).

Our experiment (with more details published in [5]) focuses on combined local (VAISALA meteoboxes situated in different heights on the receiving side) and remote atmosphere sensing using an amplitude modulated radio signal at a frequency of 10.671 GHz. There is one point-to-multipoint link – the transmitter is situated at Prague-Zizkov TV Tower and four receivers are placed at different heights above ground-level on the antenna mast in Podebrady (the link length is 49.82 km). The benefit of this unique experiment is the existence of different wave propagation angles in the atmosphere. Each link interferes differently with troposphere layers – wave to higher situated receivers travel much more horizontally than wave to lower ones. Link attenuation is the measured value on the receiving side of each link. The terrain profile of the link can be seen in Fig. 1.

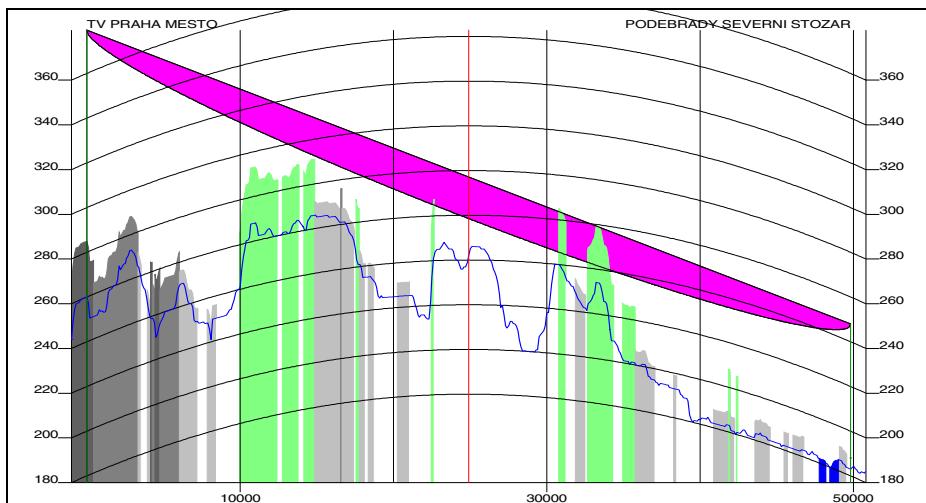


Fig. 1. Terrain profile of the experimental link

2 Data Processing Objectives

This project aims to find a suitable method for weather classification as well as various other physical properties of the atmosphere which are crucial for RF propagation. The initial aim of this novel experiment is an attempt to classify the amount of precipitation in the atmosphere. It is obvious that this can be achieved by other mathematical models but we would like to create a universal method which can then be used to classify other atmospheric properties with much more complex dependencies.

All the data used in this project were gathered between November 2007 and April 2008. Some newer data are also available, but we are unable to guarantee its consistency due to technical problems that affected the meteorological and attenuation measurements during the spring months. The following types of data were gathered:

attenuation of each of the 4 links, temperature pseudo-gradient on the Podebrady antenna mast (heights ranging from ground level to 123.86 meters above ground level), atmospheric pressure, humidity, visibility, wind speed and direction and general weather classification. All of these data were collected in short time steps (from seconds to 15 minutes in slow changing variables) so we can consider them to be continuous in time.

Our primary task was to find a method of determining basic classes of weather using a minimal amount of external data from meteorological sensors – ideally using only attenuation data. This automatic classification will be used during further evaluation of the atmospheric refraction index.

As mentioned in [5], there is no suitable way of evaluating our objective using standard statistical methods due to the inhomogeneity of the data gathered. Errors involved in the mathematical statistics produced totally imprecise and unusable models. The fact that the data gathered contains errors caused by link length has to be taken into account. For example a local shower may occur during the middle part of the link that cannot be detected by the meteorological sensors we installed. Therefore a method which is resistant to these errors has to be found – one of these being artificial neural networks or fuzzy logic. We chose back-propagation ANNs which are often used as universal classifiers.

3 Learning Set Construction

We created a dataset from all the available attenuation data – which hereinafter will be referred to it as the ALL dataset. Our classification was mainly based on the VAISALA meteobox classification of weather, covering humidity, visibility and rainfall sensor data. This can be changed in future if any sensor is available in between the receiver and the transmitter. Data were divided into CLEAR, LIGHT, NORMAL and HEAVY weather classes (the names are self-descriptive in terms of the amount of precipitation). The size of each class is given in Table 1. This table can also help to provide a picture of the weather conditions during the period under consideration. These classes are used to construct learning sets with deterministic distribution and, particularly, to validate trained ANNs.

The main problem we encountered in constructing the data sets was the small size of the heavy and normal rain classes during the months of observation, caused by an arid winter and spring – this can only be resolved by making long term measurements over an entire year.

Table 1. Weather class and learning set sizes

ALL	201600
CLEAR	184385
LIGHT	16735
NORMAL	447
HEAVY	33
DAY	5000
MIX	120

To make learning datasets of a suitable size we had to use almost the whole HEAVY class for the MIX dataset. MIX consists of uniformly distributed random representatives – 30 from each weather class. The DAY dataset was experimentally chosen from a time series of all the data. It should have the same distribution as the ALL dataset. The size of 5000 elements was experimentally chosen in order to enable the training of small-sized neural networks as well as large-sized ANNs.

4 Neural Network Architectures and Learning Process

We are trying to create a universal classifier of non-uniformly distributed errors - including the measured data. We therefore selected simple feed forward ANNs with 2 or 3 hidden layers. We did not know what size of network was suitable for nor the number of hidden layers beforehand, neither we were aware of the best size and distribution of the learning set. All simulations were performed using the Stuttgart Neural Network Simulator and its Java GUI and the final statistical evaluation was performed using Matlab software.

All the networks consisted of 4 input units, hidden layers and one output unit. The input units only acted as branching elements for the hidden layers. Hidden layer units had a standard logistic activation function with an identity on output which was selected because of the continuous nature of the attenuation data. The output unit also used a logistic activation function, but it would also be possible to use a multiple-step function, which would make a final sorting of the data by rounding the output value to its nearest representative. This approach can lead to a lowering of the Mean Square Error (MSE), which is used as the principal indicator of training quality (together with a count of the classification errors in the dataset as a whole). On the other hand, when the neural network is not trained correctly, the MSE will rise – which is also a suitable indicator for the learning process. It was experimentally verified that the MSE for correctly trained networks with this adaptation may be decreased to 85 percent of the original value.

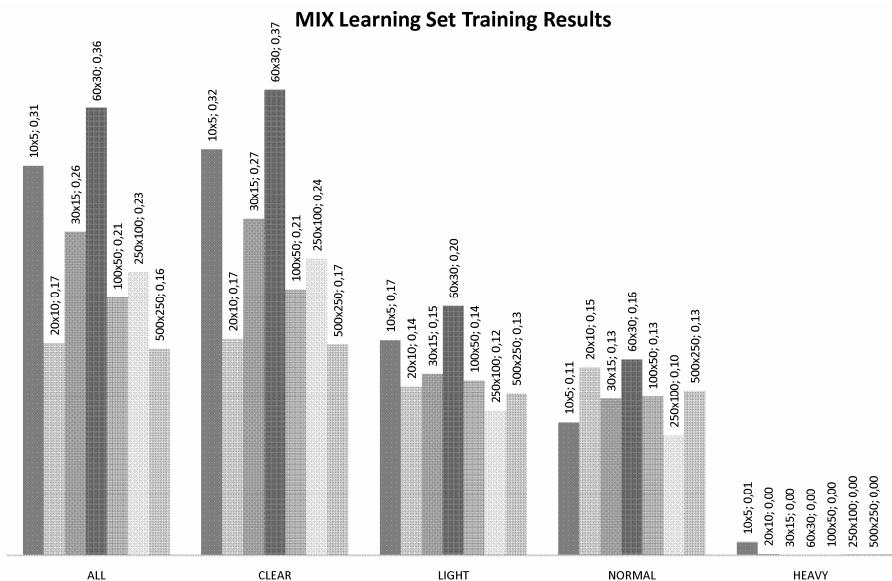
Training of the neural network was usually performed in subsequent cycles using a standard back-propagation algorithm with manual learning parameter changes. The best results achieved were recorded for further processing. Results were also checked for correctness of training – in some cases neural networks were trained to return only constant values. In these case ANNs were re-initialized and given different learning parameters.

5 Neural Network Training Evaluation

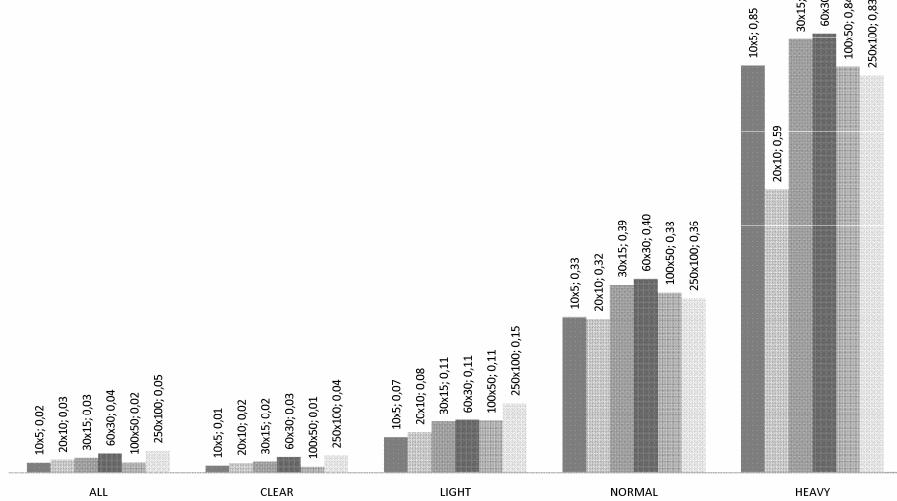
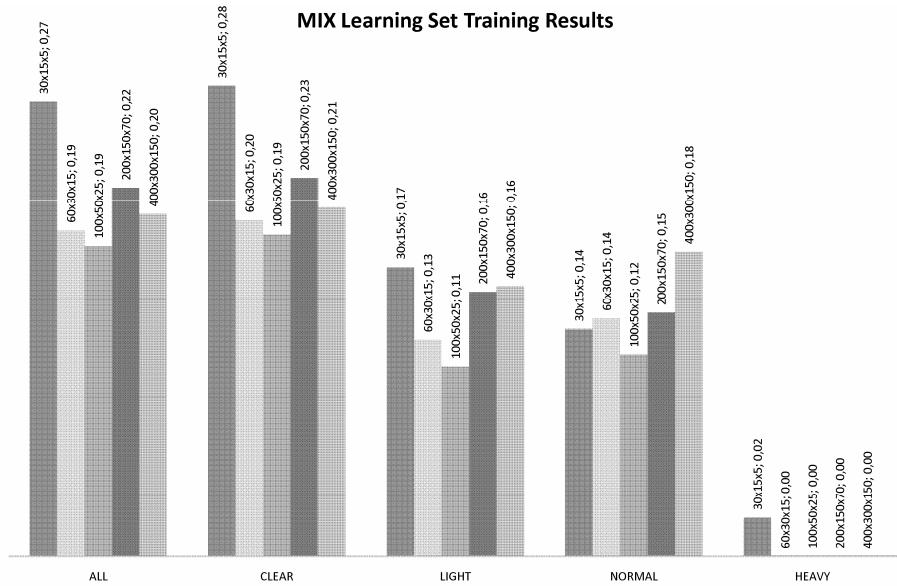
The best results achieved during the training cycles are introduced in Table 2. It is obvious that two-layered neural networks ranging in size from 20x10 (20 in first hidden layer and 10 in second hidden layer) and all small three-layer feed-forward ANNs are suitable for training the MIX dataset. In the three-layer networks we found the maximum size of network which we are able to train in reasonable amount of time – 200x150x70. On the other hand for the DAY training set the most suitable two-layered ANNs are 60x30 and more in size or 60x30x15 for the three-layered designs. There was also a presumption that 5 to 15 units in last hidden layer exerted a good influence on the quality of the network response. This will have to be examined in more detail during future research.

Table 2. The best MSE achieved for each training set and neural network size

Network	Connections	Learn MIX		Learn DAY	
		MIX	DAY	MIX	DAY
10x5	95	0.026	0.123	0.317	0.005
20x10	290	0.001	0.144	0.254	0.003
30x15	585	0.001	0.215	0.350	0.002
60x30	2070	0.000	0.179	0.362	0.001
100x50	5450	0.000	0.136	0.332	0.000
250x100	26100	0.000	0.122	0.332	0.000
500x250	127250	0.000	0.123	Unable to train	
30x15x5	650	0.001	0.217	0.289	0.002
60x30x15	2505	0.000	0.174	0.302	0.001
100x50x25	6675	0.000	0.134	0.331	0.000
200x150x70	41370	0.000	0.161	0.331	0.000
400x300x150	166750	0.012	0.224	Unable to train	

**Fig. 2.** MSE evaluation for all investigated two-layer networks using MIX learning set

The results achieved during training of two-layered networks are introduced in Fig. 2 and Fig. 3 for two-layered and Fig. 4 and Fig. 5 for three-layered networks. These graphs enable us to select the most suitable networks for our classification. From our point of view the most suitable are networks with an architecture of 20x10 and 100x50 for two-layered and 100x50x25 for three-layered networks.

DAY Learning Set Training Results**Fig. 3.** MSE evaluation for all investigated two-layer networks using DAY learning set**MIX Learning Set Training Results****Fig. 4.** MSE evaluation for all investigated three-layer networks using MIX learning set

As can be observed from the graphs, the DAY learning set involves an unusable major error in the HEAVY class as well as in other classes where rain was involved. Therefore we have to declare the MIX dataset more suitable for learning the proposed

neural networks. In comparison with the DAY dataset most errors can be observed in the CLEAR class. That is why we will abandon the idea of the uniform distribution of class members and will place more stress on the CLEAR and LIGHT classes by adding more members from these classes – preferably from the overall period of observation because of monthly variations.

DAY Learning Set Training Results

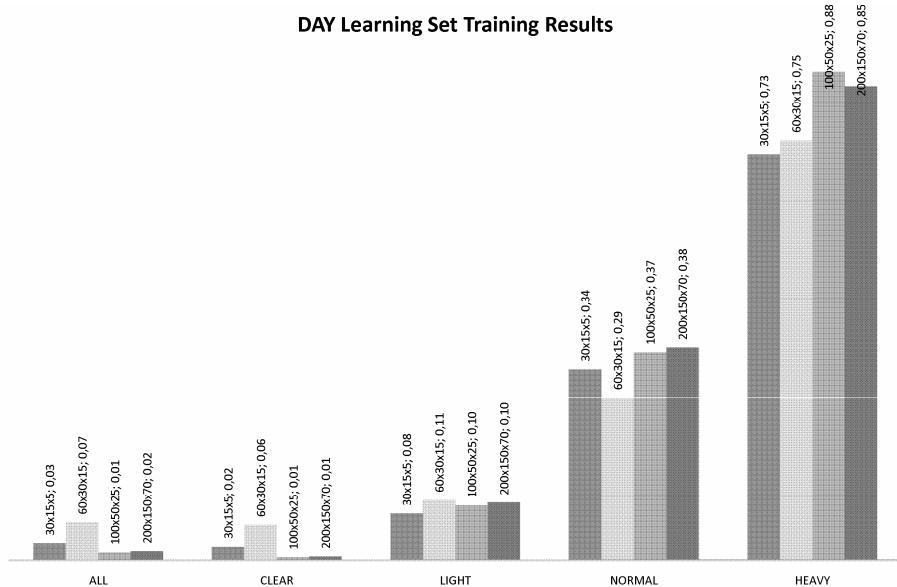


Fig. 5. MSE evaluation for all investigated three-layer networks using DAY learning set

6 Conclusions

In this paper we have introduced the initial ANN designs for weather classification using our experimental wireless links. Some of those networks are suitable and can be used for further evaluation of refraction index models – at least for selecting correct data, which can be used for the existing refraction index models valid for a clear weather conditions. The correct sizes of two-layered and three-layered neural networks were found as a solution to our problem. The learning set distribution with the best training result was selected. Important steps that can lead to more precise classification as well as significant changes in the MIX learning set to improve the quality of neural network response were also presented.

Acknowledgements. This work was supported by the Czech Science Foundation under Grant 102/07/0955 "Research on Electromagnetic Wave Refraction and Refractivity Height Profiles in Troposphere".

References

1. Grabner, M., Kvicera, V.: Statistics of Lower Atmosphere Refractivity in Czechia. In: Proc. of Asia Pacific Microwave Conference 2005, Suzhou, China (2005)
2. Allen, K.C., Ott, R.H., Violette, E.J., Espeland, R.H.: Height-Gain Study for 23 km Links at 9.6, 11.4, and 28.8 GHz. *IEEE Trans. Antennas Propag.* 30, 734–740 (1982)
3. Lam, W.I., Webster, A.R.: Microwave Propagation on Two Line-of-Sight Oversea Paths. *IEEE Trans. Antennas Propagat.* 33, 510–516 (1985)
4. Webster, A.R.: Multipath Angle-of-Arrival Measurements on Microwave Line-of-Sight Links. *IEEE Trans. Antennas Propagat.* 39, 798–803 (1991)
5. Mudroch, M., Grabner, P., Kvicera, V., Hudec, P., Bedrna, P., et al.: An Experimental Study of the Lowest Troposphere Layers at 10 GHz - First Results. In: Proceedings of the 14th Conference on Microwave Techniques COMITE 2008, Praha, pp. 73–76 (2008)

Predicting the Distribution of Fungal Crop Diseases from Abiotic and Biotic Factors Using Multi-Layer Perceptrons

Michael J. Watts¹ and Sue P. Worner²

¹ School of Biological Sciences, University of Sydney, NSW 2006, Australia

² Bio-Protection Research Centre, P.O. Box 86, Lincoln University,
Lincoln 7647, New Zealand

Abstract. Predictions of the distribution of fungal crop diseases have previously been made solely from climatic data. To our knowledge there has been no study that has used biotic variables, either alone or in combination with climate factors, to make broad scale predictions of the presence or absence of fungal species in particular regions. The work presented in this paper used multi-layer perceptrons (MLP) to predict the presence and absence of several species of fungal crop diseases across world-wide geographical regions. These predictions were made using three sets of variables: abiotic climate variables; biotic variables, represented by host plant assemblages; And finally the combination of predictions of the climate and host assemblage MLP using a cascaded MLP architecture, such that final predictions were made from both abiotic and biotic factors.

1 Introduction

The rising rate of global tourism and trade is rapidly increasing the threat to agricultural and horticultural production of many countries by unintended introductions of exotic crop diseases, especially fungal diseases. Additionally, indigenous ecosystems may be threatened by some species. There is therefore a desperate need to develop methods that have a high level of prediction to assist the risk assessment process.

A large amount of data exists that describes many features of the climate in numerous geographical locations, as well as the presence or absence of a number of species of fungal crop disease. Better prediction of their potential distribution is required so that any threat a species poses to agriculture or biodiversity of various regions can be more accurately assessed.

The factors affecting species establishment within a geographical region of a particular fungal species can be divided into two general groups: firstly, biotic factors, that include the presence of potential host species, as well as competing species; and abiotic factors, that essentially comprises the climate of the region in question. A number of models and approaches have been designed to predict the establishment of fungal crop diseases in regions where they are not normally found [3][4].

Artificial neural networks (ANN) have previously been used for many applications in ecology [1][8], including modelling the relationship between cities and the levels of contaminants in grasses [5] and the presence of certain species of freshwater fish [10]. They have also been applied to the prediction of fungal diseases, such as *Pyrenophora tritici-repentis* [34]. However, this work, and work done with other techniques such as discriminant analysis [15], regression analysis [14] and tools such as CLIMEX [19] made predictions entirely from abiotic [16] factors. To our knowledge there has been no study that has used both biotic and abiotic factors to predict potential for establishment of fungal crop diseases in areas where they are not normally found.

The goal of the research reported here was to investigate the use of ANN, specifically MLP, to model the distribution of several fungal crop diseases by predicting their presence and absence in world-wide geographical regions. These predictions were made in three ways: Firstly, using climate (abiotic) factors; Secondly, from biotic factors, represented by regional host plant assemblages; Thirdly, by using a cascaded MLP architecture that combined predictions made from both climate and host plant assemblages.

2 Method

The data set used in this study consisted of data describing the climate for each of 459 geographic regions and the presence or absence in each of those regions of 130 crop disease-causing fungal species. The species presence data were sourced from the CABI Crop Protection Compendium [2]. The climate data were compiled from Internet sites maintained by recognised meteorological organisations. None of this data contained any explicit information about the links between climate in a region and the presence or absence of any particular disease.

Forty five climate variables were available. Data describing the climate on a monthly basis for each region were converted to seasonal data to deal with the opposite of seasonal pattern between the Northern and Southern hemispheres. To represent the range of each variable within a region the minimum, mean and maximum of each variable was calculated. This resulted in a total of one hundred and thirty five input variables describing the climate in each region. Each variable was linearly normalised between zero and unity.

The second set of data comprised the presence and absence of 114 host plants in the same geographic regions as above. Only those hosts that were recorded as being present in more than 5% of the regions were retained in the data set.

To verify that the host plant species assemblages were non-random, the assemblages were subjected to a null model analysis [9]. The software used for this analysis was EcoSim version 7 [7]. The results of each run showed that the host plant assemblage was significantly non-random ($p = 0.001$).

Six target fungi species were selected, all of which were ranked as significant threats to New Zealand according to the method described in [21]. Although this technique was originally developed to estimate the risk of invasion of insect pest species, there is no reason why it should not be applicable to other threatening organisms. The six species identified are presented in Table 1.

Table 1. Target species. “Prevalence” lists the number of geographic regions in which each species is recorded as being present.

Scientific Name	Common Name	Prevalence
<i>Tilletia tritici</i>	complete bunt	126
<i>Ascochyta pisi</i>	blight: pea	92
<i>Ustilago hordei</i>	covered smut of barley	156
<i>Plasmopara halstedii</i>	downy mildew: sunflower	74
<i>Pyrenophora tritici-repentis</i>	leaf spot of wheat	96
<i>Leveillula taurica</i>	Powdery mildew	130

Table 2. Training epochs per species per data set

Species	Climate Hosts	
<i>T. tritici</i>	500	500
<i>A. pisi</i>	750	750
<i>U. hordei</i>	500	500
<i>P. halstedii</i>	750	750
<i>P. tritici-repentis</i>	750	750
<i>L. taurica</i>	500	500

The data were randomly split into two major sets. The first, containing 80% of the data, was the training and test set, from which samples were randomly drawn to form the two sets for each trial. The remaining 20% comprised the validation set, which was used to perform an independent evaluation of the prediction accuracy for each target species.

Each data division was done on the basis of regions. In other words, the 459 regions were randomly assigned into two groups in an 80%/20% ratio, and the corresponding fungi and host assemblages found such that the validation set contained data corresponding to the same regions for both the fungi and host assemblages. The same division of regions was used for the analysis of each species.

Standard three neuron-layer multi-layer perceptrons (MLP) were used in these experiments, and the learning algorithm used was unmodified back-propagation with momentum [18]. The parameters of the MLP and learning algorithm were three hidden neurons, a learning rate and momentum of 0.03 and either 500 or 750 training epochs. Table 2 lists the epochs selected for each MLP used to model each species, for each data set. These parameters were found via experimentation to yield the best balance of training and generalisation errors: more than three hidden neurons consistently caused over-training, while fewer meant the MLP were unable to learn. Similar parameters were previously found to be effective for modelling the establishment of insect pest species in another study [20]. While the target data in this work was different (presence and absence of fungal diseases, rather than insect pests) the basic problem was the same: predicting the presence or absence of a species from the same set of regional climate variables.

The method of training and evaluating the MLP (and also selecting the parameters) was similar to that suggested in [6] and [17]. A total of one thousand trials were performed over each species. For each trial, the training and test data set (consisting of 80% of the total data available) was randomly divided into a training set and a testing set in a two-thirds/one-third ratio. A MLP was then created with randomly initialised connection weights and trained over the training division. The accuracy of the MLP over the training division was then evaluated to determine how well the network had learned, and again over the testing division to determine how well the network generalised. Accuracy was measured using Cohen's Kappa statistic [1]. Whereas percentage accuracy is easily interpreted, it is also easily biased by unbalanced numbers of classes. That is, percentage correct may be misleadingly high when the data set in question has only a small number of examples from one class. Cohen's Kappa statistic takes the number of examples of each class into account and thus yields an unbiased measure of accuracy compared to percentages.

At the completion of the one thousand trials, the MLP with the highest kappa over the test data (that is, the MLP with the best generalisation performance) was selected as the winner for that species. The accuracy of this winning network was then evaluated over the validation data set, so that an unbiased estimate of the generalisation capability of the MLP could be obtained.

To produce the training data for the cascaded networks, the winning climate networks and the winning host assemblage networks for each species were selected. The relevant data for all regions were then propagated through each network. The outputs from these networks were used as the inputs for the cascaded networks, which were then trained to predict the presence or absence of the target disease species. The cascaded networks therefore combined predictions made from climate and host assemblages into one final prediction, as shown in Figure [1].

The training and testing procedure for the cascaded networks was the same as the climate and host networks above. A wider range of hidden-layer sizes were found to be useful for these networks, as shown in Table [3], while 500 training epochs and learning rate and momentum of 0.03 were found to be optimal.

Table 3. Number of hidden neurons selected for cascaded networks

Species	Hidden Neurons
<i>T. tritici</i>	9
<i>A. pisi</i>	6
<i>U. hordei</i>	10
<i>P. halstedii</i>	6
<i>P. tritici-repentis</i>	10
<i>L. taurica</i>	4

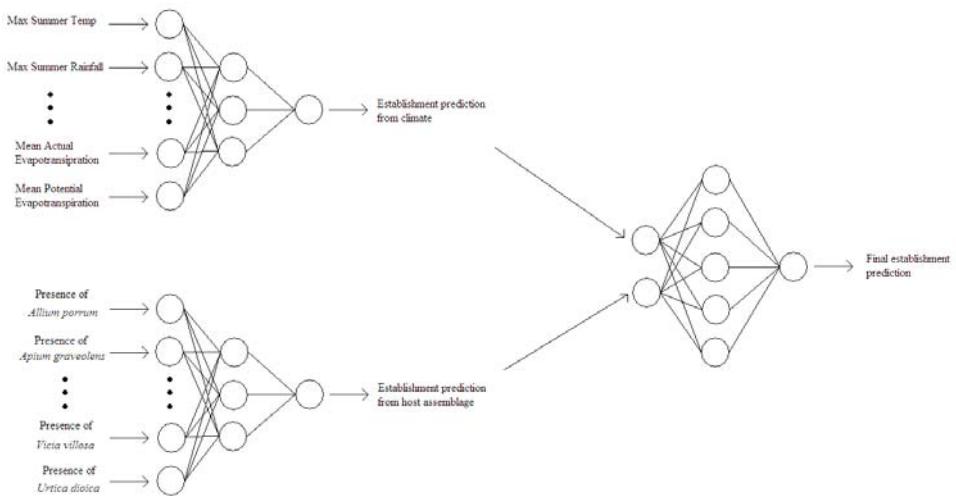


Fig. 1. Cascading MLP that combines predictions made from climate and host plant assemblages

3 Results

The accuracies for each species, as measured by Cohen's Kappa statistic [1], are presented in Table 4. These results show that there was very little over-training over either climate or host data for *T. tritici*, *A. pisi* and *U. hordei* - for each species, the mean testing accuracies are quite close to the mean training accuracies. A slight amount of over-training was apparent over the climate MLP for *L. taurica* and *P. halstedii* and more severely for *P. tritici-repentis*. Note that these last three are the less-prevalent species, which makes them more vulnerable to over-training. For the MLP that made predictions from plant assemblages, only *P. halstedii* and *P. tritici-repentis* showed significant over-training. There was a large amount of variation, as measured by the standard deviation, in the testing accuracies for the climate MLP, for all species but *T. tritici* and *U. hordei*: this ranged from 46% of the mean for *L. taurica* to 60% of the mean for *P. tritici-repentis*. Note that the low variation for *T. tritici* and *U. hordei* are due to these species being two of the three most prevalent. Variation over the host plant assemblage MLP is much higher, ranging from a low of 18% (again for *U. hordei*) to a high of 85% for *A. pisi*.

There was no sign of over-training for the cascaded MLP, with the testing accuracies being consistently high and close to the training accuracies. There was also very little variation over the testing accuracies, with a maximum of 10% of the mean for *P. halstedii*. This is almost certainly due to the fact that the training data for the cascaded MLP was derived from the best performing climate and plant assemblage MLP.

The validation accuracies show a steady increase in accuracy from climate to host, and from host to cascaded MLP, for all species except *L. taurica*. For this species, the host assemblage MLP were more accurate, although the difference in accuracy between host and cascaded networks is very small (0.04). These results show, however, that while both climate and host assemblage MLP are able to give good prediction of the presence or absence of the target fungi species, the cascaded networks, that combine predictions from both, perform the best overall. Only the MLP trained to predict the distribution of *P. tritici-repentis* performed relatively poorly, with a best validation Kappa of 0.38 - all other species had a validation Kappa of at least 0.46 which indicates a model that gives good prediction.

Table 4. Mean and standard deviation of accuracies measured as Cohen's Kappa for target species. "Train" is the accuracy over the training data set. "Test" is the accuracy over the test data set. "Validation" is the accuracy over the validation data set. "Climate" is the accuracy of the MLP trained over climate data only. "Hosts" is the accuracy over the MLP trained over regional host plant assemblages. "Cascade" is the accuracy of the cascaded MLP that combine predictions made from both climate and hosts.

<i>T. tritici</i>			<i>A. pisi</i>		
Train	Test	Validation	Train	Test	Validation
Climate	0.70/0.09	0.53/0.10	0.38	0.58/0.27	0.39/0.20
Hosts	0.75/0.14	0.40/0.10	0.44	0.47/0.40	0.26/0.22
Cascade	0.78/0.03	0.76/0.06	0.56	0.76/0.04	0.75/0.07
<i>U. hordei</i>			<i>L. taurica</i>		
Train	Test	Validation	Train	Test	Validation
Climate	0.68/0.05	0.55/0.07	0.44	0.44/0.17	0.26/0.12
Hosts	0.87/0.11	0.57/0.10	0.55	0.72/0.13	0.41/0.22
Cascade	0.84/0.02	0.84/0.04	0.67	0.71/0.04	0.71/0.06
<i>P. halstedii</i>			<i>P. tritici-repentis</i>		
Train	Test	Validation	Train	Test	Validation
Climate	0.53/0.24	0.30/0.16	0.36	0.46/0.19	0.20/0.12
Hosts	0.38/0.42	0.16/0.18	0.44	0.56/0.43	0.25/0.21
Cascade	0.76/0.04	0.73/0.07	0.58	0.81/0.03	0.81/0.05

4 Discussion

The results presented in this paper clearly show that MLP that combine abiotic and biotic factors in a cascaded architecture are able to predict the distribution of fungal crop diseases with a useful level of accuracy. Although there is a relatively large amount of variation in accuracies, the best-performing MLP were able to generalise to new data very well, especially for those MLP trained to predict more prevalent species.

As is the case with many ecological data sets, the data used in this study is likely to be very noisy. For example, while the climate in a particular region may be conducive to the establishment of a species, the species may never have gained access and therefore not become established in the region. Alternatively, while a species may be listed as being absent from a particular geographic region, this may be because it has never been officially recorded in that region, as opposed to being truly absent. Conversely, a species may be falsely recorded as being present in a region due to misidentification of a specimen. Alternatively, the species may have once been established, but since been exterminated.

While use of the maximum, minimum and mean of the climate variables provides useful information, in terms of providing the range of the variables for a region, there is a high degree of correlation between the mean and the other two statistics. There is also likely to be correlation between the climate variables themselves. This could be reduced by performing a principal components analysis (PCA) over the data and using only the top few principal components. However, it is desirable, for future work, to be able to identify which of the input variables to the climate and host plant assemblage MLP are the most significant [12][13]. A PCA transformation of the input data would remove the correlations, but may complicate the task of identifying the contribution of the original variables during the analysis of the MLP.

5 Conclusion

The results presented in this paper clearly show that it is possible to predict the distribution of fungal crop diseases. Predictions from abiotic variables are less accurate than predictions from biotic variables, while combining predictions from both, using a cascaded MLP architecture, yields accuracies that are, overall, superior to both.

Future work will focus on the analysis of the trained MLP to determine which input variables contribute most to the classification performance of the networks.

References

1. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 37–46 (1960)
2. Crop Protection Compendium - Global Module, 5th edn. ©CAB International, Wallingford, UK (2003)
3. De Wolf, E., Fransc, L.: Neural networks that distinguish infection periods of wheat tan spot in an outdoor environment. *Phytopathology* 87, 83–87 (1997)
4. De Wolf, E., Fransc, L.: Neural network classification of tan spot and *Stagonospora* blotch infection periods in a wheat field environment. *Phytopathology* 90, 108–113 (2000)
5. Dimopoulos, I., Chronopoulos, J., Chronopoulou-Sereli, A., Lek, S.: Neural network models to study relationships between lead concentration in grasses and permanent urban descriptors in Athens city (Greece). *Ecological Modelling* 120, 157–165 (1999)

6. Flexer, A.: Statistical evaluation of neural network experiments: minimum requirements and current practice. In: Trappl, R. (ed.) *Cybernetics and Systems 1996*, Proceedings of the 13th European Meeting on Cybernetics and Systems Research, Austrian Society for Cybernetic Studies, pp. 1005–1008 (1996)
7. Gotelli, N., Entsminger, G.: EcoSim: Null models software for ecology. Version 7. Acquired Intelligence Inc. & Kesey-Bear. Jericho, VT 05465 (2006)
8. Gevrey, M., Worner, S.P.: Prediction of global distribution of insect pest species in relation to climate using an ecological informatics method. *Environmental Entomology* 99(3), 979–986 (2006)
9. Gotelli, N.J.: Null model analysis of species co-occurrence patterns. *Ecology* 81(9), 2606–2621 (2000)
10. Joy, M.K., Death, R.G.: Predictive modelling and spatial mapping of freshwater fish and decapod assemblages using GIS and neural networks. *Freshwater Biology* 49, 1036–1052 (2004)
11. Lek, S., Delacoste, M., Baran, P., Dimopoulos, I., Lauga, J., Aulagnier, S.: Application of neural networks to modelling nonlinear relationships in ecology. *Ecological Modelling* 90, 39–52 (1996)
12. Olden, J.D., Jackson, D.A.: Illuminating the “black box”: a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling* 154, 135–150 (2002)
13. Olden, J.D., Joy, M.K., Death, R.G.: An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling* 178, 389–397 (2004)
14. Paul, P.A., Munkvold, G.P.: Regression and artificial neural network modeling for the prediction of gray leaf spot of maize. *Phytopathology* 95, 388–396 (2005)
15. Pietravalle, S., Shaw, M.W., Parker, S.R., van den Bosch, F.: Modeling of relationships between weather and *Septoria tritici* epidemics on winter wheat: A critical approach. *Phytopathology* 93, 1329–1339 (2003)
16. Pivonia, S., Yang, X.: Assessment of the potential year-round establishment of soybean rust throughout the world. *Plant Disease* 88, 523–559 (2004)
17. Prechelt, L.: A quantitative study of experimental evaluations of neural network learning algorithms: Current research practice. *Neural Networks* 9(3), 457–462 (1996)
18. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* 323, 533–536 (1986)
19. Sutherst, R.W., Maywald, G.F.: A computerized system for matching climates in ecology. *Agriculture Ecosystems Environment* 13, 281–299 (1985)
20. Watts, M.J., Worner, S.P.: Using artificial neural networks to determine the relative contribution of abiotic factors influencing the establishment of insect pest species. *Ecological Informatics* 3(1), 64–74 (2008)
21. Worner, S.P., Gevrey, M.: Modelling global insect pest species assemblages to determine risk of invasion. *Journal of Applied Ecology* 43, 858–867 (2006)

Using Time Lagged Input Data to Improve Prediction of Stinging Jellyfish Occurrence at New Zealand Beaches by Multi-Layer Perceptrons

David R. Pontin¹, Sue P. Worner¹, and Michael J. Watts²

¹ Bio-Protection and Ecology Division, Lincoln University,
P.O. Box 84, Canterbury, New Zealand

² School of Biological Sciences, University of Sydney, NSW 2006, Australia
david.pontin@lincoln.ac.nz, [worner@lincoln.ac.nz](mailto>worner@lincoln.ac.nz)

Abstract. Environmental changes in oceanic conditions have the potential to cause jellyfish populations to rapidly expand leading to ecosystem level repercussions. To predict potential changes it is necessary to understand how such populations are influenced by oceanographic conditions. Data recording the presence or absence of jellyfish of the genus *Physalia* at beaches in the West Auckland region of New Zealand were modelled using Multi-Layer Perceptrons (MLP) with time lagged oceanographic data as input data. Results showed that MLP models were able to generalise well based on Kappa statistics and gave good predictions of the presence or absence of *Physalia*. Moreover, an analysis of the network contributions indicated an interaction between wave and wind variables at different time intervals can promote or inhibit the occurrence of *Physalia*.

1 Introduction

The oceans of the world are undergoing fundamental shifts in their environments, primarily through human causes [1]. Factors such as changing temperatures and currents have created the possibility for the distribution of many pelagic marine species such as jellyfish to expand or at least change, with ecosystem level repercussions. Such species cause problems for swimmers around beaches, block nets used in aquaculture enterprises such as salmon farms and large numbers can deplete fish stocks by severely depleting eggs, small larvae and plankton [2]. It has been noted in several studies that there is circumstantial evidence that jellyfish (Cnidaria) populations are increasing [3, 4]. Such studies suggest that the changing marine environment is the main cause. To assess if this is true large scale population movements of pelagic species need to be understood. In this study we used marine jellyfish as a model species. Little is known about how pelagic jellyfish populations move within an area or are transported into new areas. Population movements are difficult to determine because few large scale datasets exist and consequently little work has been done in this area. One of the few studies that have attempted to model pelagic jellyfish population dynamics over a large scale was carried out by [5] who modelled *Chrysaora quinquecirrha* in the Gulf of Mexico. Their aim was to estimate

possible suitable locations for an intermediate life-stage of this species using information from the Gulf of Mexico circulation model. Their model predicted potential areas in which the intermediate life-stage could settle and complete their lifecycle. There is a growing need for this type of information for other species of jellyfish so that bloom patterns and species ranges can be understood and predicted.

The genus *Physalia* is one of the most common jellyfish in New Zealand. *Physalia* is considered to be one of the more primitive living jellyfish as it lacks many of the morphological characteristics associated with species that evolved later [6, 7]. In particular *Physalia* lack any swimming mechanisms which mean that *Physalia* is completely dependant on ocean winds and currents for movement [8]. The only adaptation for movement *Physalia* possess is the float, and there are two morphs one with a left hand sail and one with a right hand sail, allowing individuals to move at slightly different angles in the same wind condition [9, 10]. These characteristics mean that potentially, *Physalia* population movement can be modelled based on wind, current and swell information. For this reason *Physalia* is an ideal model species to investigate the problem of predicting the occurrence of pelagic species populations based on oceanographic data.

In an initial study [11] we explored the possibility that presence or absence of *Physalia* could be modelled with Artificial Neural Networks (ANN) using data sourced from Surf Lifesaving New Zealand (SLSNZ). A simple approach was used, where time lags between data points were ignored and it was showed that it was possible to model the data with moderate accuracy. The aim of this study was to use a time lag in the input data to better represent the relationship between the data and the model. With a better representation of the input data the influence of oceanographic variables on *Physalia* presence at beaches can be more precisely assessed. Such information should generate hypotheses concerning the species rate of movement and potential spawning grounds.

2 Methods

2.1 Data

Oceanographic Data: Oceanographic data was sourced from the National Institute of Water and Atmosphere (NIWA). The data contained time series outputs from NOAA/NCEP Wavewatch III model hindcast [12] representing eighty 1:25⁻¹ degree global grid cells surrounding New Zealand. Each cell contained three-hourly measurements of five variables (significant wave height (m), peak wave period (s), peak wave direction ($^{\circ}$ N) and U and V wind vector components ($m s^{-1}$)). MATLAB ® was used to transform and manipulate the files so that they were able to be incorporated in the models. All variables were transformed to daily data points, by averaging each of the eight data points for each day. Furthermore, from the U and V wind vector components, wind velocity ($m s^{-1}$) and direction were calculated. The circular mean was used for all directional variables. Once the transformations had been completed each file contained daily data for significant wave height (m), peak period (s), peak direction ($^{\circ}$ N), wind velocity ($m s^{-1}$) and wind direction ($^{\circ}$ N). For each region, data from a cell was included if the cell was less than 250km distant from the

centre of the region. Time lags were created by time-stepping the data from one to six days. In other words data from one to six days prior was included into the final datasets. For this study the oceanographic data from the West Auckland region in New Zealand was extracted (Figure 1). The West Auckland datasets contained variables from six ocean cells, with five variables each for each day that was time-stepped, and an index for the date measured from the 1st of October for each year (Table1). Furthermore, a dataset was created for window length of one in which the date index was removed to determine if the addition of a date index of this nature was appropriate.

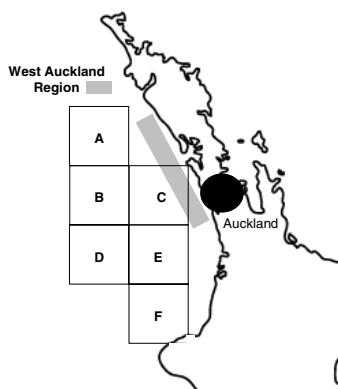


Fig. 1. Oceanic cells associated with the West Auckland region

Surf Lifesaving Data: Data recording jellyfish incidents was sourced from Surf Lifesaving New Zealand (SLSNZ). SLSNZ maintains an electronic database of all patrol records including all incidents of jellyfish stings. We accessed the records of patrols carried out from the 2000/2001 season to the 2004/2005 season. Data from the West Auckland region was used in this study. The West Auckland region comprises 8 patrolled beaches with a mean patronage of 290 people on each beach (SEM 18.96). Records that showed a beach headcount of zero, in other words when nobody was swimming, were excluded for that data, as clearly there will be no jellyfish incidents. This resulted in a minimum of 56 swimmers recorded in the district at one time. An occurrence corresponded to one or more of the beaches in the region recording a jellyfish sting. It is acknowledged that there will be correlation between the numbers of swimmers present and the number of stings but using presence/absence data across the region will minimise the effect of either a large or small number of swimmers present. The use of the SLSNZ data restricted the study to dates from late southern hemisphere spring to early autumn as this is the time when lifeguards patrol the beaches. The West Auckland dataset contained 432 data points of which 100 (23%) represented the presence of *Physalia*.

2.2 Training and Evaluation of MLP

Standard three neuron-layer MLP were used to model the data, and the learning algorithm used was unmodified back-propagation with momentum. Each network

output indicated the predicted presence or absence of *Physalia* at the region on that particular day. The method of training and evaluating the MLP (and also selecting the parameters) was similar to that suggested in [13, 14]. A number of runs were carried out using different combinations of hidden neuron layer size, learning rate and momentum. Each run consisted of 100 trials. For each trial, the training and test dataset was randomly divided into a training set, consisting of two-thirds of the available data, and a test set consisting of the remaining one-third. An MLP was then created with randomly initialised connection weights and trained over the training data set. The accuracy of the MLP over the training set was then evaluated to determine how well the network had learned the training data. The accuracy of the MLP was then evaluated over the testing dataset to determine how well the network generalised. Accuracy was measured as both the percentage of examples correctly classified and using Cohen's Kappa statistic [15]. Whereas percentage accuracy is easily interpreted, it is also easily biased by unbalanced classes. In other words, percentage correct may be misleadingly high when the dataset in question has only a small number of examples from one class. The Kappa statistic takes the number of examples of each class into account and thus yields a less biased measure of accuracy than percentages.

For each trial the contributions of each input neuron to the output of the network was also determined, using the method of Olden and Jackson as described in [16]. This method has been experimentally determined to give the least-biased estimate of the contribution of each input neuron [17] and has been used previously in ecological modelling applications [18].

At the completion of the runs, the run with the highest mean kappa over the test sets was selected as the winner for the window length for that region. The accuracy of the networks within this run was then evaluated over the validation dataset. The results were then compared and the best dataset and hence time lag was selected. A sensitivity analysis was then performed over the significant input variables of the best-generalising network within that run. Sensitivity analysis was carried out to determine the response of the network to variations in the input variables so that the influence of strongly contributing inputs could be investigated.

3 Results and Discussion

3.1 Training Parameters

The optimal training parameters for each dataset, as determined by testing accuracy, are presented in Table 1. In general, datasets with lower window sizes optimised with fewer neurons and epochs than larger window sizes. An exception was a window length of 6 days which required fewer epochs than for a window length of 5 days.

3.2 Accuracy

The accuracies of MLPs for each region are presented in Table 2. Overall percentage correct prediction and Cohen's Kappa statistic were calculated. The inclusion of a date index spanning the season significantly improved testing accuracy ($\kappa 0.63$) compared with the same data set without the date index ($\kappa 0.22$) (<0.001 , T test). Furthermore the accuracies for this data set for both the train and test set were an improvement on those found by [11] ($\kappa 0.35/0.25$), where time was represented by months using orthogonal binary encoding, further justifying the inclusion of the date index.

Table 1. Number of variables that each dataset contained and optimal training parameters by region. “neurons” is the number of hidden layer neurons and * represents the dataset without a date index.

Window length	Number of variables	neurons	learning	momentum	epochs
1*	30	3	0.15	0.7	500
1	31	3	0.15	0.2	500
2	61	6	0.1	0.05	500
3	91	7	0.1	0.05	600
4	121	7	0.1	0.05	1000
5	151	7	0.1	0.05	1000
6	181	7	0.1	0.1	600

The dataset that produced the highest testing Kappa for jellyfish presence had a window length of one and therefore contained oceanographic data that only occurred on the day of the event. This dataset was not considered for further analysis as it lacked ecological relevance because conditions that occurred in distant cells would not influence what was happening at beaches. The dataset that had the best test accuracy and had ecological relevance was the dataset that had a five day window (κ 0.40). This model however, lacked the ability to generalise (κ 0.27) compared to the dataset with a six day window (κ 0.38) which had a similar testing accuracy (κ 0.39). This possibly indicates that overtraining has occurred and further exploration of parameters surrounding the current optimum parameters may rectify this issue.

Table 2. Mean and standard deviation of performance criteria for the training, test and validation datasets. The performance criteria are overall percentage accuracy (%) and Cohen's Kappa statistic (κ). Note * represents the dataset without a date index.

Window length		Train	Test	Validation
1*	%	85.1/2.57	74.2/4.41	77.0/5.75
	κ	0.54/0.11	0.22/0.10	0.50/0.06
1	%	89.0/1.97	81.6/3.25	76.3/3.12
	κ	0.77/0.04	0.63/0.07	0.30/0.13
2	%	82.0/2.25	79.0/4.06	59.6/1.53
	κ	0.73/0.08	0.32/0.09	0.38/0.09
3	%	96.9/1.16	77.8/4.21	76.7/4.19
	κ	0.91/0.04	0.38/0.10	0.33/0.08
4	%	99.4/0.76	76.2/4.22	78.7/3.78
	κ	0.98/0.02	0.36/0.10	0.38/0.09
5	%	99.2/0.60	79.3/3.17	71.2/3.42
	κ	0.97/0.02	0.40/0.09	0.27/0.08
6	%	99.3/0.76	78.5/4.25	78.2/2.92
	κ	0.98/0.02	0.39/0.10	0.38/0.08

3.3 Contributing Variables

For the dataset which had a five day window, the top eight variables that contributed to network prediction are presented in Table 3. Wind speed in cell E both on the day

of the event and the day prior to the event had significant negative contributions. That means when there is wind in the cell the probability of *Physalia* occurring decreases. The interpretation of this result is counterintuitive but is reasonable when the result of the sensitivity analysis is considered (Figure 2b). The sensitivity analysis indicates that there is a wind threshold of approximately 7.2 m s^{-1} before which probability of *Physalia* occurring decreases. Therefore if there are winds less than 7.2 ms^{-1} in cell E on the day of the event, the model decreases the probability of an occurrence but if the wind is greater than 7.2 ms^{-1} then the model does not decrease the probability of an occurrence. This is realistic for the variable as low wind speed would delay the transportation of a swarm onto the beaches whereas higher wind speed would only facilitate the movement across the cell and not influence occurrence on beaches. Wind direction in cell C on the day prior to an event and in cell F three days prior to the appearance of jellyfish also made a significant negative contribution. In other words, as wind direction in these cells becomes more northerly, the probability of *Physalia* presence decreases. An explanation for this observation is that a Northerly wind would blow the swarm past the beach. In the case of a northerly wind in Cell F the jellyfish are likely to be blown toward the Taranaki coastline and outside the region investigated.

Table 3. Variables that were identified as the most influential contributing to the activation of the output. The letter and number after each variable indicates the oceanographic cell (Figure 1) in which the variable was measured and how many days prior to the data point the data was taken.

	Variable name	Contribution
Negative variables	Wind speed E-0	-40.7/12.2
	Wind direction C-1	-37.5/14.2
	Wind direction F-3	-33.9/15.2
	Wind speed E-1	-30.5/9.80
Positive variables	Wave period A-3	42.6/13.5
	Wave height F-3	34.6/12.2
	Wave period B-3	31.6/11.0
	Wind direction D-2	30.7/14.3

Wave period in cells A and B three days prior to the event had significant positive contribution. An increase in wave period denotes that the waves have been generated further out to sea [19] indicating that there has been sustained conditions that are likely to transport the jellyfish into the region and increase their probability of an incident at the beach. The time lag is also highly relevant as it takes time for a swarm to travel from the outer cells to the beaches and corresponds well to reality. An increase in wave height in cell F three days prior causes an increase in the likelihood of *Physalia* occurrence. That could be through either waves being generated from wind forcing within the cell or a large swell travelling through the cell both of which would create conditions that would facilitate the transportation of *Physalia* into the region. Similarly, as wind direction tended towards the north that would cause a

swarm to be blown towards the coastline. That may account for the significant positive contribution of wind direction in Cell D two days prior to an event. One important aspect the contributions indicated that wave conditions in the outer cells of the region two to three day prior to an event had a positive contribution to jellyfish incidence whereas wind conditions on the day or the day prior to an event in the inner cells could reduce the presence of *Physalia*. These results are interesting as they seem to show the particular oceanographic conditions necessary to transport swarms into the region but that wind conditions close to the shore can dictate whether or not a swarm reaches close to the beach.

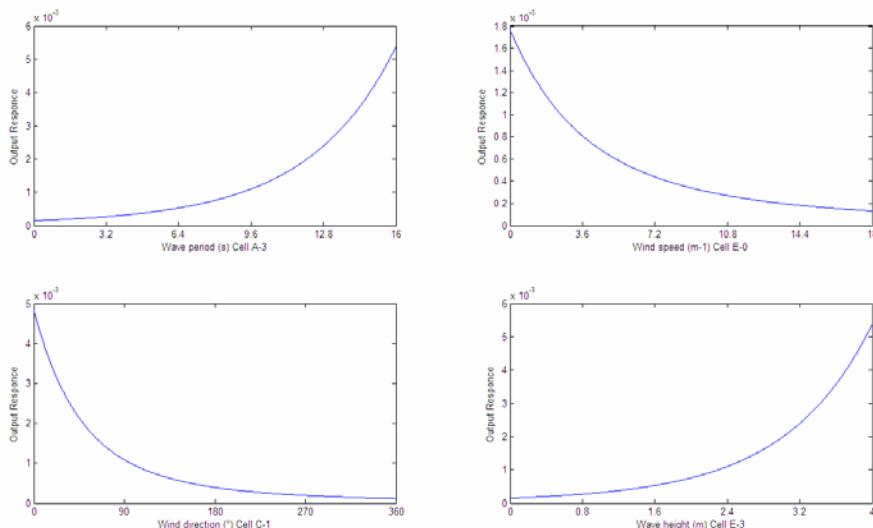


Fig. 2. Sensitivity analysis of the most significant continuous variables for the West Auckland region

4 Conclusion

This study clearly shows the potential for using MLP to predict the presence or absence of *Physalia* at beaches around New Zealand and that time lags within the data are important for good prediction. The results showed that MLP can learn to predict to a reasonable degree of accuracy the presence of *Physalia* in a target region using time lagged oceanographic data. Furthermore, the contribution analysis generated valuable ecological information to assist interpretation.

Future work will expand the study to other regions of New Zealand, and will investigate methods to improve performance, such as reducing the number of input variables by removing those that are highly correlated. We will also develop a better likelihood value for the presence of *Physalia* so that the models can be used as a warning mechanism for the general public at beaches.

References

1. Arai, M.N.: Pelagic coelenterates and eutrophication: a review. *Hydrobiologia* 451, 69–87 (2001)
2. Purcell, J.E., Uye, S., Lo, W.: Anthropogenic causes of jellyfish blooms and their direct consequences for humans: a review. *Marine and Ecology Progress Series* 350, 153–174 (2007)
3. Purcell, J.E.: Climate effects on formation of jellyfish and ctenophore blooms: a review. *Journal of the Marine Biological Association of the United Kingdom* 85, 461–476 (2005)
4. Mills, C.E.: Jellyfish blooms: are populations increasing globally in response to changing ocean conditions? *Hydrobiologia* 451, 55–68 (2001)
5. Johnson, D.R., Harriet, P.M., Burke, D.W.: Developing jellyfish strategy hypotheses using circulation models. *Hydrobiologia* 451, 213–221 (2001)
6. Collins, A.G.: Phylogeny of Medusozoa and the evolution of cnidarian life cycles. *Journal of Evolutionary Biology* 15, 418–432 (2002)
7. Dunn, C.W., Pugh, P.R., Haddock, S.H.D.: Molecular phylogenetics of the siphonophora (Cnidaria), with implications for the evolution of functional specialization. *Systematic Biology* 54, 916–935 (2005)
8. Lane, C.E.: The Portuguese Man-of-War. *Scientific America* 2002, 158–168 (1960)
9. Barnes, R.: *Invertebrate Zoology*. Saunders College, Philadelphia (1980)
10. Totton, A.K.: Studies on *Physalia physalis*: Natural history and morphology. *Discovery Reports* 30, 301–368 (1960)
11. Pontin, D.R., Watts, M.J., Worner, S.P.: Using Multi-Layer Perceptrons to Predict the Presence of Jellyfish of the Genus *Physalia* at New Zealand Beaches. In: International Joint Conference on Neural Networks, Hong Kong, pp. 1171–1176 (2008)
12. Tolman, H.L.: Validation of a new global wave forecast system at NCEP. In: Edge, B.L., Helmsley, J.M. (eds.) *Ocean Wave Measurements and Analysis*. ASCE, pp. 777–786 (1998)
13. Flexer, A.: Statistical Evaluation of Neural Network Experiments: Minimum Requirements and Current Practice. In: Trappl, R. (ed.) *Cybernetics and Systems 1996*, Proceedings of the 13th European Meeting on Cybernetics and Systems Research. Austrian Society for Cybernetic Studies, pp. 1005–1008 (1996)
14. Prechelt, L.: A Quantitative Study of Experimental Evaluations of Neural Network Learning Algorithms: Current Research Practice. *Neural Networks* 9, 457–462 (1996)
15. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 37–46 (1960)
16. Olden, J.D., Jackson, D.A.: Illuminating the “black box”: a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling* 154, 135–150 (2002)
17. Olden, J.D., Joy, M.K., Death, R.G.: An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling* 178, 379–389 (2004)
18. Joy, M.K., Death, R.G.: Predictive modelling and spatial mapping of freshwater fish and decapod assemblages using GIS and neural networks. *Freshwater Biology* 49, 1036–1052 (2004)
19. Toba, T., Lida, N., Kawamura, H., Educhi, N., Jones, I.S.F.: Wave dependence of sea-surface wind stress. *Journal of Physical Oceanography* 20, 705–721 (1990)

Modelling Climate Change Effects on Wine Quality Based on Expert Opinions Expressed in Free-Text Format: The WEBSOM Approach

Subana Shanmuganathan and Philip Sallis

Centre for Geoinformatics Research, School of Computing and Mathematical Sciences
Auckland University of Technology, New Zealand
subana.shanmuganathan@aut.ac.nz

Abstract. The motivation for modelling the effects of climate change on viticulture and wine quality using both quantitative and qualitative data within an integrated analytical framework is described. The constraints and solutions evident when taking such an approach are outlined. WEBSOM is a novel self-organising map (SOM) method for extracting relevant domain-dependent characteristics from web based texts and in this case, investigated for modelling wine quality resulting from climate variation, by web text mining published descriptions made by sommeliers about this phenomenon. This paper describes experiments using the WEBSOM method with their results.

1 Introduction

Climate change impacts on all forms of agriculture and vegetation. Current awareness of climate change and the phenomenon known as ‘global warming’ has increased scientific and commercial interest in it and predictions relating to it. The potential influence of climate variation on viticulture and *oenology* is considered as *dramatic*. Historical viticulture records show that the ability of winemakers to produce premium quality wine is *highly prone* to climate change; both short and long term. Literature reveals that in the past viticulturists and winemakers adopted subtle changes to cultivation practices and winemaking processes to overcome short term or annual climate change effects. Similarly, major shifts on cultivation in whole wine-producing regions as well occurred to overcome the long term climate effects. These can be seen as occurring over centuries to turn the climate variation effects to producing finer wine by producing grapes with a higher percentage of sugar but without comprising the other aroma, flavour and colour protein compounds in the berry ripening process. Hence, these characteristics of wine quality are considered to be the principal factors relating to climate for viticulture regions throughout the world. Irrigation, frost, wind and solar influences are also major factors in grape production of course and therefore, are determinants of wine quality [1], [2]. The main objective of the overarching research project (called *Eno-Humanas*, see www.geo-informatics.org, also [3] and [4]) is to build models based on correlations of dependent variables in the combinatorial set of relationships that comes from data collected relating to climate, atmosphere,

soil, terrain, moisture and plant response in association with sensory perception data relating to flavour, odour and fruit robustness. In essence, *Eno-Humanas* is about analysing precise ecological data and the less precise qualitative opinion data that comes from wine consumers, within an integrated framework.

This paper is about the second aspect of the data analysis and it analyses free text descriptions of wine quality coming from experts; Master Wine Sommeliers as explained in section 2. Section 3 outlines the results of Kohonen self-organising map (SOM)¹ based text mining approach to establishing implicit annual variations within expert opinions on 95 New Zealand wines, expressed in free-text extracted from a web magazine called *Web Enthusiast* [5]. Section 4 elaborates upon a novel WEB-SOM approach [6] being investigated for this purpose with the same sample data set investigated with SOM. The final section of this paper proposes future research to model the effects of climate change in greater detail with larger data sets from more grape growing regions and to look at the climate change effects on the world's major wine regions. From this analysis we expect to be able to predict the wine style and appellations suitable for future climate change, short and long term, using climate change data from models already developed.

2 The Effects of Climate Change on Viticulture

Climate change is predicted to bring about significant modifications to all forms of agriculture and vegetation on earth [7]. However, its potential impact on Viticulture, the world's most expensive cultivated crop and *enology* as the science that underpins it, suggests the variability across the globe to be inconsistent (severe in the northern hemisphere and mild in the southern) and to have a variable effect on different grape-vine varieties. Grapevine phenology, such as crop budburst, *floraison*, *veraison*, and harvest, greatly depends on weather and climate conditions in different regions, and this is a major factor in determining wine quality. For example, even a single degree centigrade change in temperature is predicted to impact on the production of the world famous Mediterranean wine appellations. Grape varieties thrive under significantly narrow/ niche climate and environmental conditions, and historical evidence as well supports this fact (see Introduction). Research findings with Australian grape-vines and wines [8] suggest that a change of grapevine varieties could be a way to overcome the future climate variation effects in that country's major wine regions. This would of course, be an extremely expensive exercise [9], which is why objective scientific analysis for scenario building and prediction is of great significance at the moment. Here onwards the paper looks at finding implicit, relevant and meaningful taste quality descriptors, indicative of annual climate change effects, within sommeliers comments of 95 New Zealand wines.

¹ A SOM (self-organising map) is a single layered, feed forward artificial neural network (ANN), useful in displaying multi D data sets on low D maps while preserving the topology of the data. ANNs are collections of elements called 'neurons' with weighted connections between them. The neuronal structure, the connections, training and recall algorithms define the network architectures that mimic biological nerve/brain cell structures. ANNs became more popular since the 1960s due to their ability to resolve issues relating to conventional algorithmic computing methodologies.

3 Previous Research

This section of the paper discusses past efforts from literature and that of our own research with wine quality and sensory data analysis. It also describes the WEBSOM.

3.1 Wine Sensory Analysis

The wine quality literature relating to sensory and chemical data analysis can be broadly categorised into the following:

Wine characterisation and discrimination using chemical and sensory properties: Most of the papers looked at fall into this category. Wine of all main appellations, such as, *champagne*, *chardonnay*, and *pinot noir*, have been studied to find the best way for identifying the differences in support of the sub-appellations within that of main. A classification for a distinctive New Zealand wine *Marlborough Sauvignon Blanc* using sensory characterisation and chemical analysis of selected aroma compounds is discussed in [10]. An investigation into designating three sub-appellations of red Niagara Peninsula *Bordeaux* style based on differences in chemical and sensory analysis on forty-one commercially available wines is reported on [11]. Similarly, [12] and [13] looked at strategies a) to establishing control over *champagne* wine quality based on sensory and b) red table wine quality characterised by pleasing and complex mouth-feel sensations respectively for these regions.

Professional versus novice taster abilities: There are many studies in this area cited in the literature and another project within *Eno-Humanas* is considering this from an audio-mining perspective to elicit the degree of emphasis (passion) expressed about wine quality in recording of wine tasting by both professionals and novices.

Wine ratings, favourable climatic conditions and price fluctuations: Research on this subject described in [14] looked at climate and global wine quality factors and discussed a year-to-year comparison over a ten year period. It includes a description of wine quality factors in juxtaposition with prices and vintage ratings. Citing many earlier studies the authors of this work pointed out that the analysis of the relationships between climate variables and wine prices as based on an underlying hypothesis that beneficial climate conditions would improve the wine quality and that in the past, these had in turn led to short term price hikes. The study also reflected that the unavailability of consistent price data for multiple regions and with different styles over many years to be a shortcoming for any analysis on studying long term effects. The vintage ratings are seen as a strong determinant of the annual economic success of a wine region but not necessarily a predictor [15] [16].

Analysis of wine taster descriptions in free-text: There are not many studies of this kind cited in the literature. Of the studies reviewed, two major approaches are outlined herein. Taster comments analysis discussed in [17] investigated into the structure of the language used by different wine experts with a software product called ALCESTE. In this study, analysts grouped the word categories in different expert corpuses by calculating the chi square of co-occurrence of words and classified the categories into different classes, such as *idealistic*, *odour*, *colour*, *somesthesiaic*, *taste* and *hedonistic*. The study concluded that the language structure used by wine experts

as not organised along their sensory dimensions instead with wine prototypes, this is reflected in a recent study [18] as well.

In another interesting piece of work [19], researchers looked at calculating synthetic liking scores by studying the correlations between pairs of original scores and word groups/counts in free text comments and then comparing these synthetic scores with that of the real for a sample set of wines studied. The authors used multiple factor analysis to establish the correlations between wine comment-liking score pairs studied.

Research conducted at University of California Davis [20] found that only 25% of wine liking ratings to be linked to wine sensory descriptive data in a map created with statistical analysis results of the latter on y axis and ratings on x axis. The authors as well found some descriptors, such as “leather” and “sour”, as having a negative effect and a few others as preferred, such as “vanilla/oak” and “canned vegetable”. Furthermore, noted even though 75% of the variations in liking scores could not be explained, the results should be read with caution.

3.2 WEBSOM Applications

WEBSOM provides an efficient methodology for full-text information retrieval and exploration of large collections of documents. It uses SOMs to statistically analyse the relations between words based on their co-occurrence in documents, and then based on the relationships, creates document maps. As the word co-occurrence details are used as basic components for SOM clustering, similar documents get clustered close to each other on a map of documents. The WEBSOM approach first developed for creating maps of documents could be applied to search and exploration of documents [21] with unsupervised [22], or partially supervised in processing of newsgroups [23], browsing interface for web pages for the exploration of document collections [24] and as a method/ tool for data mining in textual databases [25].

In this research, WEBSOM is used to modelling any implicit year-to-year variations that may exist between groups of words (depending on their co-occurrences) found in sommelier comments, that best describe wine quality/ wine sub appellations in terms of appearance, aroma and mouth-feel; this is possible as wine experts tend to describe wine quality relating to prototypes of wine style and quality (see section 3.1 Wine sensory analysis).

3.3 Text Mining of Sommelier Comments Using SOM

As a first step in implementing the *Eno-Humanas* concept to model the correlations between climate change effects on grapevine phenology that in turn impact on wine quality with ecological data and sommelier/ wine taster comments, the latter is being investigated herein using sommelier comments extracted from a popular web based wine catalogue called *Wine Enthusiast* [3]. The initial results of SOM based text mining approach to analysing the comments of 95 New Zealand wine (in free text format) are discussed in this section.

After removing stop words, such as a, the, in and etc, and stemming the comments using perl scripts, a matrix of words (*lemmas*) was created from the wine taster comments (Fig. 1). Weights w_i of (1) for the selected words were calculated by applying

the well-known information retrieval system called Salton's vector space model, which is based on a) local information from individual documents and b) global information from the collection of documents.

$$w_i = tf_i * \log\left(\frac{D}{df_i}\right) \quad (1)$$

Where,

tfi = term frequency (counts) or number of times a term i occurs in a document.

df_i = document frequency or number of documents containing term i

D = number of documents in the collection/database.

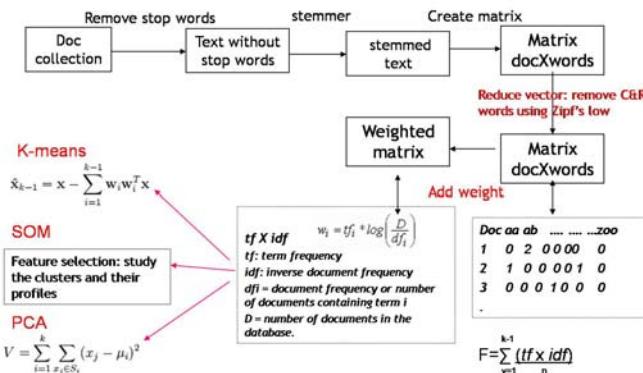


Fig. 1. Diagram on the steps used to create a word matrix of 95 New Zealand wines

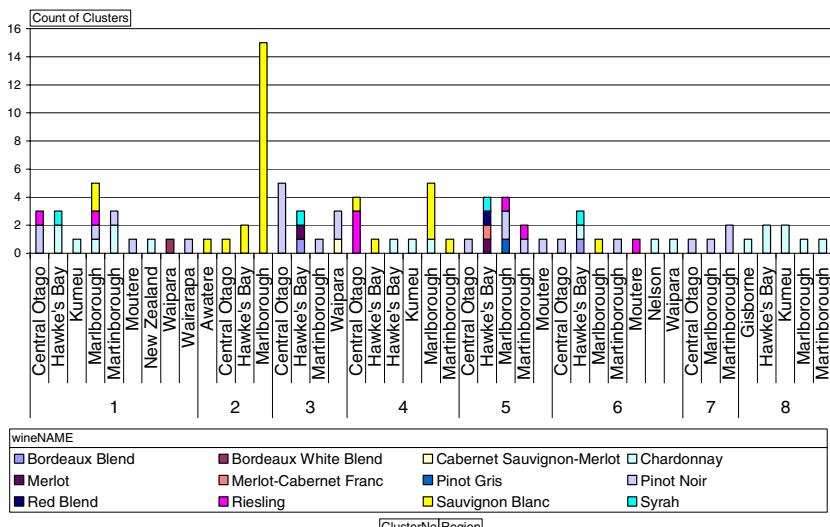
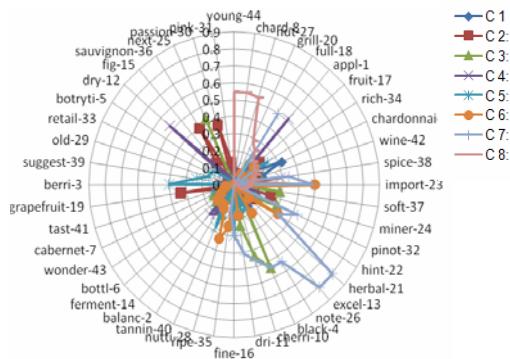
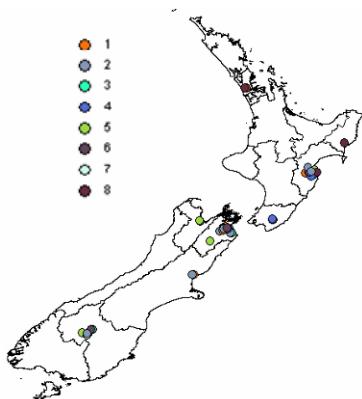
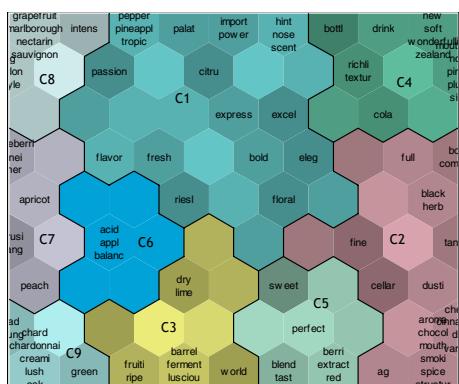


Fig. 2. Histogram showing the number of wines in different clusters (y axis) and regions (x axis) of a SOM created with 95 NZ wine word frequencies as per Fig 1 and formula (1).

**Fig. 3.** SOM cluster profile radar (word average)**Fig. 4.** 95 NZ wine clustering projected on DIVA map**Fig. 5.** SOM of word frequencies. SOM clusters C1-C9 show the grouping of words based on their co-occurrences in 95 New Zealand sommelier comments being analysed.

4 The WEBSOM Approach to Text Mining Sommelier Comments

The WEBSOM approach to wine expert comments analysis aims to establish any implicit annual variations within the comments and uses the same matrix of words used in the SOM analysis (fig 1). The WEBSOM results are discussed in this section.

no	Who	C 1:	C 2:	C 3:	C 4:	C 5:	C 6:	C 7:	C 8:	C 9:	year	wtype	Region	Cluste	price	year	rate	Comments
75	w75	0.19	0	0	0	0	0	0.31	0.13	0	2004	Sauv	Awate	1	20	2004	91 bell pepper and	
57	w57	0.04	0.05	0	0.02	0	0	0	0	0.04	2002	Pinot	Centr	1	0	2002	91 the region this is the	
60	w60	0.03	0.07	0	0.11	0	0	0	0	0	2002	Pinot	Centr	2	43	2002	91 the bottle in a slightly	
58	w58	0.03	0.1	0	0	0.09	0	0	0	0.06	2002	Pinot	Centr	4	0	2002	91 this is a	
62	w62	0.14	0.29	0	0.01	0.18	0	0	0	0	2003	Pinot	Centr	5	62	2003	91 and no it's not	
61	w61	0.05	0.23	0	0.06	0	0	0	0	0	2004	Pinot	Centr	2	60	2004	91 balcony edge to broad	
63	w63	0.08	0.09	0.04	0.23	0	0	0	0	0.04	2005	Pinot	Centr	2	63	2005	91 of reduction on the	
25	w25	0.02	0.36	0	0.33	0.07	0.21	0	0	0	2005	Pinot	Centr	4	35	2005	92 award winner back in	
59	w59	0.02	0.22	0	0.46	0	0.2	0	0	0	2005	Pinot	Centr	1	40	2005	91 wonderfully silky	
26	w26	0.02	0.11	0	0.18	0	0	0	0	0.04	2006	Pinot	Centr	4	42	2006	92 change from	
64	w64	0	0.24	0	0.36	0	0	0	0	0	2006	Pinot	Centr	5	63	2006	91 structured this is the	
71	w71	0.07	0	0.31	0	0.06	0	0.03	0.06	0.01	2001	Rieslir	Centr	2	21	2001	91 peaches and	
32	w32	0.42	0.07	0.12	0.01	0.12	0	0.13	0.06	0	2002	Rieslir	Centr	5	23	2002	92 expression of Riesling	
72	w72	0.14	0.11	0.11	0	0	0.69	0.02	0	0.1	2002	Rieslir	Centr	2	23	2002	91 Riesling is loaded	
73	w73	0.04	0.08	0	0.02	0.05	0.37	0.19	0	0	2006	Rieslir	Centr	1	27	2006	91 the sugars are amply	
11	w11	0.09	0.02	0.28	0	0.19	0.49	0.07	0	0.23	2001	Sauv	Centr	1	16	2001	93 s no oak aging or	
76	w76	0.12	0	0	0.06	0	0	0	0	0.3	2004	Sauv	Centr	1	20	2004	91 requisite pea pepper	

Fig. 6. Table showing the different descriptors & their frequencies in wine taster comments that reflect regional, wine style and annual variations in which *Pinot noir* wines from Central Otago, do not consist of C7-8 words. Two of 2005 wines, consists of C6 words in addition to C1, C2, C4 & C5 descriptors.

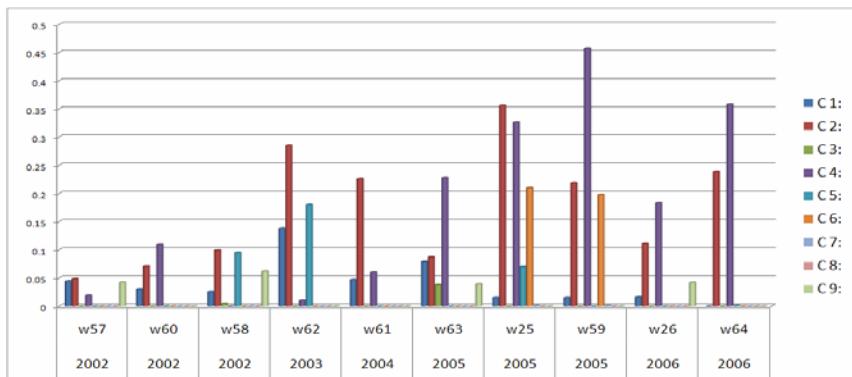


Fig. 7. Histogram showing the word frequencies of Central Otago's *Pinot noir* wines

4.1 WEBSOM Results

The different groups of wine words/descriptors from the WEBSOM approach that is based on the co-occurrences of words in the expert comments of 95 New Zealand wines, are given in fig. 5. Of these groups, C1 words/lemmas (*pepper*, *pineappl*, *tropic*, *palat*, *import power*, *herbal hint*, *nose scent*, *passion*, *citrul*, *express*, *excel*, *flavor*, *fresh*, *bold*, *eleg*, *riesl*, *floral*) are used in most of the wine comments hence, this cluster could be further divided or some discretion is needed in the selection of words to make the wine descriptor grouping more meaningful.

Based on the word frequencies of *Pinot noir* wines from Central Otago (fig. 6 & 7), 2005/6 wines have SOM cluster C4 descriptors (*bottl*, *drink*, *layer new soft*, *wonderfulli zealand*, *richli textur*, *mouthfeel noir pinot plum silki*, *cola*) at higher frequencies. C 2 descriptors (*full*, *bodi complex*, *black herb*, *fine*, *tannin*, *cellar*, *dusti*, *cherri cinnamon dri vanilla*, *ag*, *aroma*, *chocol mouth smoki*, *spice structur*) as well show some variations.

5 Conclusion and Future Work

The initial results of the WEBSOM approach to discerning the descriptors that have the potential to transform the free-text wine comments into quantitative data on wine quality appear convincing. These could be used to analyse the correlations between wine quality and climate change effects. More experimental work is needed to select appropriate wine quality descriptors and confirm the correlations between these descriptors and other wine factors, such as short term climate change effects, wine liking scores/ ratings and price. We contend that continued work of a scientific nature such as reflected in this paper, will provide useful insights into the dynamics of the interaction between environment and plant growth such that better informed decisions can be made by the growers, especially at this time when global climate change is providing much angst among both wine producers and consumers alike.

References

1. Jones, G.V., White, M.A., Cooper, O.R., Storchmann, K.-H.: Climate and Wine: Quality Issues in a Warmer World. In: Proceedings of the Vineyard Data Quantification Society's 10th Econometrics Meeting, Dijon, France (2004)
2. Hansen, A., Dale, V.: Biodiversity in US Forests under Global Climate Change. *Ecosystems* 4, 161–163 (2001)
3. Sallis, P.J., Shanmuganathan, S., Pavesi, L., Jarur, M.: A system architecture for collaborative environmental modelling research. In: Samari, W.W., McQuay, W. (eds.) The 2008 International Symposium on Collaborative Technologies and Systems (CTS 2008), Irvine, California, pp. 39–47. IEEE, New Jersey (2008)
4. Shanmuganathan, S., Sallis, P.J., Pavesi, L., Jarur, M.: Computational intelligence and geo-informatics in viticulture. In: Al-Dabass, D., Turner, S., Tan, G., Abraham, A. (eds.) Proceedings of the Second Asia International Conference on Modelling & Simulation, Kuala Lumpur, Malaysia, pp. 480–485. IEEE computer society, Los Alamitos (2008) (CD version)
5. Wine enthusiast magazine,
<http://www.winemag.com/buyingguide/search.asp?db=>
6. Lagus, K., Honkela, T., Kaski, S., Kohonen, T.: WEBSOM - A Status Report. In: Alander, J., Honkela, T., Jakobsson, M. (eds.) Proceedings of STeP 1996, pp. 73–78. Publications of the Finnish Artificial Intelligence Society (1996)
7. Atkins, T.A., Morgan, E.R.: Modelling the effects of possible climate change scenarios on the phenology of New Zealand fruit crops ISHS Acta Horticulturae 276. In: II International Symposium on Computer Modelling in Fruit Research and Orchard Management,
8. Webb, L.B.: The impact of projected greenhouse gas-induced climate change on the Australian wine industry, PhD thesis, School of Agriculture and Food Systems, University of Melbourne, p. 277 (2006)
9. Gutierrez, A.P., Luigi, P., Ellis, C.K., d'Oultremont, T.: Analysis of climate effects on agricultural systems. Report published by California Climate Change Center CEC-500-2005-188-SD, pp. 28 + appendices A1-7 (2005)
10. Parr, W.V., Green, J.A., White, K.G., Sherlock, R.R.: The distinctive flavour of New Zealand Sauvignon blanc: Sensory characterisation by wine professionals. *Food Quality and Preference* 18, 849–861 (2007)

11. Kontkanen, D., Reynolds, A.G., Cliff, M.A., King, M.: Canadian terroir: sensory characterization of Bordeaux-style red wine varieties in the Niagara Peninsula. *Food Research International* 38, 417–425 (2005)
12. Vannier, A., Bruna, O.X., Feinberg, M.H.: *Food Quality and Preference*, vol. 10, pp. 101–107 (1999)
13. Gawel, R., Iland, P.G., Francis, I.L.: Characterizing the astringency of red wine: a case study. *Food Quality and Preference* 12, 83–94 (2001)
14. Jones, G.V., White, M.A., Cooper, O.R., Storchmann, K.: *Climate and Global Wine Quality*. Climatic Change by Springer 73, 319–343 (2005)
15. Nemani, R.R., White, M.A., Cayan, D.R., Jones, G.V., Running, S.W., Coughlan, J.C.: Asymmetric climatic warming improves California vintages. *Clim. Res.* 19, 25–34 (2001)
16. Ashenfelter, O., Jones, G.V.: The demand for expert opinion: BordeauxWine. In: VDQSAnnual Meeting, d'Ajaccio, Corsica, France, October 1998. Cahiers Scientifique from the Observatoire des Conjonctures Vinicoles Europeenes, Faculte des Sciences Economiques, Espace Richter, Ave. de La Mer, BP 9606, 34054 Montpellier Cedex 1, France (2000)
17. Brochet, F., Dubourdieu, D.: Wine Descriptive Language Supports Cognitive Specificity of Chemical Senses. *Brain and Language* 77, 187–196 (2001)
18. Valentin, D.: Wine language and expertise level: A cognitive point of view. In: Proceeding of Bacchus at Brock, the third international interdisciplinary wine conference, St. Catharine, June 7-9, 11 p. (2007)
19. Bećue-Bertaut, M., Álvarez-Estebar, R., Pageś, J.: *Food Quality and Preference* 19, 122–134 (2008)
20. Frøst, M.B., Noble, A.: Preliminary study of the effect of knowledge and sensory expertise on liking for red wines. *American Journal of Enology and Viticulture* 53(4), 275–284 (2002)
21. Lagus, K., Honkela, T., Kaski, S., Kohonen, T.: In: Alander, J., Honkela, T., Jakobsson, M. (eds.) *Proceedings of STeP 1996*, pp. 73–78. Publications of the Finnish Artificial Intelligence Society (1996)
22. Kaski, S., Honkela, T., Lagus, K., Kohonen, T.: Creating an order in digital libraries with self-organizing maps. In: *Proceedings of World Congress on Neural Networks, WCNN-1996* (1996)
23. Honkela, T., Kaski, S., Lagus, K., Kohonen, T.: Newsgroup exploration with WEBSOM method and browsing interface. Technical Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo. WEBSOM home page (1996), <http://websom.hut.fi/websom/>
24. Lagus, K., Kaski, S., Honkela, T., Kohonen, T.: Browsing digital libraries with the aid of self-organizing maps. In: Hopgood, B. (ed.) *Proc. of Fifth International World Wide Web Conference, Paris*, vol. posters, pp. 71–79 (1996)
25. Lagus, K., Honkela, T., Kaski, S., Kohonen, T.: Self-organizing maps of document collections: A new approach to interactive exploration. In: *Proceeding of Knowledge Discovery and Data Mining, KDD 1996* (1996)

Part XI

**Special Session: Pattern Recognition
from Real-World Information by SVM
and Other Sophisticated Techniques**

A Support Vector Machine with Forgetting Factor and Its Statistical Properties

Hiroyuki Funaya¹, Yoshihiko Nomura², and Kazushi Ikeda¹

¹ Graduate School of Information Science, Nara Institute of Science and Technology,
Ikoma 630-0192 Japan

hiroyuki-fn@is.naist.jp, kazushi@is.naist.jp

² Nippon Steel Corporation, Himeji 671-1188 Japan
nomura.yoshihiko@nsc.co.jp

Abstract. In order to make a support vector machine applicable to time-varying problems, a forgetting factor is introduced to its cost function, in the same way as the RLS algorithm for adaptive filters. The idea of the forgetting factor is simple but it is shown to drastically change the performance of SVMs by deriving the average generalization error in a simple case where input space is one-dimensional. The average generalization error does not converge to zero, differently from the SVM in batch or online. We confirmed our results by computer simulations.

1 Introduction

A support vector machine (SVM) is one of the most successful learning classifiers in decades [1,2,3,4]. Since an SVM works in batch mode, that is, treats all the examples equally, it cannot apply to time-varying environments in the original formulation.

One method to cope with the difficulty is an online approach, where some of the training examples are selected [5] which is rather heuristic, and another is to introduce the forgetting factor [6] like the RLS algorithms for adaptive filters [7]. Although the idea of forgetting factor is simple and reasonable, the cost function introduced in [6] was just the sum of squares followed by the Tikhonov regularization, where the inequalities for margin maximization were replaced with the corresponding equations. Hence, the proposed algorithm has lost the good properties of SVMs such as sparse solutions.

In this study, we introduce the forgetting factor to SVMs in a more natural way. In [2], we formulate the SVM with forgetting factor (SVM-FF) concretely. This algorithm is based on the ν -SVM [8] and has a good geometrical property as shown in [3]. We utilize the property to analyze the generalization performance of the algorithm in the asymptotic of the example set being large in [4], following the validation by computer simulations in [5]. Finally, we give some conclusions and discussions in [6].

2 SVM with Forgetting Factor

Although an SVM is originally formulated as a learning machine which maps input vectors to the corresponding feature vectors and classifies them with the optimal hyperplane in terms of margin [9], we employ the linear kernel which has the feature space identical to the input space and the ν -SVM with homogeneous hyperplanes which has a little different criterion from the margin-maximization. These make the analysis of the algorithm possible as was done in [10] for analyzing the soft-margin effect.

Given a set of N examples (\mathbf{x}_i, y_i) , $i = 1, 2, \dots, N$, where \mathbf{x}_i is an input and y_i is the corresponding label, the ν -SVM with homogeneous hyperplanes is formulated as

$$\min_{\mathbf{w}, \xi_i, \beta} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \beta \quad \text{s.t. } \mathbf{w}' \mathbf{f}_i \geq \beta - \xi_i, \quad \xi_i \geq 0, \quad (1)$$

where \mathbf{w} is the normal vector of a hyperplane $\mathbf{w}^T \mathbf{x} = 0$, ξ_i 's are slack variables for soft margins, C is a constant determining the softness and $\mathbf{f}_i \equiv \mathbf{x}_i y_i$ [8, 10].

In the RLS algorithm for adaptive filters, the squared error at each time decays exponentially, that is, the sum of the squared errors at time N is defined as

$$\sum_{i=0}^N \lambda^{N-i} e_i^2 \quad (2)$$

where e_i denotes the error at time i and $0 < \lambda < 1$ is a constant called the forgetting factor. We introduce this idea to the ν -SVM and give an exponentially decaying weight to the slack variables. Then, the SVM with forgetting factor (SVM-FF) is formulated as

$$\min_{\mathbf{w}, \xi_i, \beta} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \lambda^{N-i} \xi_i - \beta \quad \text{s.t. } \mathbf{w}' \mathbf{f}_i \geq \beta - \xi_i, \quad \xi_i \geq 0. \quad (3)$$

The dual problem of (3) is easily derived as

$$\min_{\alpha_i} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{s.t. } \mathbf{w} = \sum_{i=1}^N \alpha_i \mathbf{f}_i, \quad 0 \leq \alpha_i \leq C \lambda^{N-i}, \quad \sum_{i=1}^N \alpha_i = 1, \quad (4)$$

where α_i 's are the Lagrange multipliers.

3 Geometry of SVM-FF

The cost function to be minimized in (4) is the norm of the weight vector \mathbf{w} . Hence, let us consider the constraints \mathbf{w} should satisfy. The minimum convex set that contains all the given points is called the convex hull of the points. The conditions that any α_i is nonnegative and that their sum is unity lead

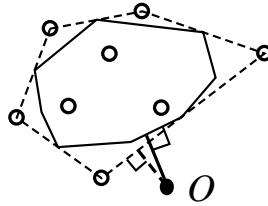


Fig. 1. Points nearest the origin O of the convex hull (the dashed line) and the reduced convex hull (the solid line, $C = 1/2$) of examples shown by o's

to the fact that the weight vector \mathbf{w} is included in the convex hull of $\{\mathbf{f}_i\}_{i=1}^N$. Additional condition that each α_i has an upper bound $C\lambda^{N-n}$ reduces the area where \mathbf{w} can exist. We call the area determined by (4) the reduced convex hull.

Bennett and Bredensteiner [11] firstly pointed out the relationship between the ν -SVM solution and the reduced convex hull (RCH), where two RCHs appeared since they considered inhomogeneous separating hyperplanes. Ikeda and Aoishi [10] added the assumption of homogeneous hyperplanes and showed that the ν -SVM solution is the point nearest the origin in the RCH. Thanks to this simple geometrical picture, the effect of the soft margin parameter was quantitatively elucidated, where the upper bound of any α_i is C (Fig. 1).

In the case of SVM-FF, each α_i has a distinct upper bound $C\lambda^{N-i}$. However, the method for deriving the average generalization error in [10] can also be applied to this case as seen in the next section.

4 Statistical Properties

The problem to be analyzed here is almost the same as that in [10]. In short, we consider unit input vectors in a plane and a noiseless case. Let us fix the true weight vector $\mathbf{w}^* = (0, 1)^T \in S^1$ and assume that N inputs, \mathbf{x}_i , $i = 1, \dots, N$, are independently uniformly chosen from S^1 . Then, the vectors $\mathbf{f}_i \equiv y_i \mathbf{x}_i$, $i = 1, \dots, N$, are uniformly distributed in S_+^1 , where $y_i \equiv \text{sgn}(\mathbf{w}^{*T} \mathbf{x}_i)$ and

$$S_+^1 \equiv \{\mathbf{f} | \mathbf{w}^{*T} \mathbf{f} \geq 0, \mathbf{f} \in S^1\}. \quad (5)$$

In this case, the SVM solution $\hat{\mathbf{w}}$ is the middle point of two endpoints, where each endpoint is the weighted centroid of the example nearest an edge, the one second nearest, and so on (Fig. 2).

When the number of example is large and the examples are dense enough, one endpoint is approximately on S_+^1 and we can discuss its statistical property by its angle denoted by θ_L and the angles of the examples consisting of the endpoint. Note that since the distributions of θ_L and θ_R gives that of the SVM solution $\hat{\mathbf{w}} = (\theta_L + \theta_R)/2$, where θ_R is the angle of the other endpoint, the generalization error defined as the prediction error for a test input is expressed as $|\theta_L - \theta_R|/(2\pi)$.

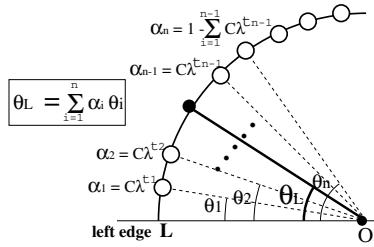


Fig. 2. The endpoint (the black circle) is the weighted centroid of examples (o's), where each weight is determined by the upper bound of each example

For convenience, we sort the examples so that the example i th nearest the edge has the angle θ_i and the upper bound $C\lambda^{t_i}$ as shown in Fig. 2. Then, θ_L of the endpoint is expressed as

$$\theta_L = \sum_{i=1}^n C\lambda^{t_i} \theta_i + \left(1 - \sum_{i=1}^n C\lambda^{t_i}\right) \theta_n. \quad (6)$$

In the sequel we call the n th example f_n the critical example since it is the example on the border the examples below which fully contribute to the SVM solution while the rest nothing.

Now, we evaluate the mean and the variance of θ_L . One important property in (6) is that the angle θ_i is statistically independent of the corresponding forgetting factor, λ^{t_i} , as well as the others, $\lambda^{t'_i}$, $i' \neq i$. Hence, the mean $E[\theta_L]$ and the variance $V[\theta_L]$ are expressed as

$$E[\theta_L] = E[X] E\left[\frac{1}{n-1} \sum_{i=1}^{n-1} \theta_i\right] + E[Y] E[\theta_n], \quad (7)$$

$$\begin{aligned} V[\theta_L] &= E[X^2] E\left[\left(\frac{1}{n-1} \sum_{i=1}^{n-1} \theta_i\right)^2\right] - E[X]^2 E\left[\frac{1}{n-1} \sum_{i=1}^{n-1} \theta_i\right]^2 \\ &\quad + E[Y^2] E[\theta_n^2] - E[Y]^2 E[\theta_n]^2, \end{aligned} \quad (8)$$

where

$$X \equiv \sum_{i=1}^n C\lambda^{t_i}, \quad Y \equiv \left(1 - \sum_{i=1}^n C\lambda^{t_i}\right). \quad (9)$$

We evaluate the expectations above one by one.

The main idea to discuss the distribution of X is that the sum of the weights, $C\lambda^1, \dots, C\lambda^{n-1}$, exceeds unity when the n th, $C\lambda^n$ is added. This means that X can take a value between $1 - C\lambda^n$ and 1 and X is assumed to be uniformly distributed in the interval. Moreover, the probability that $C\lambda^n$ causes the excess is proportional to itself. Under these assumptions, we can write the density of X in the case that the k th example is critical, as

$$p_k(x) = \frac{1}{C\lambda^k} \quad \text{if } 1 - C\lambda^k \leq x \leq 1 \quad (10)$$

and $P_k(x) = 0$ otherwise. Hence the marginal density function of X , $E[X]$ and $E[X^2]$ are respectively derived as

$$p(x) = \sum_{i=1}^N \frac{1-\lambda}{C} C\lambda^i p_k(x), \quad (11)$$

$$E[X] = 1 - \frac{C}{2(1+\lambda)}, \quad (12)$$

$$E[X^2] = 1 - \frac{C}{2(1+\lambda)} + \frac{C^2}{3(1+\lambda+\lambda^2)}. \quad (13)$$

The above discussion for X is directly applied to the distribution of Y and we can derive $E[Y]$ and $E[Y^2]$ as...

$$E[Y] = \frac{C}{2(1+\lambda)}, \quad (14)$$

$$E[Y^2] = \frac{C^2}{3(1+\lambda+\lambda^2)}. \quad (15)$$

As for the distributions of θ_i , $i = 1, \dots, n$, we can utilize the results in [10], where $\theta_{i+1} - \theta_i$ for each i is shown to obey the identical independent distribution, letting $\theta_0 = 0$. In fact, the distribution is asymptotically the same as the exponential distribution with parameter $1/N$. Hence, the expectation of θ_i and θ_i^2 are expressed as

$$E[\theta_i] = \frac{i}{N}, \quad E[\theta_i^2] = \frac{i^2+i}{N^2}, \quad (16)$$

which leads to

$$E\left[\frac{1}{n-1} \sum_{i=1}^{n-1} \theta_i\right] = \frac{n}{2N}, \quad E\left[\left(\frac{1}{n-1} \sum_{i=1}^{n-1} \theta_i\right)^2\right] \approx \frac{n^2}{3N^2}, \quad (17)$$

$$E[\theta_n] = \frac{n}{N}, \quad E[\theta_n^2] = \frac{n^2}{N^2}, \quad (18)$$

in (7) and (8), where \approx ignores the lower orders than n^2 .

As shown above, (17) and (18) depend on n and hence we need to derive the distribution of n . To do it, we introduce an approximation as follows: The time indices, t_i , $i = 1, \dots, N$, independently take one of 0 to $N-1$ with probability $1/N$, although any of 0 to $N-1$ actually appears once in $\{t_i\}_{i=1}^N$. Then, we can show that the density function $p(z)$ of the random variable $Z_i \equiv C\lambda^i$ is expressed as

$$p(z) = \frac{1}{zN \log(1/\lambda)}. \quad (19)$$

We denote the mean and the variance of $p(z)$ by μ_z and σ_z^2 . Then, using the central limit theorem for a large k , we can show

$$S_k \equiv \sum_{i=1}^k Z_i \sim N(k\mu_z, k\sigma_z^2), \quad (20)$$

where $N(\cdot, \cdot)$ denotes the normal distribution. Since the index of the critical example n is determined by $S_{n-1} < 1$ and $S_n \geq 1$, its probability function $p(n)$ is expressed as

$$p(n) = \int_{C\lambda^{N-1}}^1 \frac{1}{\sqrt{2\pi n\sigma_z^2}} \exp\left[-\frac{(x-n\mu_z)^2}{2n\sigma_z^2}\right] \int_{1-x}^1 p(z) dz dx \quad (21)$$

$$= \int_{C\lambda^{N-1}}^1 \frac{1}{\sqrt{2\pi n\sigma_z^2}} \exp\left[-\frac{(x-n\mu_z)^2}{2n\sigma_z^2}\right] \frac{\log(1-x)}{N \log \lambda} dx. \quad (22)$$

Applying (17), (18) and (22) to (7) and (8), the mean and the variance of θ_L are written as

$$\mu_d \equiv E[\theta_L] = \frac{1}{N} \sum_1^N np(n) \left[1 + \frac{1}{2(1+\lambda)} \right], \quad (23)$$

$$\begin{aligned} \sigma_d^2 \equiv V[\theta_L] &= \frac{1}{3N^2} \sum_1^N n^2 p(n) \left[1 - \frac{C}{1+\lambda} + \frac{4C^2}{3(1+\lambda+\lambda^2)} \right] \\ &\quad - \frac{1}{4N^2} \left[\sum_1^N np(n) \right]^2 \left[1 - \frac{C}{1+\lambda} + \frac{5}{4(1+\lambda^2)} \right]. \end{aligned} \quad (24)$$

Then, the average and the variance of the generalization error are expressed as $\sqrt{\sigma_d^2/\pi}$ and $(\pi - 2)\sigma_d^2/(2\pi)$. This means that the average generalization error does not converge to zero but has a floor even when the size of the example set goes to infinity.

5 Computer Simulations

In order to confirm the validity of the theoretical results, some computer simulations were carried out. As shown in the previous section, the generalization error does not converge to null but fluctuates with a finite average and a finite variance, both of which depend on the forgetting factor λ , even when the size N of the example set goes to infinity.

In fact, Fig. 3 shows that the average generalization error of the SVM-FF has a floor, differently from those of the hard-margin SVM or the soft-margin SVM.

Next, we evaluated the average and the variance of the generalization error. In Fig. 4, the theoretical results agree well with the experimental results and the validity of our analysis was confirmed.

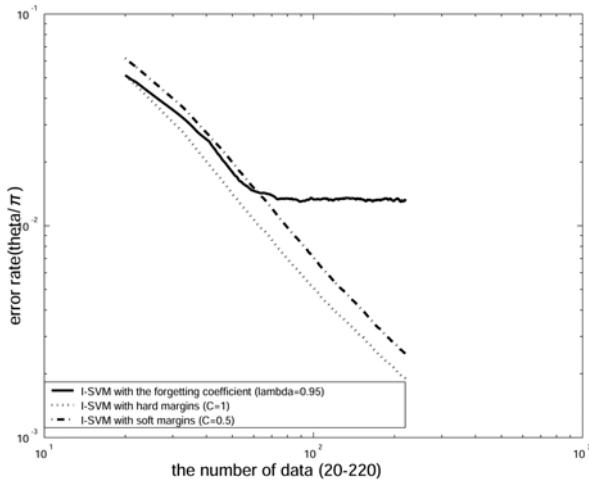


Fig. 3. The average generalization error versus the number of examples. The solid line shows the SVM-FF with $\lambda = .95$, while the dashed and dotted lines show the hard-margin SVM and the soft-margin SVM with $C = .5$.

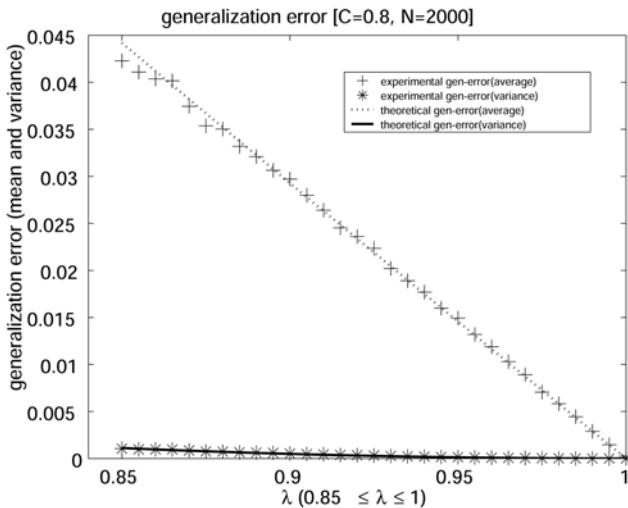


Fig. 4. The average and the variance of the generalization error of 2000 trials, where $N = 2000$ and $C = .8$

6 Conclusions and Discussions

An SVM which has the forgetting factor has been proposed and analyzed. Our asymptotic theory on its generalization error in the simple noiseless case shows that the proposed method has a floor in the average generalization error, differently from the conventional SVMs or the RLS algorithm for adaptive filters, whose errors decrease in the order of $O(1/N)$.

One of the main reasons is that the reduced convex hull does not cover the input space even when the number of examples goes to infinity due to the exponential shrinkage. In other words, important but old examples fade away in this formulation although small number of good examples are critical in SVMs. Similar phenomena will occur in the case of example-selection methods, where some recent examples have large weights while other rather old ones small and the rest zero. We need other novel methods for time-varying problems.

Acknowledgments. This study is supported in part by a Grant-in-Aid for Scientific Research (18300078) from MEXT, Japan.

References

1. Vapnik, V.N.: *The Nature of Statistical Learning Theory*. Springer, New York (1995)
2. Schölkopf, B., Burges, C., Smola, A.J.: *Advances in Kernel Methods: Support Vector Learning*. Cambridge Univ. Press, Cambridge (1998)
3. Cristianini, N., Shawe-Taylor, J.: *An Introduction to Support Vector Machines*. Cambridge Univ. Press, Cambridge (2000)
4. Smola, A.J., Bartlett, P.L., Schölkopf, B., Schuurmans, D. (eds.): *Advances in Large Margin Classifiers*. MIT Press, Cambridge (2000)
5. Klinkenberg, R., Joachims, T.: Detecting concept drift with support vector machines. In: Proc. ICML, pp. 487–494 (2000)
6. Liu, F., Zhang, T., Zhang, R.: Modified kernel RLS-SVM based multiuser detection over multipath channels. *IEICE Trans. Fundamentals* E86-A(8), 1979–1984 (2003)
7. Haykin, S.: *Adaptive Filter Theory*, 3rd edn. Prentice-Hall, Englewood Cliffs (1996)
8. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. *Neural Computation* 12(5), 1207–1245 (2000)
9. Cortes, C., Vapnik, V.: Support vector networks. *Machine Learning* 20, 273–297 (1995)
10. Ikeda, K., Aoishi, T.: An asymptotic statistical analysis of support vector machines with soft margins. *Neural Networks* 18(3), 251–259 (2005)
11. Bennett, K.P., Bredensteiner, E.J.: Duality and geometry in SVM classifiers. In: Proc. ICML, pp. 57–64 (2000)

Improved Parameter Tuning Algorithms for Fuzzy Classifiers

Kazuya Morikawa and Shigeo Abe

Graduate School of Engineering, Kobe University
Rokkodai, Nada, Japan
`076t272t@stu.kobe-u.ac.jp`, `abe@kobe-u.ac.jp`
<http://www2.eedept.kobe-u.ac.jp/~abe>

Abstract. We propose two methods for tuning membership functions of a fuzzy classifier by the support-vector-machine (SVM) like training. For each class, we define a membership function in the feature space. In the first method, we tune the slopes of the membership functions so that the margin between classes is maximized. This method is similar to a linear all-at-once SVM. We call this AAO tuning. In the second method, for each class the membership function is tuned so that the margin between the class and the remaining classes are maximized. This method is similar to a linear one-against-all SVM. This is called OAA tuning. According to the computer experiment, the kernel-discriminant-analysis (KDA) based fuzzy classifiers tuned by AAO tuning and by OAA tuning and SVM show comparable classification performance.

1 Introduction

One of the advantages of fuzzy classifiers is the interpretability of the classifier behavior using fuzzy rules. And extracting fuzzy rules from data is widely accepted as a method for building fuzzy classifiers that are comparable in generalization abilities to, and are advantageous in analyzing system behaviors over, neural networks and various types of fuzzy systems have been developed [1][2][3]. After fuzzy rule extraction, fuzzy rule tuning, i.e., tuning of membership functions, is one of the important steps in realizing fuzzy systems with high generalization ability.

Membership functions can be tuned by multi-layer neural networks [4][5], genetic algorithms [6], reducing the total number of misclassifications allowing correctly classified data to be misclassified [7][8], and support-vector-machine like training [9]. In [9], fuzzy classifiers based on kernel discriminant analysis (KDA) are developed for two-class problems, in which one-dimensional membership functions are defined on the vector that maximally separates two classes in the feature space. Membership functions are tuned by support-vector-machine (SVM) like training to improve generalization ability of the classifier. But this tuning method is restricted to pairwise classification for multi-class problems.

In this paper, we propose two methods for tuning slopes of the membership functions defined for multi-class problems. First, for each class we define a membership function. Then in the first method, the membership functions are tuned

so that the margin between classes is maximized. This is similar to training a linear all-at-once SVM and is called AAO tuning. To reduce computation time for AAO tuning, the second method tunes the membership function for a class so that the margin between the class and the remaining classes is maximized. This is similar to training a linear one-against-all SVM and is called OAA tuning. By computer experiment, we compare classification performance of the two methods.

In Section 2, we discuss fuzzy classifiers defined in the feature space, and in Section 3, we discuss the conventional tuning method and the two proposed tuning methods. In Section 4, we compare the tuning methods for some benchmark data sets.

2 Fuzzy Classifiers

To classify m -dimensional vector \mathbf{x} into one of n classes, we consider generating a fuzzy classifier in the feature space. Assume we have M training data pairs $\{\mathbf{x}_i, y_i\}$ ($i = 1, \dots, M$), where $y_i \in \{1, \dots, n\}$ are class labels. For class i , in the feature space we define a line $\psi_i(\mathbf{x})$, on which a membership function is defined.

We calculate the center of class i data on $\psi_i(\mathbf{x})$, μ_i , by

$$\mu_i = \frac{1}{|X_i|} \sum_{j \in X_i} \psi_i(\mathbf{x}_j) \quad \text{for } i = 1, \dots, n, \quad (1)$$

where X_i is the set of training data indices belonging to class i and $|X_i|$ is the number of elements in X_i .

We define the following fuzzy rule for class i :

$$\text{If } \psi_i(\mathbf{x}) \text{ is } \mu_i \text{ then } \mathbf{x} \text{ belong to class } i. \quad (2)$$

Namely, if $\psi_i(\mathbf{x})$ is closer to μ_i , it is more probable that \mathbf{x} belongs to class i .

To calculate the distance from $\psi_i(\mathbf{x})$ to μ_i , we define the distance $d_i(\mathbf{x})$, where $d_i(\mathbf{x}) = |\psi_i(\mathbf{x}) - \mu_i|$ and based on the distance we define the membership function for class i as follows:

$$m_i(\mathbf{x}) = 1 - \beta_i d_i(\mathbf{x}) - b_i, \quad (3)$$

where β_i is a positive parameter to control the slope of the membership function and b_i is a bias term. In defining membership functions, usually we do not include bias terms, but to improve the classification ability, we include bias terms.

Depending on the value of $\psi_i(\mathbf{x})$, (3) may give negative membership. To avoid this we can use the following equivalent membership function:

$$m_i(\mathbf{x}) = \exp(-\beta_i d_i(\mathbf{x})^2 - b_i). \quad (4)$$

Using the fuzzy rules, \mathbf{x} is classified into the class

$$\arg \max_{i=1, \dots, n} m_i(\mathbf{x}), \quad (5)$$

where \arg returns the subscript that maximizes $m_i(\mathbf{x})$.

In a KDA-based fuzzy classifier defined for two-class problems [9], mapped training data are projected onto vector \mathbf{w} in the feature space, where \mathbf{w} is obtained so that the projected data are maximally separated. In this case, for the two classes, $\psi_i(\mathbf{x})$ ($i = 1, 2$) are given by

$$\psi_i(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}), \quad (6)$$

where $\phi(\mathbf{x})$ is the mapping function that maps input \mathbf{x} into the feature space.

For the kernel fuzzy classifier with ellipsoidal regions [8], $\psi_i(\mathbf{x})$ is given by the kernel Mahalanobis distance and the membership function for class i is given by

$$m_i(\mathbf{x}) = \exp \left(-\beta_i (\phi(\mathbf{x}) - \mathbf{c}_i)^T Q_{\phi_i}^+ (\phi(\mathbf{x}) - \mathbf{c}_i) \right), \quad (7)$$

where \mathbf{c}_i is the center of class i data in the feature space and $Q_{\phi_i}^+$ is the pseudo-inverse of the covariance matrix Q_{ϕ_i} .

3 Tuning Membership Functions

In this section, first we discuss the conventional method for tuning the slope parameter for the KDA-based fuzzy classifier and then proposed to extend the method to multi-class problems.

3.1 Conventional Methods

In [9], to tune β_i for $i = 1, 2$, the difference of the membership functions is defined:

$$\begin{aligned} L(\mathbf{x}) &= m_1(\mathbf{x}) - m_2(\mathbf{x}) \\ &= -\beta_1 d_1(\mathbf{x}) + \beta_2 d_2(\mathbf{x}) + (b_2 - b_1). \end{aligned} \quad (8)$$

Then, defining $\boldsymbol{\beta} = (-\beta_1, \beta_2)^T$, $\mathbf{d} = (d_1(\mathbf{x}), d_2(\mathbf{x}))^T$, and $b = b_2 - b_1$, (8) is rewritten as follows:

$$L(\mathbf{d}) = \boldsymbol{\beta}^T \mathbf{d} + b. \quad (9)$$

Equation (8) is the same as the decision function for the two-dimensional space $(d_1(\mathbf{x}), d_2(\mathbf{x}))$ for a linear SVM. Thus, β_1 , β_2 , and $b = b_1 - b_2$ are determined by training a linear SVM. But by this method, an extension to multi-class problems is limited to pairwise SVMs.

3.2 Proposed Methods

For a tuning method to be applicable to one-against-all classification, instead of (8), we define

$$L_i(\mathbf{x}) = \beta_i d_i(\mathbf{x}) + b_i. \quad (10)$$

From (3), for \mathbf{x} belonging to class i if

$$L_i(\mathbf{x}) < L_j(\mathbf{x}) \quad \text{for } j \neq i, i = 1, \dots, n \quad (11)$$

are satisfied, \mathbf{x} is correctly classified.

Since (10) is a linear function, we can determine the coefficient parameters by training a linear all-at-once SVM [10] called AAO tuning. We consider solving a linear programming support vector machine as follows:

$$\underset{\text{minimize}}{Q(\boldsymbol{\beta}, \mathbf{b}, \boldsymbol{\xi})} = \sum_{i=1}^n \beta_i + C \sum_{j=1}^M \sum_{i \neq y_j, i=1}^n \xi_{ji} \quad (12)$$

$$\begin{aligned} \text{subject to} \quad & \beta_i d_i(\mathbf{x}_j) - \beta_{y_j} d_{y_j}(\mathbf{x}_j) + b_i - b_{y_j} \geq 1 - \xi_{ji} \\ & \beta_i > 0 \quad \text{for } i \neq y_j, i = 1, \dots, n, j = 1, \dots, M, \end{aligned} \quad (13)$$

where C is a margin parameter that determines the trade-off between the classification error and the generalization ability, ξ_{ji} are non-negative slack variables for \mathbf{x}_j and class i , and $y_j \in \{1, \dots, n\}$ is a class label. To solve (12) and (13) by linear programming, we need to change the variables nonnegative. Thus, defining $b_i^+ \geq 0$, $b_i^- \geq 0$, we can express $b_i = b_i^+ - b_i^-$ and the following linear programming program is obtained:

$$\underset{\text{minimize}}{Q(\boldsymbol{\beta}^+, \boldsymbol{\beta}^-, b^+, b^-, \boldsymbol{\xi})} = \sum_{i=1}^n \beta_i + C \sum_{j=1}^M \sum_{i \neq y_j, i=1}^n \xi_{ji} \quad (14)$$

$$\begin{aligned} \text{subject to} \quad & \beta_i d_i(\mathbf{x}_j) - \beta_{y_j} d_{y_j}(\mathbf{x}_j) + b_i^+ - b_i^- - b_{y_j}^+ + b_{y_j}^- \geq 1 - \xi_{ji} \\ & \beta_i > 0 \quad \text{for } i \neq y_j, i = 1, \dots, n, j = 1, \dots, M. \end{aligned} \quad (15)$$

The number of variables for (14) and (15) is $3n + M(n - 1)$ and the number of constraints is $M(n - 1)$.

The optimization problem given by (14) and (15) are inefficient to solve, if the number of training data and/or the number of classes are large.

To solve the optimization problem in such a situation, we consider approximating the all-at-once formulation given by (14) and (15) by one-against-all formulation called OAA tuning. Namely, for class i we solve the following problem:

$$\underset{\text{minimize}}{Q(\beta_i, b_i, \boldsymbol{\xi}_i)} = \beta_i + C \sum_{j=1}^M \xi_{ij} \quad (16)$$

$$\text{subject to} \quad y_{ij} (\beta_i d_i(\mathbf{x}_j) + b_i) \geq 1 - \xi_{ij}, \quad \beta_i > 0 \quad \text{for } j = 1, \dots, M, \quad (17)$$

where $y_{ij} = 1$ if \mathbf{x}_j belongs to class i and -1 otherwise.

Like (14) and (15), we define $b_i = b_i^+ - b_i^-$, where $b_i^+ \geq 0$, $b_i^- \geq 0$. Thus, (16) and (17) become

$$\underset{\text{minimize}}{Q(\beta_i^+, \beta_i^-, b_i^+, b_i^-, \boldsymbol{\xi}_i)} = \beta_i + C \sum_{j=1}^M \xi_{ij} \quad (18)$$

$$\text{subject to} \quad y_{ij} (\beta_i d_{ij}(\mathbf{x}_j) + b_i^+ - b_i^-) \geq 1 - \xi_{ij}, \quad \beta_i > 0 \text{ for } j = 1, \dots, M. \quad (19)$$

We solve the above problem for $i = 1, \dots, n$. Thus, for each class the number of variables is $3 + M$ and the number of constraints is M .

The optimization problem given by (18) and (19) can be solved much faster than that by (14) and (15) but for a large sized problem, we need to speed up tuning using the decomposition technique [11].

4 Computer Experiments

We compared the classification performance of the proposed methods using two-class [12], and multi-class problems [10]. We used a KDA-based fuzzy classifier [9] and tuned the membership functions using the two proposed methods. The fuzzy classifier has a parameter ε to determine the threshold to select independent data and in our study we set $\varepsilon = 10^{-8}$. As a baseline classifier, we used the fuzzy pairwise L1 support vector machine [10]. We used the simplex method to train a linear programming SVM.

4.1 Evaluation Conditions

Table 1 lists the benchmark datasets used in our study. For two class problems, there are 100 or 20 pairs of training and test data sets. Each multi-class problem consists of one training data set and one test data set.

We used RBF kernels and determined the values of the kernel parameter and the margin parameter by fivefold cross-validation. For two-class problems we determined the values using the first five training data sets. The parameter ranges are as follows: $\gamma = \{0.1, 0.5, 1, 5, 10, 15\}$, $C = \{1, 10, 50, 100, 500, 1000, 2000, 3000, 5000, 8000, 10000, 50000, 100000\}$. Table 2 shows the selected parameter values. Four five problems, γ values are different for OAA tuning and AAO tuning.

Table 1. Benchmark data sets

Data	Sets	Inputs	Training	Test	Classes
Banana	100	2	400	4900	2
B. cancer	100	9	200	77	2
Diabetes	100	8	468	300	2
German	100	20	700	300	2
Heart	100	13	170	100	2
Image	20	18	1300	1010	2
Ringnorm	100	20	400	7000	2
F. solar	100	9	666	400	2
Splice	20	60	1000	2175	2
Thyroid	100	5	140	75	2
Titanic	100	3	150	2051	2
Twonorm	100	20	400	7000	2
Waveform	100	21	400	4600	2
Iris	1	4	75	75	3
Numeral	1	12	810	820	10

Table 2. Parameter setting

Data	SVM	OAA Tuning	AAO Tuning		
	C	C	C		
Banana	15	100	10	50	10
B. cancer	0.1	500	0.1	3000	0.1
Diabetes	0.1	3000	0.5	10	0.1
German	0.1	50	0.5	50	0.1
Heart	0.1	50	0.1	50	0.1
Image	15	500	15	1	15
Ringnorm	15	1	0.1	500	0.1
F. solar	0.5	10	15	100	0.1
Splice	10	10	10	1	10
Thyroid	15	100	5	100	1
Titanic	0.5	10	1	10	15
Twonorm	0.5	1	0.5	50	0.5
Waveform	10	1	0.5	50	0.5
Iris	0.1	100	1	10	1
Numeral	0.1	500	0.5	50	0.5
					500

4.2 Results

Table 3 shows the recognition rates of the test data sets. For two-class problems the average recognition rates and their standard deviations are shown. We performed the t-test with the significance level of 0.05 and the best and worst recognition rates are shown in bold and italic, respectively.

From the table, the classification performance of OAA tuning and AAO tuning is comparable with that of SVM. In addition, between the two proposed methods, the classification performance of OAA tuning is comparable with that of AAO tuning. But for the diabetes, flare solar, and titanic problems, the standard deviation by the OAA tuning are worse than those by AAO tuning. Thus, for some problems, AAO tuning is more robust than OAA tuning.

For multi-class problems, the recognition rates by AAO tuning are slightly lower than those by SVM. To check the effect of ε on the recognition rate, we changed the value of ε , fixing the values of γ and C determined for $\varepsilon = 10^{-8}$, and evaluated the recognition rate of the iris test data set. Table 4 shows the results. For $\varepsilon = 10^{-4}$ and 10^{-6} the best recognition rate of 97.33% was obtained. Thus, there is still a room for improving the recognition rate by determining the optimal value of ε by cross-validation.

As discussed in Section 3, AAO tuning requires more computation time and memory than OAA tuning. We compared the computation time of the tuning methods including training and testing the classifier for the parameter values determined by cross-validation. Table 5 shows the computational time for the iris and numeral data sets by AAO tuning and OAA tuning. From the table, the optimization problem by OAA tuning can be solved much faster than that by AAO tuning especially for the numeral data set with 10 classes.

Table 3. Recognition rates of the test data set

Data	SVM	OAA Tuning	AAO Tuning
Banana	89.3±0.52	89.1±0.55	88.0±0.65
B. cancer	72.4±4.67	73.8±4.55	73.5±4.81
Diabetes	76.3±1.83	75.3±3.47	75.8±1.85
German	76.2±2.27	75.3±2.16	75.4±2.41
Heart	83.7±3.41	82.4±3.19	82.3±3.28
Image	97.3±0.41	97.2±0.40	97.2±0.38
Ringnorm	97.8±0.30	97.9±0.27	97.7±0.35
F. solar	67.6±1.74	60.1±5.29	66.3±1.55
Splice	89.2±0.71	89.3±0.61	89.3±0.61
Thyroid	96.1±2.08	95.0±2.30	92.9±3.11
Titanic	77.2±1.12	72.8±3.36	77.1±1.86
Twonorm	97.6±0.14	97.3±0.21	97.3±0.22
Waveform	90.0±0.44	90.3±0.44	90.3±0.41
Iris	97.33	94.67	94.67
Numerical	99.63	99.39	99.51

Table 4. Recognition rates of the iris test data set for different values of ε

ε	Test
10^{-1}	90.67
10^{-2}	90.67
10^{-3}	94.67
10^{-4}	97.33
10^{-5}	94.67
10^{-6}	97.33
10^{-7}	96.00
10^{-8}	94.67

Table 5. Computation time (seconds)

Data	OAA Tuning	AAO Tuning
Iris	0.121	0.188
Numerical	220	2708

5 Conclusions

In this paper we discuss two methods of tuning membership functions. The first method tunes membership functions all at once by a linear SVM. We call this method AAO tuning. And the second method tunes the membership function for a class by a one-against-all SVM. We call this method OAA tuning. According to the computer experiment, classification performance of a kernel discriminant based fuzz classifier by AAO tuning and OAA tuning is comparable to that of the SVM, and OAA tuning is faster than AAO tuning.

References

1. Kuncheva, L.I.: *Fuzzy Classifier Design*. Physica-Verlag (2000)
2. Fullér, R.: *Introduction to Neuro-Fuzzy Systems*. Physica-Verlag (2000)
3. Abe, S.: *Pattern Classification: Neuro-Fuzzy Methods and Their Comparison*. Springer, Heidelberg (2001)

4. Lin, C.-T., Juang, C.-F.: An adaptive neural fuzzy filter and its applications. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics* 27(4), 635–656 (1997)
5. Kim, J., Kasabov, N.: HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. *Neural Networks* 12(9), 1301–1319 (1999)
6. Ishibuchi, H., Nakashima, T., Murata, T.: Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics* 29(5), 601–618 (1999)
7. Abe, S., Thawonmas, R.: A fuzzy classifier with ellipsoidal regions. *IEEE Trans. Fuzzy Systems* 5(3), 358–368 (1997)
8. Kaijeda, K., Abe, S.: KPCA-based training of a kernel fuzzy classifier with ellipsoidal regions. *International Journal of Approximate Reasoning* 37(3), 145–253 (2004)
9. Hosokawa, R., Abe, S.: Fuzzy classifiers based on kernel discriminant analysis. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4669, pp. 180–189. Springer, Heidelberg (2007)
10. Abe, S.: Support Vector Machines for Pattern Classification. Springer, Heidelberg (2005)
11. Torii, Y., Abe, S.: Decomposition techniques for training linear programming support vector machines. *Neurocomputing* (in press)
12. Intelligent Data Analysis Group,

<http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm>

Accelerated Classifier Training Using the PSL Cascading Structure

Teo Susnjak and Andre L.C. Barczak

Massey University, Albany, New Zealand

teo_susnjak@yahoo.com.au, A.L.Barczak@massey.ac.nz

Abstract. This paper addresses the problem of excessively long classifier training times associated with using the Adaboost algorithm within the framework of a cascade of boosted ensembles (CoBE). We present new test results confirming the acceleration of the training phase and the robustness of the **Parallel Strong** classifier within the same **Layer** (PSL) training structure recently proposed by [1]. The findings demonstrate a speed up of an order of magnitude over the current training methods without a compromise in accuracy. We also present a modified version of the PSL training structure that further decreases the duration of the training phase while preserving accuracy.

1 Introduction

The Adaboost machine learning algorithm used in conjunction with a cascading structure to organise classifiers has become a popular approach to designing real-time object detectors. Notable object detection applications have been produced using this technology targeting faces [2,3,4], pedestrians [5], motor vehicles [6] and general surveillance.

However, the problem is that it usually takes an inordinate amount of time to train such classifiers. The training may require weeks [2] or even months [3] to produce an effective classifier. This is seen as one of the major obstacles to the wider usage of this methodology [7] and one of its greatest drawbacks [3]. It is an impediment not only to producing effective real-world object detection applications but also makes the testing of new theories and algorithms as well as verifications of others' research a significant challenge [4].

Since the seminal work of Viola-Jones [2], which introduced a real-time face detector using the cascading structure, there have been a number of proposals put forward that attempted to shorten the training phase while still producing classifiers whose runtime accuracy remained unaffected. [6] focused on shortening the training phase by reducing the number of weak classifiers. They incorporated the output of the previous stage into subsequent stages as an additional stronger and more discriminatory weak learner. Others identified the feature selection component of Adaboost as the bottleneck. [3] used feature filtering, while [8] proposed a forward feature selection (FFS) algorithm in order to achieve notable training time reductions.

Our contribution is twofold. The limited experiments on the PSL structure [1] until now have demonstrated substantial acceleration of the training phase however, the accuracy appeared erratic. We aim to confirm the efficiency and robustness of the PSL training structure through more extensive tests. Secondly, we propose a modification to the PSL structure. The modification produces fewer weak classifiers within the final classification rule. This indicates a faster convergence to a zero training error rate and thus a further reduction in the training time. The improvement comes without a compromise in accuracy.

2 PSL Structure

The PSL structure is shown in Fig. 1b and is contrasted with the standard cascading structure in Fig. 1a. The PSL structure was proposed primarily in order to address the artificial threshold adjustments of the cascaded structure and its inability to maintain a 100% hit rate on the positive training samples. The PSL structure guarantees a 100% hit rate on the positive training samples for each cascade layer without artificial threshold adjustments.

The PSL structure differs from the standard cascading structure during the training phase in that it executes multiple, independent Adaboost rounds (nodes) within the same layer. Whereas the standard cascading structure executes a single Adaboost round until the layer targets have been met, the PSL structure begins a new Adaboost node once a specified criterion is achieved. A criterion may be a maximum number of weak classifiers per node.

Once an Adaboost node reaches the criterion and ends, only the positive training samples that have been misclassified are forwarded to the next node. This process continues from node to node until all the positive training samples have been correctly classified. Meanwhile 100% of the negative training samples are made available to all the nodes. A negative training sample is considered rejected by a layer only if a unanimous decision to reject it has been reached amongst the nodes within that layer. When a layer's target false alarm rate has been reached then the misclassified training negatives become the negative training sample set for the next layer.

In the detection phase, a sample is accepted and immediately passed to the next layer without further computation if it receives a positive vote from any node within a layer. A sample is rejected by a layer as a negative only if a unanimous decision is reached by all the nodes in the same layer.

2.1 Modified PSL Structure

Here we present the modification to the PSL training structure with the aim of realizing further reductions in the training time without compromising accuracy. The modification concerns limiting the size of the negative training sample set passed to each of the nodes during training. In Fig. 2a, we see how the standard PSL approach allows each Adaboost node in a layer to see 100% of the training negatives irrespective of the prior nodes' ability to correctly classify them. We

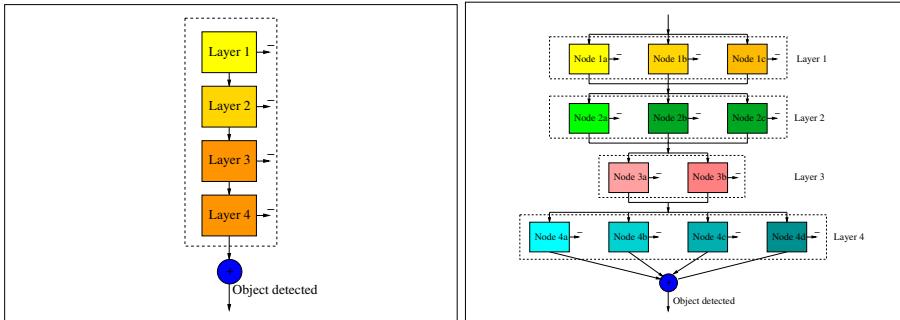


Fig. 1. a) The standard cascade structure. b) the PSL structure [1].

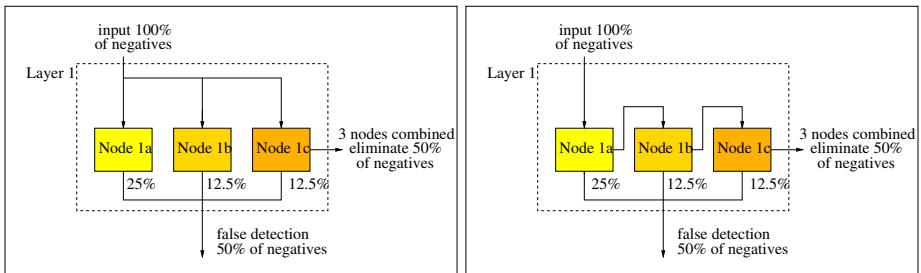


Fig. 2. a) The standard PSL training approach highlighting the usage of the negative training samples. 100% of the negative training samples are made available to each node in the layer irrespective of their prior classification outcome. b) The modified PSL approach showing how only the first node has access to 100% of the negative training samples. Thereafter, only the correctly classified negatives are passed to the next node. Meanwhile, if a negative is misclassified then it is immediately passed on to the next layer.

propose that only the correctly classified negative training samples from previous nodes are passed to subsequent nodes as shown in Fig. 2b.

Recall, that during the training phase only a unanimous decision by all the PSL nodes will result in the layer rejecting a sample as a negative. Therefore, if we include the negative training samples which have already been misclassified by previous nodes into subsequent nodes, we make it harder for Adaboost to concentrate only on rejecting the negative training samples that still stand a chance of being rejected outright by the entire layer.

In summary, the modification should simplify the training by: 1) making the training set smaller for each subsequent node in the layer, 2) allowing Adaboost to focus only on the negative training samples which still have the possibility of being rejected by a layer, thus accelerating the convergence of the layers' target false alarm rate, 3) creating fewer weak classifiers since more control has been gained over which samples are rejected by each node.

3 Method

In order to maintain continuity with experiments conducted on the PSL structure in our previous work [1], we implemented a simple version of Adaboost to cover three classifier structures as well as the modified PSL. The first classifier was trained using the standard, single layered, monolithic Adaboost (in the form of [9]) with a limit of 10000 weak classifiers making up the final strong classifier. The second was trained using standard cascaded Adaboost (in the form of [2]) though it was limited to 50 weak classifiers per layer. The last two were trained using the PSL structure and its variant in which each node within a layer was limited to 15 weak classifiers.

We tested the above structures on the UCI Letters dataset comprising 20000 samples with 16 features each. We used 80% of the samples for training and the remaining 20% for testing. We limited the scope to using only binary classifiers therefore the classifiers were trained in the mode one-against-all for every class with the weak classifier being a simple threshold (a stump classifier). The target false detection rates were set to 0.5 per layer for both the cascaded and PSL structures.

4 Results

We will begin by examining the comparisons between the three main structures and then we will compare the PSL and the modified PSL structures. The analysis of the results will focus on three areas. The first will compare the training phases while the second and third will examine accuracy and the runtime performance respectively.

All PSL classifiers converged to zero training error as did the cascaded except one. As expected, a third of the monolithic classifiers failed to do likewise. The final classifiers were then optimized in order to fairly compare the accuracy. This was done by creating receiver operating characteristic (ROC) graphs for the cascaded and the PSL classifiers that can be seen in Fig. 3a and 3b. The ROC curve for the monolithic classifier is omitted. The omission here is due to the nature of the training method employed which doesn't generate a standard ROC curve since in our training the weak learner attempts to minimize the total error instead of maximizing the hit rate as is used in the Viola-Jones [2] approach.

Table 1a presents the training runtimes in seconds for each of the classes and training structures. The runtimes for the monolithic classifiers are characterized by exorbitantly lengthy periods. The longest training times were in excess of 15 hours. The cascaded structure performed orders of magnitude faster than the monolithic structure while the PSL performed faster still with runtime gains of an order of magnitude for more difficult sets.

Referring back to the ROC curve Fig. 3a and 3b, we're able to see that the PSL reaches optimal accuracy in fewer layers than the cascaded signifying a faster convergence. The large number of layers found in the standard cascaded

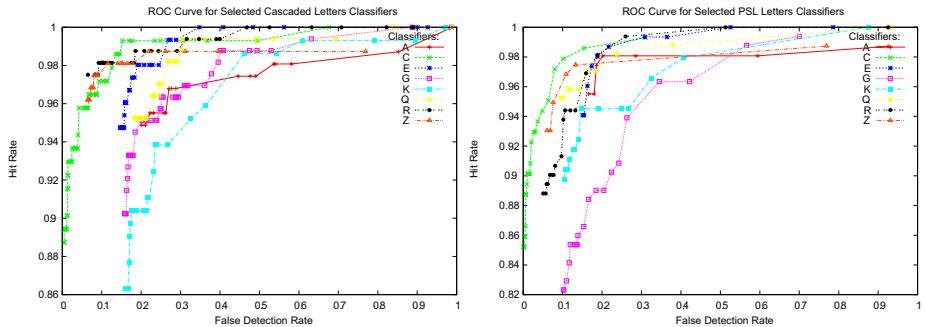


Fig. 3. UCI Letters ROC curves on test samples for selected classifier classes. a) Cascaded classifiers. b) PSL classifiers.

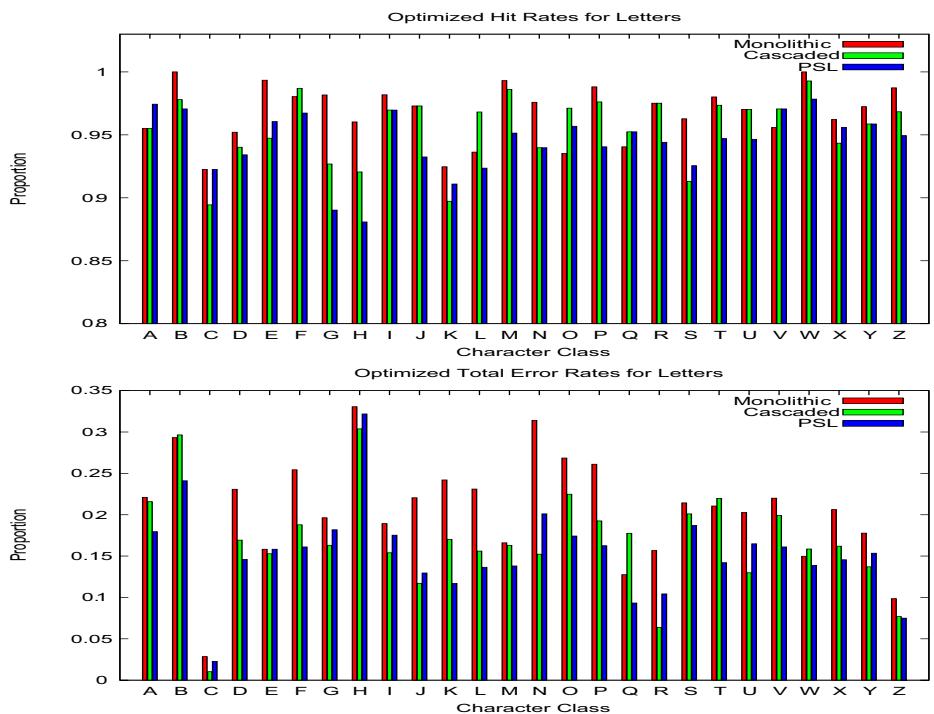


Fig. 4. Accuracy results on the test set for the three structures on the UCI Letters dataset. a) Hit rate. b) Total error rate.

approach is an important observation because it has been shown [10] that the final detection rate decreases as the cascade levels increase. This is due to the fact that during training, each layer will receive a sub-set of the original positive

training sample set from the previous layer ensuring that the detection rate becomes irreversibly compromised at each step.

Figure 4a demonstrates the hit rates for the three structures. The monolithic classifiers produce figures which are in general marginally higher than others. The PSL classifiers perform at a comparable rate to the cascaded classifiers but overall somewhat lower. The diminished hit rates for the PSL structure can in part be justified by the lower false detection rates achieved against both the monolithic and the cascaded structures. While the monolithic classifiers achieved the highest hit rates they likewise exhibited the highest false detection rates. This is further confirmed by the Fig. 4b showing the total error rates. There we see that the highest total error rates are associated with the monolithic structure while the majority of the lowest are attributed to the PSL structure. This confirms that the PSL structure is robust and has the generalization ability that is as good if not better than the structures being considered.

Lastly, we examine the detection runtime of each of the structures in terms of the time required to perform a classification. Table IIb shows the number of weak classifiers that comprise each final strong classifier. The PSL classifiers contain significantly fewer weak classifiers which translates to a runtime advantage. The runtime advantage of the PSL structure over the cascaded is up to 57% in some cases. The cascaded structure has already been shown to enable real-time classification. Thus, the computational efficiency offered by the PSL structure further increases the likelihood of executing difficult classification problems in real-time .

4.1 Results Comparing the Two PSL Structures

The ROC curves for both structures showed no apparent differences indicating that the generalization of the modified PSL structure has remained intact. Results from the hit rates on test samples demonstrated a slightly better performance to that of the standard PSL but this was counter balanced by lower false detection and overall total error rates of the standard PSL structure. This is an indication that both structures were operating at slightly different points on the ROC curve. Nonetheless, the data points to the viability of the modified structure.

The training times are inconclusive in showing that the modified PSL is significantly faster due to the training runtimes for the PSL structure being very low in the first place. A hint of a potential acceleration can be seen when examining the class 'C' which was the most difficult to train for all structures resulting in longest training times. In training this sample, the modified PSL structure achieved a speed up of 33% whilst preserving accuracy. More interestingly, it needed 57% fewer features to achieve the same classification. The reduction of features has been pointed out to reduce training times linearly [6]. The last two columns of table IIb point to a general reduction in the number of features required to train classifiers using the modified PSL structure. Out of 26 classifiers only 3 failed to realise this reduction. The reduction of features making up a classifier will also results in faster runtimes.

Table 1. a) Training times (seconds) for the four structures for each of the UCI Letters character classes. b) Number of features required to train classifiers for each of the UCI Letters character classes in the four training structures.

Class	Mono.	Casc.	PSL	Mod. PSL	Class	Mono.	Casc.	PSL	Mod. PSL
A	223	2	6	6	A	494	136	155	145
B	345	7	5	5	B	677	481	201	159
C	53572	844	18	12	C	10000	2378	1298	738
D	51927	61	8	8	D	10000	966	453	373
E	54952	76	7	6	E	10000	906	351	299
F	5924	25	5	5	F	3460	698	235	208
G	52813	197	8	7	G	10000	1391	787	539
H	54384	98	9	8	H	10000	911	683	471
I	792	4	5	5	I	1110	273	201	204
J	1406	6	6	5	J	1579	449	310	236
K	53774	115	7	7	K	10000	1192	519	353
L	53947	48	6	6	L	10000	780	260	275
M	431	3	5	5	M	763	202	186	123
N	1344	16	6	6	N	1506	669	279	244
O	52480	59	7	7	O	10000	922	362	307
P	12804	40	6	5	P	5165	806	259	205
Q	53633	92	6	6	Q	10000	987	330	214
R	52547	148	8	7	R	10000	1247	752	487
S	53347	63	7	7	S	10000	924	546	425
T	21126	35	7	6	T	6813	740	264	218
U	6972	18	6	6	U	3778	538	354	302
V	53366	11	6	5	V	10000	556	237	244
W	228	3	5	5	W	510	228	142	140
X	4002	22	6	6	X	2817	550	285	262
Y	1303	14	5	5	Y	1491	576	221	205
Z	731	5	1	1	Z	1072	371	129	140

5 Conclusion and Future Work

The research confirms that the PSL structure is robust in its ability to rapidly create binary classification rules that generalize as well as the best available standard training structures but with a substantial reduction in training times. The monolithic and the cascaded training structures produced classifiers for the UCI Letters dataset which had higher hit rates than the PSL but this was counter balanced by the the lower false detection rates and overall lower error rates of the PSL structure.

The PSL structure achieves zero training errors very easily. It is at least an order of magnitude faster to train than the cascaded structure for sample sets that exhibited a higher training difficulty. It is precisely these more difficult training sample sets that are of interest to us since this is the category into which real-world classification problems fall into.

The detection phase is faster to run due to the smaller number of weak classifiers. Cascade thresholds no longer need to be adjusted while a 100% positive training set hit rate is guaranteed.

The newly presented modified PSL structure trains classifiers that require fewer weak classifiers without a degradation in accuracy. The outcome of this is faster training and runtime phases. The UCI Letters dataset used here for training was amenable to very rapid training times by the standard PSL structure thus, in order to witness a more accurate distinction between its training times and those of the modified structure, a larger and more complex dataset will need to be tested.

Future research will require experiments on both PSL structures with larger datasets. Lastly, it is also the intention of the authors to explore the PSL structure within a multi-class Adaboost framework.

References

1. Barczak, A.L.C., Johnson, M.J., Messom, C.H.: Empirical evaluation of a new structure for adaboost. In: SAC 2008: Proceedings of the, ACM symposium on Applied computing, Fortaleza, Ceara, Brazil, pp. 1764–1765. ACM, New York (2008)
2. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57, 137–154 (2004)
3. Verschae, R., del Solar, J.R., Correa, M.: A unified learning framework for object detection and classification using nested cascades of boosted classifiers. Mach. Vision Appl. 19(2), 85–103 (2008)
4. McCane, B., Novins, K.: On training cascade face detectors. In: Image and Vision Computing, New Zealand, Palmerston North, pp. 239–244 (2003)
5. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: ICCV, pp. 734–741 (2003)
6. Withopf, D., Jahne, B.: Improved training algorithm for tree-like classifiers and its application to vehicle detection. In: Jahne, B. (ed.) Proc. IEEE Intelligent Transportation Systems Conference ITSC 2007, pp. 642–647 (2007)
7. Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Towards optimal training of cascaded detectors. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 325–337. Springer, Heidelberg (2006)
8. Wu, J., Rehg, J.M., Mullin, M.D.: Learning a rare event detection cascade by direct feature selection. In: NIPS (Advances in Neural Information Processing Systems), Vancouver, Canada (2003)
9. Freund, Y., Schapire, R.E.: A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence 14(5), 771–780 (1999)
10. Zhang, C., Viola, P.: Multiple-instance pruning for learning efficient cascade detectors. In: NIPS 2007 (December 2007)

Imitation Learning from Unsegmented Human Motion Using Switching Autoregressive Model and Singular Vector Decomposition

Tadahiro Taniguchi^{1,2} and Naoto Iwahashi^{2,3}

¹ Ritsumeikan University, College of Information Science and Engineering
Shiga 525-8577, Japan
taniguchi@ci.ritsumei.ac.jp

² National Institute of Information and Communications Technology
Kyoto 619-0288, Japan
naoto.iwahashi@nict.go.jp

³ Advanced Telecommunications Research Institute International
Kyoto 619-0288, Japan

Abstract. This paper presents a method, which enables a robot to extract demonstrator's key motions based on imitation learning through unsegmented human motion. When a robot learns another's motions from unsegmented time series, the robot has to find what he learns from the continuous motion. The learning architecture is developed mainly based on a switching autoregressive model (SARM), a simple phrase extraction method, and singular vector decomposition to discriminate key motions. In most previous research on methods of imitation learning by autonomous robots, target motions that were given to robots were segmented into several meaningful parts by the experimenters in advance. However, to imitate certain behaviors from the continuous motion of a person, the robot needs to find segments that should be learned. In our approach, the learning architecture converts the continuous time series into a discrete time series of letters by using SARM, finds candidates of key motions by using a simple phrase extractor which utilizes n-gram statistics, and removes meaningless segments from the keywords by utilizing singular vector decomposition (SVD) to achieve this goal. In our experiment, a demonstrator displayed several unsegmented motions to a robot. The results revealed that the framework enabled the robot to obtain several prepared key motions.

1 Introduction

Imitation learning is one of the most important learning capabilities human beings have. Imitation learning enables people to acquire a variety of motions by observing another's behaviors. This reduces teaching cost to make a learner obtain new skills. In imitation learning, people acquire not only the motion or skill itself but also extract what they should learn by themselves by observing another's action or interacting with another. Imitation learning is a key capability for obtaining social skills. Therefore, if we want to develop autonomous

robots that can live together with us, we should embed such a capability on them. If a humanoid robot, which can learn different behaviors incrementally and autonomously without any explicit commands or signals, is developed, it will have a great impact on our society. The idea is called “emergence of behavior” [4]. Tani et al. proposed a recurrent neural network with parametric bias (RNNPB), which enables a robot to acquire several behaviors from a user’s continuous motion [6]. Although RNNPB is a single recurrent neural network, it can be used to obtain several motion patterns because it has several parametric biases that the RNN’s parameter space. This is an example of an approach toward emergence of behavior with a nonparametric model. In contrast, Taniguchi et al. proposed a learning framework, which combines a switching Autoregression model and a keyword extraction method [8]. This approach is based on a parametric model. In the previous research, they applied their proposed model to a robot, and have a robot imitate human right hand motion whose dimension of state space is only three. Scalability of the method has not been evaluated. Additionally, the method assumes that human motion has the same intrinsic structure as written natural language. It is difficult to assure that the human motion satisfies the assumption. Therefore, we modify the model and describe a new model in which we focus on the low dimensionality of human key motions.

2 Algorithm

In this section, we introduce a learning algorithm, which is used to segment human motion data, acquire a dynamical model, and extract key motions. We call human motions that are distinguishable and become targets of imitation learning as gkey motionsh. To extract such a discriminative region from unsegmented time series obtained from observed human motion is difficult for an autonomous robot. Taniguchi et al. proposed a learning architecture, which consists of a Switching AR model and a keyword extraction method. However, their method does not work if several conditions are not satisfied¹. To overcome such a problem, we assume another condition derived from “motor coordination” instead of the conditions about distribution of n-grams [8]. Our proposed learning architecture mainly consists of three learning modules: the switching autoregressive model (SARM), a simple phrase extraction method, and a method for detecting motor coordination using singular vector decomposition (SVD). An overview of the proposed model is shown in Figure II.

2.1 Switching Autoregression Model

SARM models a multidimensional time series using multiple AR models whose hidden states are switched from one to another based on a Markov process [4].

¹ For example, they assumed that n-grams, which are converted from the key motions of the time series from a participant, are distributed over the set of time series similar to keywords of documents written in natural languages. However, whether this assumption is satisfied or not is uncertain.

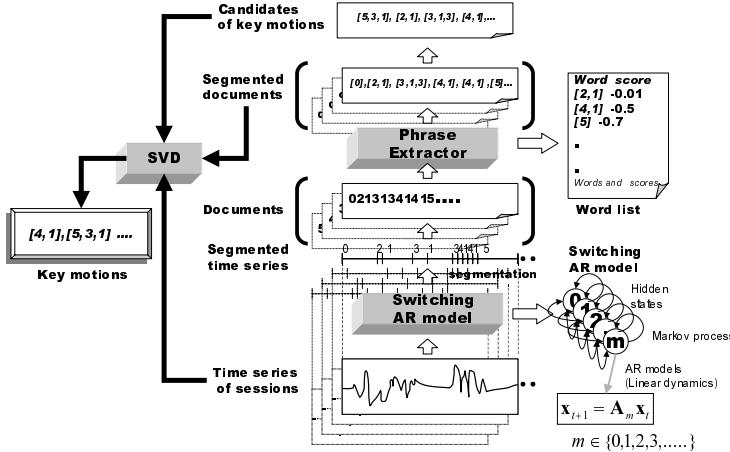


Fig. 1. Overview of the proposed model

A SARM model has M hidden states. We denote the hidden state at t by s_t . A hidden state, j , has an AR model.

$$x_t = A_j x_{t-1} + v_t, \quad (1)$$

where j represents the index of the hidden states, x_t represents a state variable at t , and $v_t \sim N(0, Q_t)$. First, we will explain how to determine which AR model should be chosen when a robot observes a time series, $x_{1:T}$. To achieve that purpose, the probability, $\Pr(s_t = j|x_{1:T})$, should be calculated. Suppose that the hidden state has Markov properties,

$$\Pr(s_t = j|x_t, x_{1:t-1}) \quad (2)$$

$$= \frac{1}{c} \Pr(x_t|s_t = j, x_{1:t-1}) \Pr(s_t = j|x_{1:t-1}) \quad (3)$$

$$= \frac{1}{c} \Pr(x_t|s_t = j, x_{1:t-1}) \sum_i \Pr(s_t = j|s_{t-1} = i, x_{1:t-1}) \Pr(s_{t-1} = i|x_{1:t-1}) \quad (4)$$

$$= \frac{1}{c} L_t(j) \sum_i Z(i, j) \Pr(s_{t-1} = i|x_{1:t-1}), \quad (5)$$

where c is the normalization constant and Z is the transition matrix of hidden states. $Z(i,j)$ represents the probability that the hidden state will transit from i to j in one time step.

$$L_t(j) = N(x_t; A_j x_{t-1}, Q_j) \quad (6)$$

is the likelihood of prediction error at time t given by the AR model, j , where N is a multidimensional normal distribution whose center vector is $A_j x_{t-1}$ and whose variance-covariance matrix is Q_j . On the backward pass, we have

$$\Pr(s_t = j|x_{1:T}) = \sum_k \frac{\Pr(s_t = j|x_{1:t}) \Pr(s_{t+1} = k|s_t = j)}{\Pr(s_{t+1} = k|x_{1:t})} \Pr(s_{t+1} = k|x_{1:T}). \quad (7)$$

The derivation of these equations is almost the same as the derivation of the hidden Markov model (HMM). The parameters, A_j, Q_j , can be estimated using the expectation maximization (EM) algorithm below [4].

$$A_j = \left(\sum_l \sum_{t=2}^T W_t^j P_{t,t-1} \right) \left(\sum_l \sum_{t=2}^T W_t^j P_{t,t-1} \right)^{-1} \quad (8)$$

$$Q_j = \frac{1}{\left(\sum_l \sum_{t=2}^T W_t^j \right)} \left(\sum_l \sum_{t=2}^T W_t^j P_t - A_j \sum_l \sum_{t=2}^T W_t^j P'_{t,t-1} \right) \quad (9)$$

where $W_t^j \equiv \Pr(s_t = j|x_{1:T})$, $P_t \equiv x_t x'_t$, and $P'_t \equiv x_t x'_{t-1}$. In the original paper about a SARM [4], the initial distribution, $P(s_1 = j) = \pi_j$, was also re-estimated. However, we have not re-estimated this parameter for the sake of simplicity. We extract a “document” from the time series of the posterior probabilities.

$$s_t^* = \operatorname{argmax}_j \Pr(s_t = j|x_{1:T}) = \operatorname{argmax}_j W_t^j, \quad (10)$$

where s_t^* is a sequence of the most likely hidden states. By ignoring neighboring identical states, we compress sequences containing the same successive letters into a document that does not contain any repetitive letters. Finally, we obtain “documents” written in letters representing hidden states of SARM. The number of documents is the same as the number of sessions in which a user exhibits unsegmented motion.

2.2 Simple Phrase Extraction Method

We assume that a key motion is represented by a sequence of linear dynamics. By adequately connecting several letters representing hidden states corresponding to linear dynamics, key motions that may have nonlinear dynamics can be represented. Taniguchi et al. used Umemura’s keyword extraction method, which does not use any dictionaries arranged in advance [79]. The method extracts key motions by using only the information about the distribution of n-grams. Therefore, it is overly dependent on the distribution of the letters. On the other hand, people seem to notice another’s key motions by catching characteristic information involved in the motion’s dynamics itself, not in distribution of the key motions. Our proposed method mainly uses an intrinsic feature value the time series data of a key motion itself has. To achieve this, we adopt a simple phrase extraction method, which only involves an n-gram occurrence rate to pick up candidate key motions.

First, the phrase extractor calculates occurrence rates of n-gram $w = [c_1, c_2, \dots, c_n]$ in all target documents, where w represents an n-gram, c_i represents a letter corresponding to SARM’s hidden state. N-grams whose $n < n_{max}$ are taken into consideration. Next, it calculates *score* of each n-gram as follows.

$$\text{Score} = \log(\min(UB, \#(w))/(1 + \#(w))) \quad (11)$$

where $\#(w)$ is the frequency w appears, and UB (upper bound) is a parameter of this method. This score becomes negative, and it approaches 0 when the n-gram occurrence rate is high. UB mainly limits the score of 1-gram, which appears many times. It segments a target document based on the scores. At this time, the segmentation is executed so that the total score of the segmented document is maximized. In the segmentation, the method reduces computational costs by using a Viterbi search. After segmentation, it counts the n-grams, which appear in the segmented document, and recognizes n-grams, which satisfy the following conditions, as key-motion candidates.

- $score > score_{min}$
- A key motion appears at least two times
- A key motion corresponds to a time series whose time period is longer than T_{min}

The third condition means that people must not recognize a too short of a time series as a key motion.

2.3 Detecting Motor Coordination Using Singular Vector Decomposition

The simple phrase extraction method can pick up key motion candidates, but the probability that all the candidates are key motions is low because it uses only the information about the occurrence rate. Therefore, we use other information that discriminates key motions from the key motion candidates. To achieve this we focus on “motor coordination” in organized human movement.

“Motor coordination” is widely known in research fields of human skill. When people acquire a motion and exhibit it repeatedly, bodily motor coordination is organized. The dimensionality of the movement becomes reduced because the muscles work together by intercorrelating with each other. This suggests that the dimensionality of key motions’ movement will be reduced. The proposed method uses this information to discriminate key motions from the candidates. To reduce the dimensionality of a time series, SVD is widely used. SVD is used to calculate the optimal matrix, whose rank is smaller than a target matrix, by approximating the original matrix. By using this SVD feature, we can evaluate the intrinsic dimensionality of a time series. When the target time series is described as $X = [x_1^t, x_2^t, \dots, x_T^t]^t$, its SVD and a feature value $D(X)$ become

$$X = USV^T, \quad D(X) = \sum_{k=1}^N kp_k, \quad p_k = \sigma_k / \sum_{j=1}^N \sigma_j \quad (12)$$

where $S = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_N\}$ and N is the dimension of the state space. σ_i is a singular value ($\sigma_1 > \sigma_2 > \dots > \sigma_N \geq 0$). If the $D(X)$ is small enough, the motion seems to have motor coordination. Therefore, if $D(X) < D_{threshold}$ is satisfied, we put the candidate into a group of key motions.



Fig. 2. Gypsy 5 Torso

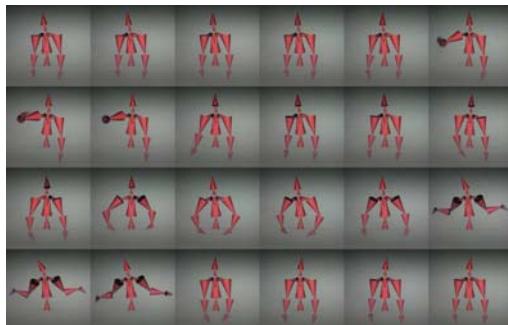


Fig. 3. Unsegmented motion exhibited by a participant

3 Experiment

To evaluate the proposed method, we conducted an experiment. In the experiment a participant exhibited several unsegmented motion trajectories which contains several key motions. The learning system had to extract meaningful motion segments from the human motion only through unsupervised learning process.

3.1 Conditions

In this experiment, we used a participant's entire upper body motion, which includes a DOF of 27², as target time series. To measure upper participants' body motions, we used a motion capture system, Gypsy 5 Torso, produced by Meta motion. We asked a participant to move for 20 seconds, 5 times. We asked the participant to exhibit two types of key motions, i.e., "shrug" and "hi", and recorded the movement. An example of a portion of an exhibited unsegmented human motion is shown in Figure 3.

We had the SARM learn these data. The SARM output 20 documents written in indecies representing the hidden states of SARM after estimating the model parameters and most probable hidden states at each time step. In this experiment

² It included Euler angles of 9 joints, i.e., a head, a neck, a chest, L and R elbows, L and R shoulders, and L and R collars.

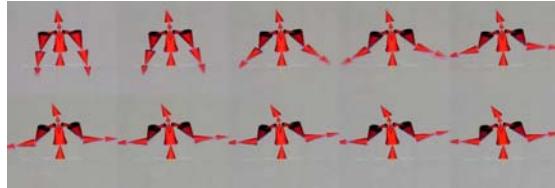


Fig. 4. Acquired “shrug motion” corresponding to [5,2,5,2]

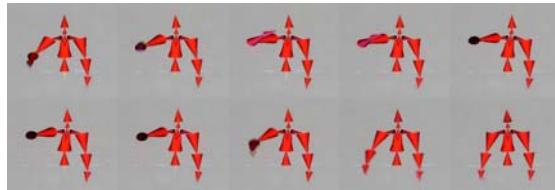


Fig. 5. Acquired “hi motion” corresponding to [0,6,1,6]

the number of linear models was set to 8, the diagonal elements of a transition probability matrix Z was set to 0.98, and the other elements of Z were set equally. The initial parameters of linear models A_i, Q_i were generated randomly.

3.2 Result

Simple phrase extraction method segmented the documents and extracted candidates of key motions. They were [0, 1, 0, 1], [0, 6, 1, 6], [1, 0, 1, 0], [2, 1], [5, 1, 5, 1], and [5, 2, 5, 2]. Parameters for the simple phase extraction method and evaluation criteria based on SVD was set as shown in Table 1. To evaluate the candidates, $A, D(X)$ for each n-gram was calculated by using SVD. The result is shown in Figure 6. This shows that finally two key motions were extracted as key motions. The two motions, corresponding to [5,2,5,2] and [0,6,1,6] respectively, were shown in Figure 4 and Figure 5. They corresponded to the target motions, “shrug” and “hi”. In contrast, candidates whose $D(X)$ is higher than $D_{threshold}$ found looking like meaningless motion to most people. However, the extracted segments are proved to be only a portion of all the exhibited key motions. Table 2 shows the occurrence rate of key motions in the exhibited human movement, and the

Table 1. Parameters of the learning system in the experiment

UB	0.8
$score_{min}$	-0.6
n_{max}	10
$D_{threshold}$	3

Table 2. Occurrence rate of key motions and extracted n-grams

<i>Shrug</i>	11
[5, 2, 5, 2]	2
<i>Hi</i>	13
[0, 6, 1, 6]	4

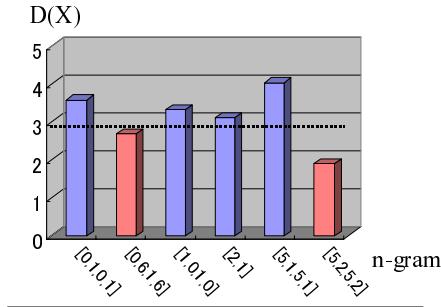


Fig. 6. $D(x)$ for each motions calculated by SVD

occurrence rate of n-grams in the segmented documents. This result shows that the method is not usable to recognize human motion. However, our experiment shows that the method can be used to extract key motions to some extent.

4 Conclusion and Discussion

We described an imitation learning method for extracting key motions from unsegmented human motion in this paper. A motion extraction method, which uses information of human body movement, was proposed and tested by applying it to a high dimensional system with a DOF of 27 using upper-body motion. The method was used to extract two embedded key motions. However, several shortcomings of our approach appeared during the experiment. First, a letter corresponding to a hidden state of SARM represents an AR model. Therefore, if the initial position is different, the motion recognized as the same letter becomes different for a human observer even if the AR model is the same. We should take other segmental models into consideration, e.g., HMM [25] and interval system [3]. To construct reasonable evaluation criteria is also an important problem. The problem we treated in this paper is originally unsupervised learning problem, and the objective answer is not obvious. Speech recognition problems are similar to our problem.

To sophisticate our learning method more theoretically is also required. Especially, a phrase extraction method is now only a heuristic method. Therefore, it includes many parameters, which must be set by a designer carefully before robots start to learn. To invent an alternative phrase extraction method whose parameters can be automatically decided is required. In addition to that, two separate learning processes, i.e., SARM and a keyword extractor, should be integrated based on statistical learning theory. To integrate the two computations as a single optimization problem based on an information theory will give more feasible learning results and farther perspective of imitation learning.

Acknowledgment. This paper was partially supported by a research grant from the National Institute of Informatics, by the Japanese Ministry of Education,

Culture, Sports, Science and Technology, under a Grant-in-Aid for Young Scientists (Start-up) Project No. 20800060 and Grant-in-Aid for Creative Scientific Research No. 19GS0208.

References

1. Asada, M., MacDorman, K.F., Ishiguro, H., Kuniyoshi, Y.: Cognitive developmental robotics as a new paradigm for the design of humanoid robots. *Robotics and Autonomous Systems* 37(2-3), 185–193 (2001)
2. Inamura, T., Toshima, I., Tanie, H., Nakamura, Y.: Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research* 23(4), 363–377 (2004)
3. Kawashima, H., MAatsuyama, T.: Multiphase Learning for an Interval-Based Hybrid Dynamical System (Special Section Concurrent/Hybrid Systems: Theory and Applications). *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 88
4. Murphy, K.P.: Switching Kalman filters. Dept. of Computer Science, University of California, Berkeley, Tech. Rep. (1998)
5. Sugiura, K., Iwahashi, N.: Learning object-manipulation verbs for human-robot communication. In: Workshop on Multimodal Interfaces in Semantic Interaction at the International Conference on Multimodal Interfaces (2007)
6. Tani, J., Ito, M., Sugita, Y.: Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB. *Neural Networks* 17(8-9), 1273–1289 (2004)
7. Taniguchi, T., Iwahashi, N.: Computational model of role reversal imitation through continuous human-robot interaction, pp. 25–31 (2007)
8. Taniguchi, T., Iwahashi, N., Sugiura, K., Sawaragi, T.: Constructive approach to role-reversal imitation through unsegmented interactions. Journal ref: *Journal of Robotics and Mechatronics* 20(4), 567–577 (2008)
9. Umemura, K.: Related word-pairs extraction without dictionaries. Technical report, IPA Exploratory Software Project development result report (2000) (in Japanese),
<http://www.ipa.go.jp/archive/NBP/12nendo/12mito/smdata/10-36h/10-36h.pdf>

Vision Based Mobile Robot for Indoor Environmental Security

Sean W. Gordon¹, Shaoning Pang¹, Ryota Nishioka², Nikola Kasabov¹,
and Takeshi Yamakawa²

¹ Knowledge Engineering and Discovery Institute, Auckland University of Technology, Auckland 1020, New Zealand

² Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 2-4, Hibikino, Wakamatsu, Kitakyushu, 8080196, Japan

Abstract. This paper presents our development on mobile robot in charge of security in an office environment. The developed robot uses curiosity on environmental change for threat detection. The platform used to develop the security robot is the ‘WITH’ mobile robot , imported from Kitakyushu Institute of Technology Japan. Within the environment of a research office, the WITH security robots capability has been tested on threat detection, programmable security navigation and threat tracking. The results achieved from the tests have been highlighted below: (1) Environment curiosity modelling: this allows for more dynamic threat detection than object recognition method, (2) Programmable navigation system: enables multi-points security monitoring.

1 Introduction

Security is one of the important factors in our everyday lives especially when it comes to security of an area such as offices or banks which holds valuable and confidential property. In order to safeguard these valuable commodities a fool proof security system is required. Most current security systems involve stationary security cameras and mobile security officers. These security officers need to patrol the premises regularly to maintain the desired level of security. These officers may in addition need to regularly check specific security points, such as a bank vault, to ensure their security. However the involvement of humans increases the security risk related to human error. The possibility of human error can be minimized by introducing a more advanced security solution such as the use a mobile intelligent agent in place of a security officer.

A mobile security robot can provide the standard of security needed by an office environment. Like the current solution of using human security officers a mobile robot is capable of patrolling in an office environment and checking specified security points without outside assistance. Additionally, using a mobile security robot can reduce the risk of human error and the robot can be equipped with additional sensors such as an omnidirectional camera, which may be difficult for a security officer to equip and use.

In this paper, we develop a mobile security robot to fulfil the role of security officer. This security robot is capable of moving along a patrol path, checking pre-specified security points en route, in a real office environment. The security robots threat detection is based on curiosity modelled on unexpected objects in the environment, particularly unknown faces.

2 Relevant Works

There are two areas of research that are pertinent to this research, the development of security robots and methods of modelling curiosity.

2.1 Review of Security Robots

Previous research has been made on solutions to the problem of threat detection and identification for security robots. One possible solution is to use face recognition, which is used by several security robots. One such robot is the iBotGuard (developed at the Hong Kong Polytechnic University by J. Liu *et al* [2]) which uses an invariant face recognition approach to detect threats. A. Treptow *et al* [3] has developed a robot that uses a thermal camera to locate a face, extracts the located face using a CCD camera and then uses the eigenface method to recognize it.

A security robot developed by Andreasson *et al* [4] uses a different solution. First the robot builds a 3D model of its environment and then it compares subsequent scans to this baseline, searching for any deviations. In a real-world experiment in an indoor office the robot was able to detect several changes in the environment.

2.2 Review of Curiosity Modelling

Within the research literature there are several methods of curiosity modelling that have been reported. One method is to use uncertainty as a measure of curiosity. A robot system developed by E. Uğur *et al* [5] uses this method to drive the robot to learn the transversability affordance of its environment. Also, the DIDO system developed by P. D. Scott and S. Markovitch [6] uses this method of curiosity modelling. Here, the DIDO system uses uncertainty in the result of a given operation focussing on outcomes that are likely while ignoring outcomes that are unlikely. Another method used by P. Oudeyer *et al* [7] is to use predictions and the errors of those predictions as a measure of curiosity. This curiosity is used to drive the robot to learn the outcome of a given action.

Curiosity has also been modelled in simulation using ILDA [8]. In this implementation curiosity is the difference between a new instance and what the robot has learned. The curiosity of each new instance is evaluated - if the curiosity is high the instance is used to update ILDA, otherwise the instance is discarded.

3 Vision Curiosity Modelling

3.1 Feature Extraction for Robot Vision

To maintain the security of an office environment, our security robot patrols the area searching for a specific type of object. It performs this search by constantly obtaining images from its camera and extracting any part of the image that has this type of object. Assume $X = \{\mathbf{x}_1; \mathbf{x}_2; \mathbf{x}_3; \dots; \mathbf{x}_N\}$ represents the set of normalised image vectors used to train the robot (where each image vector is a vectorised image of an object). Given this, the average vector \bar{x} of X is defined as

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i. \quad (1)$$

Each image vector differs from \bar{x} by vector Θ

$$\Theta_i = \mathbf{x}_i - \bar{x} \text{ for } i = 1 \dots N. \quad (2)$$

Principal Component Analysis (PCA) is used to extract features from the image vectors. To do this an eigenvector matrix $U = \{\mathbf{u}_1; \mathbf{u}_2; \dots; \mathbf{u}_M\}$ is obtained by solving the following eigen-decomposition function:

$$UC = \lambda U^T, \quad (3)$$

where λ is the eigenvalue matrix and C is the covariance matrix of X , computed using the following formula:

$$C = \frac{1}{N} \sum_{i=1}^N \Theta_i \Theta_i^T. \quad (4)$$

Although this will represent the discriminant feature space not all of the eigenvectors in U are required to represent it fully. U' is the subset of U containing the M' ($\ll M$) eigenvectors with the largest associated eigenvalue. The features for each image of an object are found by the security robot using the following formula:

$$y = U'^T (\mathbf{x} - \bar{x}), \quad (5)$$

where \mathbf{x} is the image vector and y is the features of \mathbf{x} .

3.2 Vision Curiosity Computation

For threat detection, the security robot is required to get familiar with the office environment (including office employees and static artifacts), and distinguish any outliers from the known objects. For real-time computation of robot, the basic idea of curiosity is to catch quickly the difference of any new acquired information to the known office scenario.

Given X_{tr} as the dataset (i.e. office image features) necessary for the robot to learn the office, X_{tr} represents the office knowledge as a number of data clusters

$X_{s1}, X_{s2}, \dots, X_{sm}$, where each cluster intrinsically represents a different part of the complete office knowledge. As a new image \mathbf{x} acquired by robot, the curiosity is to tell how much the image differs from every part of the office knowledge. In practice, the mean feature vector of each cluster is used to represent each part of knowledge. Thus, to evaluate the difference of \mathbf{x} to cluster $X_{si}, 1 \leq i \leq m$, a mathematical residue is defined

$$\xi_i = \|\mathbf{x} - \bar{\mathbf{x}}_{si}\| \quad (6)$$

where $\bar{\mathbf{x}}_{si} = \frac{1}{|X_{si}|} \sum_{\mathbf{x}_j \in X_{si}} \mathbf{x}_j$ is the mean vector of cluster X_{si} .

Thus, the general difference over the whole office environment is computed as,

$$\langle e \rangle = \min(\xi_1 \dots \xi_m) \quad (7)$$

The above residue is applied to detect threats as, if the calculated $\langle e \rangle$ is greater than the pre-specified threshold, then the robot is considered to confront any unknown objects/threats, otherwise it is considered as a known object in the office.

However, as the curiosity varies over time, the above residue can be evaluated in practice more reliability as the smoothed derivative of the error curve of $\langle e \rangle$ corresponding to a set of recent acquired images,

$$\langle e(t) \rangle = \frac{\sum_{i=0}^{\theta} e(t-i)}{\theta + 1} \quad (8)$$

$$\langle e(t-\tau) \rangle = \frac{\sum_{i=0}^{\theta} e(t-\tau-i)}{\theta + 1} \quad (9)$$

where τ is a time window parameter, and θ a smoothing parameter. Then, the curiosity \mathcal{C} at time (t) can be calculated by as,

$$\mathcal{C}(t) = \frac{\langle e(t) \rangle - \langle e(t-\tau) \rangle}{\tau}. \quad (10)$$

In this way, threat detection is conducted by: if \mathcal{C} is greater than a pre-specified threshold, the the office scene confronted by robot at current moment is judged as containing some unsafe object, otherwise it is considered safe. The above method often confront the difficulty of threshold determination, as the threshold is training data dependent. To solve the difficulty, the curiosity at time (t) can be alternatively calculated by,

$$\mathcal{C}(t) = \langle e(t) \rangle - \langle e(t-\tau) \rangle, \quad (11)$$

In this method, curiosity of threats raises only when $\mathcal{C}(t)$ get a positive value.

4 Tests and Discussions

For testing we envision a scenario where the security robot is patrolling in an office environment, moving along a pre-set path and examining particular security

points en route. A small number of people (less than 10) are allowed in this office, all others are a security threat and their presence in the office should trigger an alarm. As navigating during a security patrol, the robot constantly evaluates the curiosity on every confronted object, which is detected by its vision system. An object/person with a high curiosity will be taken as a possible threat, and will be tracked by the robot.

In a real office environment, we test the developed security robots capabilities with threat detection, security navigation and threat tracking.

4.1 Threats Detection Testing

The scenario assumed for this curiosity test is that the robot has detected a face and now needs to decide if the face is a threat. To test the robots threat detection images of the faces of 11 people in the lab were collected. Two of these people were designated as threatening people, the remaining 9 are considered to be known people (allowed in the office). Additionally faces from the BioID face database [9] were also used as threats for the threat detection testing. If the curiosity value for an image was higher than 5.0, the image was designated a threat.

Initially all 9 known people were used to train the robot, then curiosity was calculated for two different sets of training images. The first 'known' set consisted of images of the known people, the second 'threatening' set was comprised of images from the BioID database as images of the two threatening people. The percentage of unknown faces correctly classified as threats (curious) was recorded, as was the percentage of known faces correctly classified as non-threats (not curious). The robustness of the threat detection to different numbers of known people was tested by removing one of the known people from the training and test images then repeating the test. The results of these tests can be seen in Table. II

As can be seen, the percentage of threatening (unknown) faces correctly regarded as threats varies between 75% and 80%, while the percentage of known faces which were not regarded as threats is $\approx 90\%$ for most of the tests. Both these numbers are lower than our best expectations, we believe this is due to PCAs sensitivity to changes in light, viewing angle etc. Despite this PCA has

Table 1. Threat detection test results

known people	training faces	test faces (known/unknown)	True Curious Rate	True Known Rate
9	1911	5375 (3389/1986)	75.48 %	91.65 %
8	1662	4875 (2889/1986)	75.58 %	90.20 %
7	1413	4375 (2389/1986)	78.90 %	91.34 %
6	1164	4125 (2139/1986)	80.06 %	90.14 %
5	918	3625 (1639/1986)	80.72 %	87.31 %
4	668	3426 (1440/1986)	79.81 %	99.44 %

a distinct advantage in its low computational requirements. The robots VGN-UX70 minicomputer has relatively low computing power, a very accurate algorithm would exceed its capabilities and not run in real time. PCA not only can run in real time on the mincomputer but can also attain good accuracy.

4.2 Realtime Threats Detection

This was tested further using a more realistic experiment. The robots threat detection was first trained using 9 known people as in the previous test. Two different sets of testing images were then submitted to the robot. The first set of testing images were obtained from a known person in the lab, the second set were taken in a different environment, and the subject was a threatening (unknown) person. A sample of the testing images (with the persons face highlighted) can be seen in Fig. 1b. For each set of images a curiosity value was calculated for each image by calculating an average of the curiosity value of the current image and the 3 previous images. The results of this test can be seen in Fig. 1a. Although there is some anomalies within the first 20 timesteps, the majority of the unknown faces have been classified as curious, with a curiosity value above 9.5. Conversely despite some anomalies within the first 10 time steps the majority known faces have all been classified as such, with a curiosity value less than 4. Thus over 85% of the testing images have been correctly classified and there is a large curiosity difference between the known and unknown faces. This large difference improves the robustness of the calculation, small irregularities in the curiosity value will not have a significant effect on the outcome of the classification.

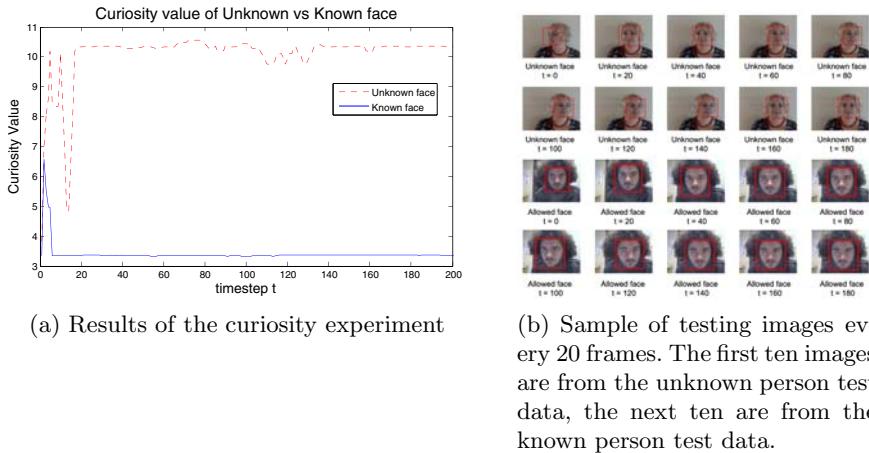


Fig. 1. Realtime Curiosity Test Data and Results

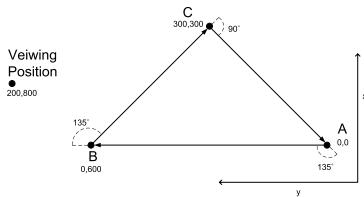


Fig. 2. The patrol path designed for the navigation testing

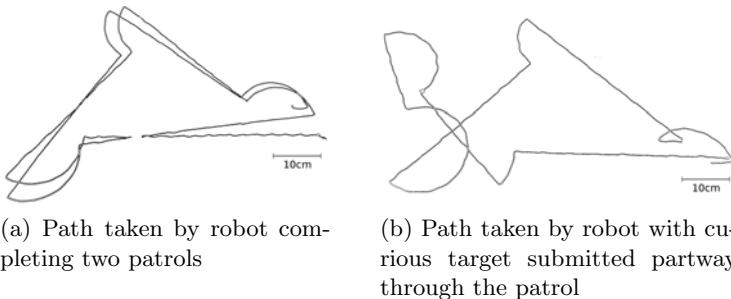


Fig. 3. The path of the robot recorded during the navigation experiments

4.3 Security Navigation and Threat Tracking

Navigation plays an important role in effective robot security. Two major requirements of this are the ability to move along a novel patrol path supplied to it and the ability to be able to deviate from its initial patrol path to investigate some curiosity.

To test these requirements two test cases were used. The first test case was to supply the robot with a patrol path, then observe how closely the robots path matched the supplied patrol path. A triangular path was decided on as seen in Fig. 2. The path the robot took when given this patrol path can be seen in Fig. 3a. Although there are some deviations the robot still does visit each security point in order, completing two patrols successfully.

The second test case of the navigation testing was similar to the previous test. However in this test the navigation system was instructed to move investigate a specific curiosity point partway through its patrol (the appropriate viewing position can be seen in Fig. 2). After ten seconds the robot was instructed to return to its patrol. The resulting path of the robot can be seen in Fig. 3b, the robot moves along its patrol path as before, however a short time after it reaches the second security point the robot moves up from this point and to the left. This is approximately the same position as the viewing position the robot needed to move to (position (200mm,800mm)).

In both tests the robot performs as expected, although their is some deviation from the ideal path. There are numerous possible sources for this deviation,

including the robots wheels slipping. Robot localisation would be needed to correct this deviation.

5 Conclusion and Future Work

A security robot that uses curiosity in unexpected objects to detect threat is being developed. The security robot consists of two modular systems: a vision curiosity system and a navigation system. The vision curiosity system detects faces/object automatically, evaluates face curiosity using PCA, and identifies faces that do not match a set of known faces that the system has previously been trained on. The navigation system can move the robot along a pre-defined patrol path and can move to investigate any curiosity that has been detected.

References

1. Mori, K., Sato, M., Sonoda, T., Ishii, K.: Toward realization of swarm intelligence. In: Proceedings of the 7th POSTECH-KYUTECH Joint Workshop on Neuroinformatics (August 2007)
2. Liu, J.N.K., Wang, M., Feng, B.: iBotGuard: An Internet-Based Intelligent Robot Security System Using Invariant Face Recognition Against Intruder. *IEEE Transactions on Systems, Man and Cybernetics - part B: Cybernetics* 35(1) (2005)
3. Treptow, A., Cielniak, G., Duckett, T.: Active people recognition using thermal and grey images on a mobile robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2103–2108 (2005)
4. Andreasson, H., Magnusson, M., Lilienthal, A.: Has Something Changed Here? Autonomous Difference Detection for Security Patrol Robots. In: Proceedings of the 2007 IEEE International Conference on Intelligent Robots and Systems (2007)
5. Uğur, E., Doğar, M.R., Çakmak, M., Şahin, E.: Curiosity-driven learning of traversability affordance on a mobile robot. In: IEEE 6th International Conference on Development and Learning (2007)
6. Scott, P.D., Markovitch, S.: Learning Novel Domains Through Curiosity and Conjecture. In: Proceedings of International Joint Conference for Artificial Intelligence, pp. 669–674 (1989)
7. Oudeyer, P., Kaplan, F., Hafner, V.V., Whyte, A.: The Playground Experiment: Task-Independent Development of a Curious Robot. In: Proceedings of the AAAI Spring Symposium on Developmental Robotics (2005)
8. Shimo, N., Pang, S., Kasabov, N., Yamakawa, T.: Curiosity-Driven Multi-Agent Competitive and Cooperative LDA Learning. *International Journal of Innovative Computing, Information & Control* 4(7), 1537–1552 (2008)
9. BioID Face Database [online].
<http://www.bioid.com/downloads/facedb/index.php>

Multiobjective Multiclass Soft-Margin Support Vector Machine Maximizing Pair-Wise Interclass Margins

Keiji Tatsumi, Ryo Kawachi, Kenji Hayashida, and Tetsuzo Tanino

Osaka University, Yamada-Oka, Suita, Osaka, Japan

`tatsumi@eei.eng.osaka-u.ac.jp,`
`hayashid@sa.eei.eng.osaka-u.ac.jp,`
`kawachi@sa.eei.eng.osaka-u.ac.jp,`
`tanino@eei.eng.osaka-u.ac.jp`

Abstract. The *all together* model is one of the support vector machine (SVM) for multiclass classification by using a piece-wise linear function. As a novel all together model, we already proposed a hard-margin multi-objective SVM model for piecewise linearly separable data, which maximizes all of the geometric margins simultaneously for the generalization ability. In addition, we derived a single-objective convex problem whose optimal solution is weakly Pareto optimal for the proposed SVM. However, in the real-world classification problem the data are often piecewise linearly inseparable. Therefore, in this paper we extend the hard-margin SVM for the data by introducing penalty functions for the margin slack variables based on the geometric distances between outliers and the support hyperplanes, and incorporating those functions into the objective functions. Moreover, we derive a single-objective second-order cone programming (SOCP) problem, and show that its optimal solution is weakly Pareto optimal for the proposed soft-margin SVM. Furthermore through numerical experiments we verify that the SOCP model maximizes the geometric margins in the sense of multiobjective optimization.

1 Introduction

The support vector machine (SVM) is a powerful machine learning method for binary-class classification problems. Some kinds of extensions to multiclass classification have been investigated [5]. In this paper, we focus on the *all together* method which finds a discriminant function directly by solving an optimization problem with all patterns [24][10][11], where all patterns are classified into the corresponding classes by using a piece-wise linear function. This model is formulated as a single-objective optimization problem of maximizing the sum of margins between all of the pairs of classes, where the margin is defined as the distance between two normalized support hyperplanes parallel to the corresponding discriminant hyperplane. However, as we point out it in [7][8][9], the margin is not always equal to the geometric margin which is defined as the minimal distance of

patterns to the corresponding discriminant hyperplane, and the geometric margin can exactly indicate the relation between each pattern and the discriminant function. Therefore, in [8,9], we emphasized that maximizing the geometric margins is important for the generalization of multiclass classification, and proposed a hard-margin multiobjective SVM model which maximizes all of the geometric margins simultaneously. Moreover, we derived a single-objective second-order cone programming (SOCP) problem by using scalarization approaches for multiobjective optimization, which are solvable convex programming problems, and showed theoretically that the optimal solution of the SOCP is weakly Pareto optimal for the proposed multiobjective model. Moreover, we applied them to some examples to demonstrate that the proposed models can achieve maximization of the geometric margins.

In this paper, we consider the classification problem for piecewise linearly inseparable data, which are often shown in the real-world problems. Thus, we extend the proposed hard-margin model into a soft-margin one by introducing penalty functions of the margin slack variables based on the geometric distances between outliers and the support hyperplanes, and incorporating those functions into the objective functions. In addition, we derive an SOCP model whose optimal solution is weakly Pareto optimal for the proposed soft-margin SVM model. Moreover, we verify that the SOCP model maximizes the margins and minimizes the penalty functions in the sense of multiobjective optimization through some numerical experiments.

2 Multiclass Classification

2.1 All Together Model

In this paper, we consider the following multiclass classification problem: For given data $D = \{x^i, y_i\}, i = 1, \dots, m$, where $x^i \in \Re^n$ is an input pattern and $y_i \in K := \{1, \dots, k\}$ denotes the corresponding class, we construct a classifier which divides all patterns into the corresponding classes:

$$f(x) = \arg \max_p \{w^{p\top} x + b^p\}. \quad (1)$$

where $w^p \in \Re^n$ and $b^p \in \Re$, $p \in K$ are decision variables and the linear function $w^{p\top} x + b^p$ indicates the degree of confidence when a point x is classified into class p . Then,

$$(w^p - w^q)^\top x + (b^p - b^q) = 0, \quad q \neq p, \quad p, q \in K, \quad (2)$$

is the discriminant hyperplane which distinguishes between classes p and q . Note that the representation of discriminant hyperplanes (2) is not unique. For any constants $t(\neq 0)$, $s \in \Re$ and any vector $v \in \Re^n$, $(w^{1\top}, \dots, w^{k\top})$, (b^1, \dots, b^k) and $(tw^{1\top} + v^\top, \dots, tw^{k\top} + v^\top)$, $(tb^1 + s, \dots, tb^k + s)$ are different representations of the same discriminant hyperplanes. The discriminant function (1) is used in *all together* model or *one against all* model, which are representative SVMs for multiclass classification [5]. In this paper, we focus on the all together model.

Now, in this section, we suppose that data D are piecewise linearly separable. Then, there exist an infinite number of discriminant functions to distinguish all classes correctly. In the multiclass classification, on the analogy of the binary SVM, the model maximizing $1/\|w^p - w^q\|$ for all pairs (p, q) , $q \neq p$, $p, q \in K$ was proposed [2,4,10,11],

$$(O) \quad \begin{aligned} \min_{w,b} \quad & \frac{1}{2} \sum_{p=1}^k \sum_{q=1, q \neq p}^k \|w^p - w^q\|^2 \\ \text{s.t.} \quad & (w^p - w^q)^\top x^i + (b^p - b^q) \geq 1, \quad i \in I_p, \quad q \neq p, \quad p, q \in K, \end{aligned}$$

where I^p denotes an index set defined by $I^p := \{i \in \{1, \dots, m\} \mid y^i = p\}$. However, the margin in model (O) is not necessarily equal to the *geometric margin* defined as the distance of the nearest pattern in a pair of classes to the corresponding discriminant hyperplane classifying all patterns in the pair of classes correctly, as we pointed out it in [8,9].

$$d_{pq}^g(w, b) := \min \left\{ \min_{i \in I_p} \frac{|(w^p - w^q)^\top x^i + (b^p - b^q)|}{\|w^p - w^q\|}, \min_{i \in I_q} \frac{|(w^p - w^q)^\top x^i + (b^p - b^q)|}{\|w^p - w^q\|} \right\}, \quad q > p, \quad p, q \in K.$$

Thus, it cannot guarantee that margins obtained by minimizing $\|w^p - w^q\|$, $q \neq p \in K$ in the model (O) are equal to the corresponding geometric margins $d_{pq}^g(w, b)$. Therefore, in [8,9] we proposed a hard-margin multiobjective SVM (M1) which maximizes all geometric margins simultaneously.

$$(M1) \quad \begin{aligned} \max_{w,b} \quad & \left(d_{12}^g(w, b), d_{13}^g(w, b), \dots, d_{(k-1)k}^g(w, b) \right) \\ \text{s.t.} \quad & (w^p - w^q)^\top x^i + (b^p - b^q) \geq 1, \quad i \in I_p, \quad q \neq p, \quad p, q \in K, \end{aligned}$$

Moreover, since model (M1) is difficult to solve directly, we proposed the following model (M2) using a vector $\sigma \in \Re^{k(k-1)/2}$:

$$(M2) \quad \begin{aligned} \max_{w,b,\sigma} \quad & (\theta_{12}(w, \sigma), \theta_{13}(w, \sigma), \dots, \theta_{(k-1)k}(w, b, \sigma)) \\ \text{s.t.} \quad & (w^p - w^q)^\top x^i + (b^p - b^q) \geq \sigma_{pq}, \quad i \in I_p, \quad q > p, \quad p, q \in K, \\ & (w^q - w^p)^\top x^i + (b^q - b^p) \geq \sigma_{pq}, \quad i \in I_q, \quad q > p, \quad p, q \in K, \\ & \sigma_{pq} \geq 1, \quad q > p, \quad p, q \in K, \end{aligned}$$

where $\theta_{pq}(w, b)$, $q > p$, $p, q \in K$ is defined by

$$\theta_{pq}(w, \sigma) = \frac{\sigma_{pq}}{\|w^p - w^q\|}.$$

Suppose that there exist Pareto optimal solutions of (M2). Then, we showed that the optimal solutions of (M2) can be considered to be equivalent to those of (M1) as follows [8,9].

Theorem 1. If (w^*, b^*, σ^*) is Pareto optimal for (M2), (w^*, b^*) is Pareto optimal for (M1). Conversely, if (w^*, b^*) is Pareto optimal for (M1), $(w^*, b^*, \sigma(w^*, b^*))$ is Pareto optimal for (M2), where an element of $\sigma(w, b)$ is defined by

$$\begin{aligned}\sigma_{pq}(w, b) := \min & \left\{ \min_{i \in I_p} |(w^p - w^q)^\top x^i + (b^p - b^q)|, \right. \\ & \left. \min_{i \in I_q} |(w^q - w^p)^\top x^i + (b^q - b^p)| \right\}, \quad q > p, \quad p, q \in K.\end{aligned}$$

In addition, we derived two kinds of single-objective optimization problems by scalarization approaches to multiobjective optimization, ε -constraint approach and Benson's method, and transform them into single-objective second-order cone programming (SOCP) problems which are solvable convex programming ones. We called an SOCP model based on ε -constraint approach (ε -P2). Furthermore, we showed theoretically that weakly Pareto optimal solutions of the multiobjective problem (M2) can be obtained by solving SOCP models, and applied them to some simple examples to demonstrate that they can achieve maximization of the geometric margins [89].

However, in the real-world classification problem the data are often piecewise linearly inseparable. Thus, we extend the hard-margin SVM to a soft-margin one for the data in the next section.

3 Multiobjective Soft-Margin Model Maximizing Geometric Margins

Now, we focus on the classification problem for piecewise linearly inseparable data. In the case, instead of the existing hard-margin model (O), the following model (OS) is often used:

$$\begin{aligned}(OS) \quad \min_{w, b} \quad & \frac{1}{2} \sum_{p=1}^k \sum_{q=1, q>p}^k \|w^p - w^q\|^2 + C_\xi \sum_{p=1}^k \sum_{q=1, q>p}^k \eta_{pq}(\xi) \\ \text{s.t.} \quad & (w^p - w^q)^\top x^i + (b^p - b^q) \geq 1 - \xi_{pqi}, \quad i \in I_p, \quad q \neq p, \quad p, q \in K, \\ & \xi_{pqi} \geq 0, \quad i \in I_p, \quad q \neq p, \quad p, q \in K,\end{aligned}$$

where ξ_{pqi} is a margin slack variable to relax the constraints, and $\eta_{pq}(\xi)$ is a penalty function defined by

$$\eta_{pq}(\xi) := \sum_{i \in I_p} \xi_{pqi} + \sum_{i \in I_q} \xi_{qpi}, \quad q > p, \quad p, q \in K.$$

Next, we extend the hard-margin model (M2) by considering a new penalty function $\zeta(\sigma, \xi)$ defined by

$$\zeta_{pq}(\sigma, \xi) := \frac{\frac{\eta_{pq}(\xi)}{\|w^p - w^q\|}}{\frac{\sigma_{pq}}{\|w^p - w^q\|}} = \frac{\eta_{pq}(\xi)}{\sigma_{pq}}, \quad q > p, \quad p, q \in K,$$

which is a normalized sum of geometric distances $\eta_{pq}(\xi)/\|w^p - w^q\|$ between outliers and the support hyperplanes by the margin $\theta_{pq}(w, \sigma) = \sigma_{pq}/\|w^p - w^q\|$. These functions are incorporated into the objective functions $\theta_{pq}(w, \sigma)$ in the following model:

$$\begin{aligned} & \max_{w, b, \sigma, \xi} (\theta_{12}(w, \sigma), \dots, \theta_{(k-1)k}(w, \sigma), -\zeta_{12}(\sigma, \xi), \dots, -\zeta_{(k-1)k}(\sigma, \xi)) \\ \text{s.t. } & (w^p - w^q)^\top x^i + (b^p - b^q) \geq \sigma_{pq} - \xi_{pqi}, \quad i \in I_p, \quad q > p, \quad p, q \in K, \\ & (w^q - w^p)^\top x^i + (b^q - b^p) \geq \sigma_{pq} - \xi_{qpi}, \quad i \in I_q, \quad q > p, \quad p, q \in K, \\ & \sigma_{pq} \geq 1, \quad q > p, \quad p, q \in K, \\ & \xi_{pqi} \geq 0, \quad i \in I_p, \quad q \neq p, \quad p, q \in K. \end{aligned} \quad (\text{MS2})$$

We can see that (MS2) maximizes all geometric margins and minimizes all penalty functions, simultaneously. However, since (MS2) is complicate, we derive a single-objective convex optimization problem whose optimal solution is weakly Pareto optimal for (MS2) in the next section.

4 SOCP Model Based on ε -Constraint Approach

In this section, we propose the following single-objective model, which is derived from the ε -constraint approach to (MS2):

$$\begin{aligned} & \max_{w, b, \sigma, \xi} \theta_{rs}(w, \sigma) = \frac{\sigma_{rs}}{\|w^r - w^s\|} \\ \text{s.t. } & \theta_{pq}(w, \sigma) \geq \varepsilon_{pq}, \quad q > p, \quad (p, q) \neq (r, s), \quad p, q \in K, \\ & \zeta_{pq}(\sigma, \xi) \leq \mu_{pq}, \quad q > p, \quad p, q \in K, \\ (\varepsilon\text{-PS}) \quad & (w^p - w^q)^\top x^i + (b^p - b^q) \geq \sigma_{pq} - \xi_{pqi}, \quad i \in I_p, \quad q > p, \quad p, q \in K, \\ & (w^q - w^p)^\top x^i + (b^q - b^p) \geq \sigma_{pq} - \xi_{qpi}, \quad i \in I_q, \quad q > p, \quad p, q \in K, \\ & \sigma_{pq} \geq 1, \quad q > p, \quad p, q \in K, \\ & \xi_{pqi} \geq 0, \quad i \in I_p, \quad q \neq p, \quad p, q \in K, \end{aligned}$$

where a pair (r, s) and constants ε_{pq} , $(p, q) \neq (r, s)$, $p > q \in K$ and μ_{pq} , $p > q \in K$ are appropriately selected such that the feasible region of $(\varepsilon\text{-PS})$ is not empty. This method maximizes only one of the objectives of (MS2) while the others are transformed to constraints. Then, the following theorems are known about the ε -constraint approach.

Theorem 2. [3] Let (w, b, σ, ξ) be an optimal solution of $(\varepsilon\text{-PS})$ for some (r, s) . Then (w, b, σ, ξ) is weakly Pareto optimal for (MS2).

Theorem 3. [3] (w, b, σ, ξ) is Pareto optimal for (MS2) if and only if there exist an ε_{-rs} and a μ such that (w, b, σ, ξ) is optimal for $(\varepsilon\text{-PS})$ for any (r, s) , $r, s \in K$.

Here ε_{-rs} denotes a vector in which the element ε_{rs} is removed from $k(k-1)/2$ -dimensional vector ε . These theorems imply that any Pareto optimal solution of (MS2) can be obtained by solving $(\varepsilon\text{-PS})$ with an appropriate choice of ε_{-rs} and μ .

However, $(\varepsilon\text{-PS})$ is also difficult to solve because of its fractional nonlinear objective functions. Hence, by making use of one degree of freedom of $(\varepsilon\text{-PS})$, we add a constraint $\sigma_{rs} = c_{rs}$ with an appropriate constant $c_{rs} \geq 1$ to obtain the following model:

$$\begin{aligned}
 & \max_{w, b, \sigma, \xi} \frac{c_{rs}}{\|w^r - w^s\|} \\
 \text{s.t. } & \theta_{pq}(w, \sigma) = \frac{\sigma_{pq}}{\|w^p - w^q\|} \geq \varepsilon_{pq}, \quad q > p, \quad (p, q) \neq (r, s), \quad p, q \in K, \\
 & \zeta_{rs}(\sigma_{-rs}, c_{rs}, \xi) = \frac{\eta_{rs}(\xi)}{c_{rs}} \leq \mu_{rs}, \\
 & \zeta_{pq}(\sigma, \xi) = \frac{\eta_{pq}(\xi)}{\sigma_{pq}} \leq \mu_{pq}, \quad q > p, \quad (p, q) \neq (r, s), \quad p, q \in K, \\
 (\varepsilon\text{-PS2}) \quad & (w^r - w^s)^\top x^i + (b^r - b^s) \geq c_{rs} - \xi_{rsi}, \quad i \in I_r, \\
 & (w^s - w^r)^\top x^i + (b^s - b^r) \geq c_{rs} - \xi_{sri}, \quad i \in I_s, \\
 & (w^p - w^q)^\top x^i + (b^p - b^q) \geq \sigma_{pq} - \xi_{pqi}, \\
 & \quad i \in I_p, \quad q > p, \quad (p, q) \neq (r, s), \quad p, q \in K, \\
 & (w^q - w^p)^\top x^i + (b^q - b^p) \geq \sigma_{pq} - \xi_{qpi}, \\
 & \quad i \in I_q, \quad q > p, \quad (p, q) \neq (r, s), \quad p, q \in K, \\
 & \sigma_{pq} \geq 1, \quad q > p, \quad (p, q) \neq (r, s), \quad p, q \in K, \\
 & \xi_{pqi} \geq 0, \quad i \in I_p, \quad q \neq p, \quad p, q \in K,
 \end{aligned}$$

where σ_{-rs} denotes the vector in which the element σ_{rs} is removed from σ . This problem can be regarded as a second order cone programming (SOCP) problem. The SOCP is a convex programming problem having a linear objective function and linear and second-order cone constraints, which can be efficiently solved by a number of methods such as the primal-dual interior point method within the almost same time as a quadratic programming problem of the same size [1]. Moreover, several commercial and noncommercial solvers have been developed [6]. Then, we can show the following theorems.

Theorem 4. Let $(\hat{w}, \hat{b}, \hat{\sigma}, \hat{\xi})$ be an optimal solution of $(\varepsilon\text{-PS})$ and $c_{rs} = t\hat{\sigma}_{rs}$ for $t \geq 1$. If $(w^*, b^*, \sigma_{-rs}^*, \xi^*)$ is an optimal solution of $(\varepsilon\text{-PS2})$, then $(w^*, b^*, (\sigma_{-rs}^*, c_{rs}), \xi^*)$ is optimal for $(\varepsilon\text{-PS})$.

Theorem 5. If $(w^*, b^*, \sigma^*, \xi^*)$ is an optimal solution of $(\varepsilon\text{-PS})$, then for any $t \geq 1$, $(tw^*, tb^*, t\sigma_{-rs}^*, t\xi^*)$ is an optimal solution of $(\varepsilon\text{-PS2})$ with $c_{rs} = t\sigma_{rs}^*$.

Theorem 4 shows that for an optimal solution $(w^*, b^*, \sigma_{-rs}^*, \xi^*)$ of $(\varepsilon\text{-PS2})$, $(w^*, b^*, (\sigma_{-rs}^*, c_{rs}), \xi^*)$ is optimal for $(\varepsilon\text{-PS})$. Thus, Theorem 2 implies that the optimal solution is weakly Pareto optimal for (MS2). In addition, the result together with Theorems 3 and 5 suggests that we can obtain any Pareto optimal solution of (MS2) by solving $(\varepsilon\text{-PS2})$ with an appropriate choice of ε_{-rs} and μ . Consequently, we can conclude that various discriminant functions maximizing the geometric margins can be obtained by solving $(\varepsilon\text{-PS2})$ as a pair (r, s) and the corresponding constant vectors ε_{-rs} and μ are varied.

Finally, in the next section, we apply the proposed models to some classification problems.

5 Numerical Examples

In this section, we report the results of numerical experiments where existing models and proposed models were applied to benchmark problems, models (O) and (ε -P2) for Wine and models (OS) and (ε -PS2) for Iris, respectively. Note that we used hard-margin models for Wine because Wine data can be regarded as piecewise linearly separable. We used optimization tools in MathWorks Matlab 7.0.1¹ and Mosek version 5.0² to solve four models. We set $(r, s) = (2, 3)$ and $c_{rs} = 10$, and constant vectors ε_{-rs} and μ in (ε -P2) and (ε -PS2) were determined on the basis of solutions obtained by the model (O) and (OS), respectively.

We compared objective function values of (M2) and (SM2) at solutions obtained by the existing and proposed models for Wine and Iris, respectively. We can observe that all solutions obtained by the proposed models dominate ones by existing models. Tables 1 and 2 show a part of obtained results, which indicate classification rates and objective function values of solutions obtained by two models.

Table 1. Comparison of results obtained by two models for Wine

Model	Classification rate		Objective function values		
	(training)	(test)	θ_{12}	θ_{13}	θ_{23}
(O)	100.00	88.89	0.3812	0.4858	0.4368
(ε -P2)	100.00	94.44	0.3814	0.4889	0.5127

Table 2. Comparison of results obtained by two models for Iris

Model	Classification rate		Objective function values					
	(training)	(test)	θ_{12}	θ_{13}	θ_{23}	ζ_{12}	ζ_{13}	ζ_{23}
(OS)	98.52	100.00	0.9804	0.3047	0.4017	0.6640	0.0000	12.07
(ε -PS2)	99.26	100.00	0.9816	1.4908	0.4021	0.6610	0.0000	12.07

Therefore, we can verify that Pareto optimal solutions of (S2) or (MS2) are obtained by solving (ε -P2) or (ε -PS2).

6 Conclusion

In this paper, we have focused on the *all together* model of the support vector machine (SVM) for multiclass classification, which uses a piece-wise linear function to construct a discriminant function. We have pointed out that the margins maximized in the existing all together model are necessarily equal to the geometric margins defined as the minimal distance of patterns to the corresponding discriminant hyperplane and that maximizing geometric margins is

¹ <http://www.mathworks.com/>

² <http://www.mosek.com/>

important for the generalization. Moreover, although in the existing model the sum of margins between all pairs of classes is maximized, the multiclass classification should be essentially formulated as a multiobjective optimization problem which maximizes all geometric margins.

Therefore, we have proposed a multiobjective soft-margin SVM model which maximizes all geometric margins and minimizes penalty functions based on the distances between outliers and the support hyperplanes simultaneously. In order to solve the proposed soft-margin model, we have derived single-objective second-order cone programming (SOCP) problem by the ε -constraint approach and some techniques, which can be efficiently solved by several interior point methods. Furthermore, we have shown theoretically that the optimal solution of the proposed SOCP model is weakly Pareto optimal for the multiobjective soft-margin SVM, and verified those results through some numerical experiments.

For future works, we should investigate the potential performance of the proposed multiobjective SVM model. Therefore, we need to estimate the classification ability of discriminant functions corresponding to many Pareto optimal solutions of the SVM model which can be obtained by solving the SOCP model with a pair (r, s) and the corresponding constant vectors ε_{-rs} and μ varying.

References

1. Alizadeh, F., Goldfarb, D.: Second-order cone programming. *Mathematical Programming, Ser. B* 95, 3–51 (2003)
2. Bredensteiner, E.J., Bennett, K.P.: Multicategory classification by support vector machines. *Computational Optimization and Applications* 12, 53–79 (1999)
3. Ehrgott, M.: *Multicriteria optimization*. Springer, Berlin (2005)
4. Guermeur, Y.: Combining discriminant models with new multiclass SVMs, Neuro COLT2 Technical Report Series (2000)
5. Hsh, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* 13(2), 181–201 (2002)
6. Mittelmann, H.D.: An independent benchmarking of SDP and SOCP solvers. *Mathematical Programming, Ser. B* 95, 407–430 (2003)
7. Tatsumi, K., Hayashida, K., Higashi, H., Tanino, T.: Multi-objective multiclass support vector machine for pattern recognition. In: Proceedings of SICE Annual Conference 2007, pp. 1095–1098 (2007)
8. Tatsumi, K., Kawachi, R., Hayashida, K., Tanino, T.: Multi-objective multiclass support vector machine maximizing exact margins. In: Proc. ICOTA7, pp. 381–382 (2007)
9. Tatsumi, K., Tanino, T., Hayashida, K.: Multiobjective multiclass support vector machine for pattern recognition. In: Proc. 4th Taiwan-Japan Symposium on Fuzzy Systems and Innovational Computing, pp. 94–100 (2007)
10. Vapnik, V.: *Statistical learning theory*. A Wiley-Interscience Publication, Hoboken (1998)
11. Weston, J., Watkins, C.: Multi-class support vector machines, Technical report CSD-TR-98-04, Univ. London, Royal Holloway (1998)

Functional Networks Based on Pairwise Spike Synchrony Can Capture Topologies of Synaptic Connectivity in a Local Cortical Network Model

Katsunori Kitano and Kazuhiro Yamada

Department of Human & Computer Intelligence, Ritsumeikan University,
1-1-1 Nojihigashi, Kusatsu, Shiga 5258677, Japan

Abstract. In order to develop a method to infer the underlying synaptic connectivity only from spatio-temporal patterns of spiking activity observed in a neuronal network, we here investigated characteristics of a network of functional connections defined by pairwise spike synchrony. We first conducted numerical simulations of a computational model of a local cortical network and constructed a functional network based on the obtained spike data. The proposed analysis with the optimal parameters defining functional connections showed that characteristics of connectivity of functional networks are in good agreement with those of synaptic connectivity in the computational model in terms of statistical indices such as the clustering coefficient and the shortest path length. The result suggests that it is possible to extract at least statistical characteristics of synaptic connectivity from spatio-temporal patterns of spiking activity.

1 Introduction

The essence of neural information processing is interactions between a huge number of neurons. Structure of a spatial synaptic wiring pattern in a neuronal network is one of the primary determinants to shape a way of communications between neurons. Therefore, understanding of a role of synaptic connectivity in neural information processing has been of importance.

Recently, effective synaptic connectivity has been intensively studied with the method combining anatomy and electrophysiology [12,34]. It is true such an analysis is quite powerful to reveal a detailed connectivity even between same or different cell types, but its usefulness is limited to only a very local region of a network. On the other hand, optical imaging techniques such as a Ca^{2+} influx measurement have visualized spatio-temporal patterns of neural activity in comparatively large scale neural networks [5,6]. Since information on the underlying network structure must be included in the observed neural activity, especially if it is spontaneous activity, it might be possible to extract information on synaptic connectivity from the spatio-temporal patterns of neural activity.

In the present study, we investigated similarities between synaptic connectivity and connectivity of the “functional” network constructed based on neural

activity using a local cortical network model. We first conducted numerical simulations of our cortical model changing the parameters defining spatial patterns of synaptic connections. Using spike data obtained by the simulations, we next defined a “functional” connection for a neuron pair by pairwise spike coherence and compared such a functional network with the synaptic network of our cortical model in terms of statistics characterizing the network connectivity. Finally, we explored parameters used to define a functional connection to examine whether the functional network can capture properties of synaptic connectivity.

2 Materials and Methods

2.1 Computational Model of a Local Cortical Network

A local cortical network model simulated in the present study is basically the same as the model proposed in our previous work [7]. The network consisted of 4096 excitatory neurons and 1024 inhibitory neurons. The regular spiking type of neurons are modeled as excitatory neurons whereas the fast spiking type of neurons were incorporated as inhibitory neurons [8,9]. In models of synapses, synaptic currents through AMPA, NMDA and GABAergic channels were incorporated [10,11]. In addition, each of neurons received background synaptic inputs that could produce in vivo-like spiking activity with high variability [12].

Both types of neurons were arranged in a two-dimensional grid. A neuron pair within a distance (a four-grid interval) was connected, which made a two-dimensional regular network. For the regular network, we introduced a random wiring probability p into excitatory synaptic connections, following the small-world network paradigm [13]. We randomly chose synaptic connections to be rewired with the probability, then randomly chose a neuron as the presynaptic neuron of each the chosen connection. The resultant network showed similar network topology to one obtained by the normal operation in the small-world paradigm, that is, a network with a large value of p showed a random-like structure whereas one with an intermediate value of p is a small-world network (see “Characterization of network connectivity”). We typically investigated the regular pattern ($p = 0.00$), a small-world one ($p = 0.05$) and a random one ($p = 1.00$).

2.2 Spike Data Analysis

To examine a degree of pairwise spike synchronization, we applied the coherence measure to pairs of spike trains [14]. Supposed that $\{x_i(k)\}$ ($k = 1, 2, \dots, N_b$) is the binary sequence obtained by binning the spike train of neuron i , the measure for the pair was calculated as follows,

$$\kappa_{ij}(\tau) = \frac{\sum_k^{N_b} x_i(k + \tau)x_j(k)}{\sqrt{\sum_k^{N_b} x_i(k)\sum_k^{N_b} x_j(k)}}. \quad (1)$$

For all the possible pairs, we obtained values of the measure setting the bin size as 5ms.

2.3 Definition of Functional Connections

According to values of the coherence measure, we defined functional connections for neuron pairs. We employed two ways to determine whether a neuron pair is functionally connected or not. In the first manner, we obtained a distribution of the measure out of randomly sampled pairs and then set the P -value of the distribution as a threshold value. If a value of the measure for a neuron pair was larger than the threshold, we made the pair connect functionally. The second method was that we set the criteria of functional connections cell by cell. To determine which cells were functionally connected to a reference neuron i , we first sorted the other neurons in descending order of values of the measure between each of them and the neuron i and then selected N_c neurons from the head of the order as functionally connected neurons with the neurons i . We here referred to the first and the second methods as *the common threshold* and *the minimum connection number*, respectively. We should note that the functional connections by both the methods were defined as non-directed ones because of the nature of the coherence measure.

2.4 Characterization of Network Connectivity

To characterize structure of networks, some statistics are commonly used. The degree is the number of connections that a neuron has. The clustering coefficient is the probability that three neurons make a close loop. The shortest path length is the minimum number of connections from a neuron to another. We here calculated the clustering coefficients and the shortest path lengths and averaged them over the network in order to represent topological characteristics of the synaptic networks and functional networks. Since these statistics are basically defined for non-directed networks, we here considered the synaptic networks as non-directed ones. While both indices are large in the regular network, both become small in the random network. When the average of clustering coefficients remains large and the average of shortest path lengths take a small value, we refer to the connectivity as “small-world” [13].

3 Results

We conducted numerical simulations with our cortical network model changing not only the random rewiring probability p but a multiplier factor of excitatory synaptic strengths that considerably affected patterns of neural activity. Typical values of p are 0.00, 0.05, and 1.00. And the multiplier factor of the synaptic strength was set as 1.00, 1.125, and 1.250 that yielded an excitatory postsynaptic potential of 0.8, 0.9 and 1.0 mV at the resting state, respectively.

3.1 Distribution of Pairwise Spike Coherence

We first confirmed if networks with different synaptic connectivity generated different patterns of neural activity. To show the results quantitatively, we obtained

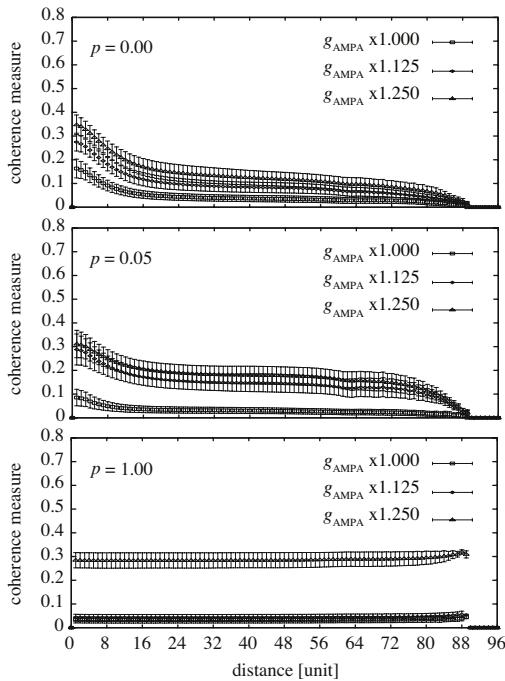


Fig. 1. Dependence of the coherence measure on distances between neurons in pairs. From top to bottom, results for the regular ($p = 0.00$), a small-world ($p = 0.05$), and a random network ($p = 1.00$) are indicated. In each panel, results with different multiplier factors of excitatory synaptic strengths are shown. *symbols* and *error bars* represents means and standard deviations, respectively.

distributions of the coherence measure on distances between neurons in Fig.1. While the synchrony rapidly decayed with an increase in the distances in the regular network with $p = 0.00$, the measure sustained even for distant pairs of neurons in the small-world network with $p = 0.05$. In the random network with $p = 1.00$, the degree of spike synchrony is independent of the distances. Thus, such a dependence of the measure on distances between neurons suggested characteristics of synaptic connectivity, namely, distributions of lengths of synaptic connections.

3.2 Coherence Measure and Underlying Structure

It is thought that there are two underlying structures to induce high coherent activity for neuron pairs: mutual connections between neuron pairs and common inputs to them. To see if and to what degree the coherence measure can detect mutual connections rather than common inputs, we obtained the posterior probability of existence of mutual connections when a value of the coherence measure was observed for a neuron pair. Let random variables X and Θ represent the

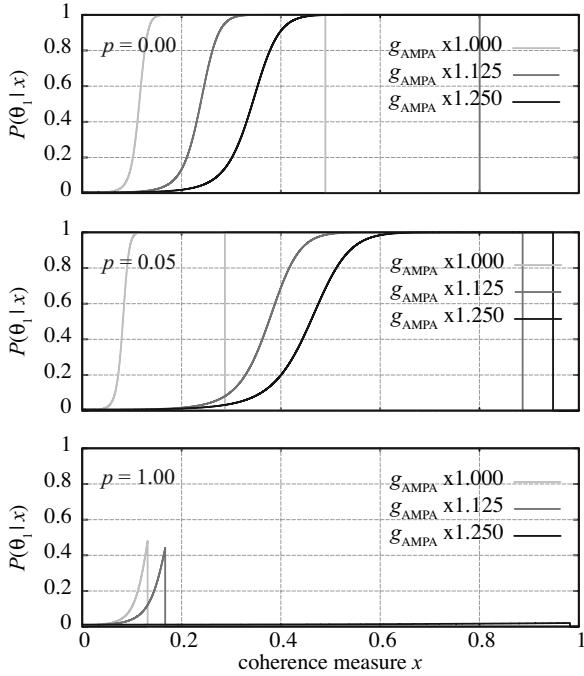


Fig. 2. Posterior probability that there exist mutual connections in a pair of neurons when the coherence measure took a value of x , $P(\Theta = \theta_1|X = x)$. From top to bottom, results for the regular ($p = 0.00$), a small-world ($p = 0.05$), and a random network ($p = 1.00$) are indicated. In each panel, results with different multiplier factors of excitatory synaptic strengths are shown.

observed coherence measure and existence of mutual connections, respectively. The discrete variable Θ took two values; θ_0 meant that there is no mutual connections whereas θ_1 denoted there exists such connections. Figure 2 indicates the probability $P(\Theta = \theta_1|X = x)$. In the regular and the small-world network, as the observed value of the coherence measure for a neuron pair increased, the probability monotonically increased, which implied that the high values of the coherence measure could be related to existence of mutual connections to considerable extent. However, in the random network, the probability reached only a chance level even with an increase in the coherence measure.

3.3 Comparison between Functional and Synaptic Networks

At first, we compared the functional network with the synaptic network in a direct manner. In such a direct comparison, we optimized the parameter P -value or N_c so that the mean squared error between the statistics of the functional networks and that of synaptic ones was minimized. When p remained small, the correlation coefficient between the matrix of functional connections and that of synaptic connections kept values around 0.6. However, as p increased, the

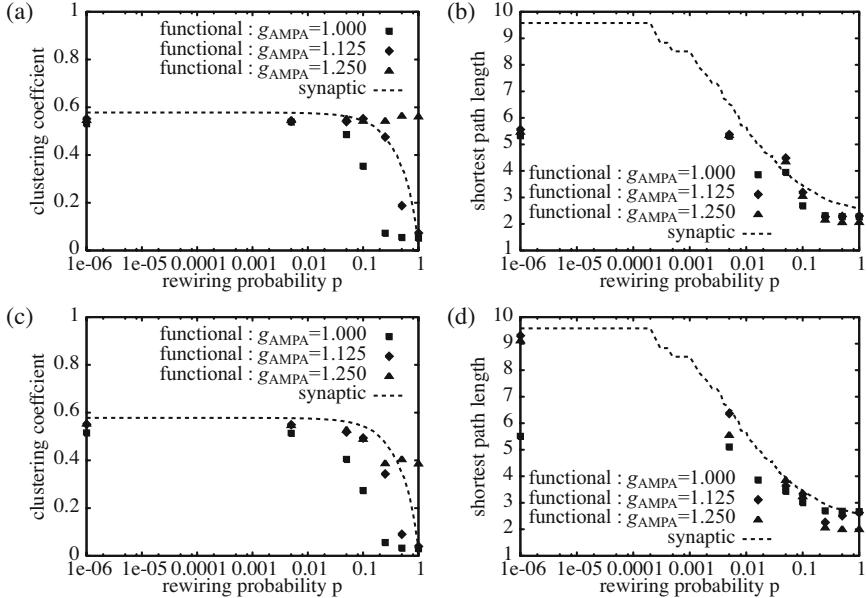


Fig. 3. Comparison between functional and synaptic connectivity in terms of the statistics. (a) clustering coefficients of functional networks defined by the common threshold over a network. (b) shortest path lengths by the same analysis as (a). (c) clustering coefficients of functional networks defined by the common number of connections over a network. (d) shortest path lengths by the same analysis as (c).

coefficient rapidly decreased (not shown). The result implied that it be hard to construct a functional network similar to the synaptic one at a detailed level.

We next compared the two networks with each other in terms of the statistical indices, the clustering coefficient and the shortest path length. As illustrated in Fig. 3, the first analysis, the common threshold, failed to capture the structural characteristics of the synaptic connectivity (Fig. 3(a) and (b)) since the shortest path length of the functional network took small values especially for small p . On the other hand, results by the second analysis, the minimum connection number, were in better agreement with the statistics of the synaptic network, especially for an intermediate multiplier factor of 1.125 (Fig. 3(c) and (d)). Taken together, we concluded that the minimum connection number yielded better results than the common threshold in constructing the functional network similar to the underlying synaptic network.

4 Conclusions

In the present study, we investigated the relationship between synaptic networks and functional networks defined by pairwise spike synchrony. Some studies have applied similar functional network analyses to experimental data obtained by

the Ca^{2+} influx imaging or multielectrode arrays so far [15][16]. However, an advantage of using spike data obtained by simulations with a computational model is to have detailed configurations on not only functional networks but also synaptic ones, which enables us to compare them with each other and reveal the detailed relationship between them.

Although the coherence measure was a simple index to describe a degree of pairwise spike synchrony, it could predict the underlying structure to considerable extent (Fig.2). However, the measure tended to be strongly influenced by variations of firing rates and to become larger with an increase in the firing rate (not shown). Therefore, as a result of the common threshold analysis, the neurons with high firing rates were given many functional connections whereas the neurons with low firing rates were given less such connections. On the other hand, the minimum connection number analysis, which improved such a tendency by hand, could capture the statistical characteristics of the synaptic connectivity. This results implies possibility to infer statistical nature of underlying synaptic connectivity based on spatio-temporal patterns of spiking activity, if a method to estimate the degree can be developed and/or a method to detect a degree of spike coherence independent of firing rates is applied [17].

References

- Yoshimura, Y., Callaway, E.M.: Fine-scale Specificity of Cortical Networks Depends on Inhibitory Cell Type and Connectivity. *Nat. Neurosci.* 8, 1552–1559 (2005)
- Yoshimura, Y., Dantzker, J.K.M., Callaway, E.M.: Excitatory Cortical Neurons from Fine-scale Functional Networks. *Nature* 433, 868–873 (2005)
- Song, S., Sjöström, P.J., Reigl, M., Nelson, S., Chklovskii, D.B.: Highly Nonrandom Features of Synaptic Connectivity in Local Cortical Circuits. *PLoS Biol.* 3, e68 (2005)
- Kalisman, N., Silberberg, G., Markram, H.: The Neocortical Microcircuit as a Tabula Rasa. *Proc. Nat. Acad. Sci.* 102, 880–885 (2005)
- Cossart, R., Aronov, D., Yuste, R.: Attractor Dynamics of Network UP States in the Neocortex. *Nature* 423, 283–288 (2003)
- Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D., Yuste, R.: Synfire Chains and Cortical Songs: Temporal Modules of Cortical Activity. *Science* 304, 559–564 (2004)
- Kitano, K., Fukai, T.: Variability vs Synchronicity of Neuronal Activity in Local Cortical Network Models with Different Wiring Topologies. *J. Comput. Neurosci.* 23, 237–250 (2007)
- Destexhe, A., Pare, D.: Impact of Network activity on the Integrative Properties of Neocortical Pyramidal Neurons in Vivo. *J. Neurophysio.* 81, 1531–1547 (1999)
- Erisir, A., Lai, D., Rudy, B., Leonard, C.S.: Function of Specific K^+ Channels in Sustained High-frequency Firing of Fast-spiking Neocortical Interneurons. *J. Neurophysiol.* 82, 2476–2489 (1999)
- Destexhe, A., Mainen, Z.F., Sejnowski, T.J.: Kinetic Models of Synaptic Transmission. In: Koch, C., Segev, I. (eds.) *Methods in Neural Modeling*, pp. 1–25. MIT, Cambridge (1998)
- Tsodyks, M., Markram, H.: The Neural Code Between Neocortical Pyramidal Neurons Depends on Neurotransmitter Release Probability. *Proc. Nat. Acad. Sci.* 94, 710–723 (1997)

12. Destexhe, A., Rudolph, N., Fellous, J.M., Sejnowski, T.J.: Fluctuating Synaptic Conductances Recreate in Vivo-like Activity in Neocortical Neurons. *Neurosci* 107, 13–24 (2001)
13. Watts, D.J., Strogatz, S.H.: Collective Dynamics of ‘Small-world’ Networks. *Nature* 393, 440–442 (1998)
14. Wang, X.J., Buzsaki, G.: Gamma Oscillation by Synaptic Inhibition in a Hippocampal Interneuronal Network Model. *J. Neurosci.* 16, 6402–6413 (1996)
15. Tsukamoto-Yasui, M., Sasaki, T., Matsumoto, W., Hasegawa, A., Toyoda, T., Usami, A., Kubota, Y., Ochiai, T., Hori, T., Matsuki, N., Ikegaya, Y.: Active hippocampal networks undergo spontaneous synaptic modification. *PLoS ONE* 2, e1250 (2007)
16. Tang, A., Jackson, D., Hobbs, J., Chen, W., Smith, J.L., Patel, H., Prieto, A., Petrusca, D., Grivich, M.I., Sher, A., Hottowy, P., Dabrowski, W., Litke, A.M., Beggs, J.M.: A Maximam Entropy Model Applied to Spatial and Temporal Correlations from Cortical Networks in Vitro. *J. Neurosci.* 28, 505–518 (2008)
17. Nakahara, H., Amari, S.: Information-geometric Measure for Neural Spikes. *Neural Comput.* 14, 2269–2316 (2002)

Prediction of the *O*-Glycosylation by Support Vector Machines and Semi-supervised Learning

Hirotaka Sakamoto¹, Yukiko Nakajima¹, Kazutoshi Sakakibara¹,
Masahiro Ito², and Ikuko Nishikawa¹

¹ Ritsumeikan University, College of Information Science and Engineering
Shiga 525-8577, Japan
nishi@ci.ritsumei.ac.jp

² Ritsumeikan University, College of Life Science

Abstract. Glycosylation is one of the main topics in understanding the life systems. More than a half of the protein is glycosylated to acquire the function, structural stability and biological diversity. *O*-glycosylation is one of the two main types of the mammalian protein glycosylation. Though it is known to serine or threonine specific, any consensus sequence is still unknown, while the binding process and the consensus sequence are clarified for the other type of *N*-glycosylation. We use support vector machines (SVM) for the prediction of *O*-glycosylation sites using the experimental data as the input information such as protein primary sequences, structural and biochemical characters around a prediction target aiming to elucidate the glycosylation mechanism and the existence of any motives. The present paper also reports the results obtained by the semi-supervised learning using transductive SVM considering a possibility of unobserved glycosylation sites, and by the marginalized kernel considering hidden variables.

1 Introduction

Protein glycosylation and its function is one of the main research topics in understanding the life systems [1]. The carbohydrate chains play an important role in various life phenomena by combining to the protein or lipid. Especially in the protein, more than 50% is glycosylated to attain its function, biological diversity, structural stability and so on. Moreover, it is known to affect the onset of serious diseases such as Alzheimer's disease, cancer or respiratory illness.

N-glycosylation and *O*-glycosylation are two main types in the mammalian protein glycosylation. *N*-glycosylation is known as a binding process to an asparagine (Asn) residue in a sequence of Asn-X-Ser/Thr, and its mechanism is clarified including the synthetic pathway of the glycoprotein. However, mucin-type *O*-glycosylation is known to serine (Ser) or threonine (Thr) specific, though any consensus sequence nor the mechanism is still unknown.

On the other hand, the experimental findings in bioinformatics are rapidly growing, and stored on the several public database. Especially, UniProt [2] is the largest database of the protein sequence information, whose data is updated based on the latest experimental reports, and opened to the public via Internet.

In our approach, we use the latest experimental data obtained from the public database, and predict the *O*-glycosylation sites by support vector machines (SVM)[3]. Through the learning from the data, we aim to investigate a condition for the *O*-glycosylation, which might be expressed as a consensus sequence or any motif, and elucidate the mechanisms of the glycosylation. Similar approach has been reported by Julenius et al.[4], who constructed the original *O*-glycosylation database O-Glycbase[5], and used the layered neural network for the prediction of the *O*-glycosylation sites. Li et al.[6] used SVM for the prediction on the protein data obtained from UniProt6.1, and the approach is closer to the present authors'[7][8][9].

The paper is organized as follows. Section 2 describes the data used as the input to SVM. Various lengths of a protein sequence around a prediction target site, its structural information and some biochemical information are used to predict whether each Ser or Thr site is glycosylated. Section 3 compares the prediction efficiency to study which information is relevant to the prediction and therefore to the glycosylation mechanisms. In Section 4, transductive SVM (TSVM) is used for the semi-supervised learning. A site without annotation as *O*-glycosylated in the present database has a possibility to be found glycosylated in a future experiment, therefore it is used as an unlabeled data during the learning. Another approach is a marginalized kernel, which assumes hidden variables for each amino acid residue. Section 5 gives the summary and future works.

2 Protein Data as Input to SVM

2.1 UniProt Database

98 protein sequences of mammalian protein are selected from UniProt12.2[2], which have annotation as *O*-glycosylated from the experimental findings. On 98 sequences, there are 452 sites of Ser or Thr with the annotation (which are denoted positive sites in the following), while the rest 6004 sites of Ser or Thr are without annotation (which are denoted negative sites).

2.2 Amino Acid Sequences

A protein is a high-polymer compound which is composed of 20 kinds of amino acid in a linear chain with a three dimensional structure. Sequence information is the primary information of a protein.

As an input to SVM, a subsequence of length W_s is cut out from an original protein sequence with a prediction target site at the center. For example, a target site and the first and second nearest neighbor amino acid residues on both sides compose a subsequence of $W_s = 5$. Each amino acid residue is encoded by a sparse coding, which uses 21 bits to express 20 kinds of amino acid and a null character for the outside of a sequence terminal.

2.3 Secondary Structure and Solvent Accessibility

The secondary structure is a regular structure observed in a relatively local range of a protein sequence, which is distinguished as a helix, a sheet, a turn and so on.

The solvent accessibility also gives the information on a local protein structure. It is an index of the exposure of each amino acid residue to a solvent. If a residue is more exposed to the protein surface to possess a higher accessibility, then a carbohydrate chain or enzymes may have more chance to access the residue.

Tertiary structure data of proteins are registered in Protein Data Bank(PDB) [10]. Among the above 98 mammalian proteins, 41 proteins are found to be partially registered in PDB with a three dimensional position data of each amino acid residue. 29 proteins shows a perfect consistency of the primary sequences, while the other 12 proteins contain slight inconsistencies of the sequences. Therefore, the best match sequence is searched by a multiple sequence alignment using ClustalW [11] for each of the 12 protein sequences. For other 31 proteins, GTOP [12] is used for the sequence homology search of PDB. All the homologs with the E-value up to e-10 and the sequence identity larger than 30% are jointly used to fully utilize the available information in PDB, by adopting the PDB file with the highest similarity for each residue. Thus, the tertiary structure data in PDB is acquired for 72 proteins in total, and is used to calculate a secondary structure and a solvent accessibility of each amino acid residue using DSSP [13]. DSSP is a software available on Internet, which enables to estimate a secondary structure and an accessibility of each amino acid residue.

However, the structure information registered in PDB is not for a whole sequence even for the above 72 proteins, because of the experimental difficulties. Therefore, as an alternative approach to obtain the secondary structure information, PSIPRED [14] is used to estimate the secondary structure directly from the primary sequence information. The secondary structure of whole sequences is obtained for all 98 proteins by PSIPRED.

In both approaches by PDB and by PSIPRED, the secondary structure is classified into three classes of α -helix, β -sheet and the others, and encoded by the sparse coding of 3 bits.

2.4 Hydropathy

Hydropathy index expresses the hydrophobicity of each amino acid by a real number ranging from -4.5 to 4.5. An amino acid with higher hydrophobicity tends to stay inside the protein. We use an average hydropathy over the length $W_h = 1, \dots, 11$ around a prediction target.

3 Prediction by SVM

3.1 SVM and Parameter Settings

SVM is used to predict whether a target site is glycosylated(positive) or not(negative), as a 2-class classification problem. An advantage of SVM for the present study is that the increase of the input dimension does not lead to the explosion of the learning time or the number of adjustable parameters, as the available experimental data is still limited even in the present database for the learning task. W_s is set to 3, ..., 51 in the followings.

Radial basis function (RBF) given by (II) is used as a kernel in this section

$$K(\mathbf{x}, \mathbf{x}') \propto \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2) , \quad (1)$$

where γ is a kernel parameter. Another parameter is a margin size C . An open software package SVM-light [5] is used, and C is varied from 0.1 to 100, while γ is varied from 1.0×10^{-4} to 1.0×10^0 , to find the best values. All 452 positive sites and the same number of selected negative sites are used, for the learning and testing by 10-fold cross validation(CV).

The evaluation is done by the following quantities. t_p and f_p are the numbers of truly and falsely positive sites, respectively. Similarly, t_n and f_n are the numbers of truly and falsely negative sites, respectively.

Matthews Correlation Coefficient: r

$$r = \frac{t_p t_n - f_p f_n}{\sqrt{(t_n + f_n)(t_n + f_p)(t_p + f_n)(t_p + f_p)}} , \quad (2)$$

which considers the size difference of two class. The numbers of positive and negative data are set equal for the prediction experiment, but the total number of available data is much larger for the negative sites by nature. r takes a real number in $[-1.0, 1.0]$ as an ordinary correlation coefficient.

Accuracy: A_c , Sensitivity: S_n and Specificity: S_p

$$A_c = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} , \quad S_n = \frac{t_p}{t_p + f_n} , \quad S_p = \frac{t_n}{t_n + f_p} , \quad (3)$$

which are the ratios of the correct classification of all, positive and negative sites, respectively.

3.2 Prediction Results

Prediction performances with and without structure information or hydropathy are compared. Fig.II compares Matthews correlation coefficient r obtained by different input information. First, let us focus on the basic result obtained by primary sequence information only. The performance improves with the increase of W_s and saturates around $W_s = 31$. This implies the detailed information of each amino acid residue is effective up to the 15th nearest neighbors of a target site. The r values are not less than in the other published reports [16]. Second, the addition of the secondary structure information estimated by PSIPRED gives some improvement, especially in $W_s \leq 21$. However, the addition of an average hydropathy ($W_h = 9$ in the figure) improves the results for $W_s \leq 11$, but largely reduces for the larger W_s .

Fig.II shows the results using the secondary structure and accessibility obtained from PDB. Because of the limitation of the data registered in PDB as mentioned in section 2.3, the number of proteins is reduced to 72. Even for the 72 proteins, the structure information is available only for their partial sequences. Neither information of the secondary structure nor the accessibility improve the

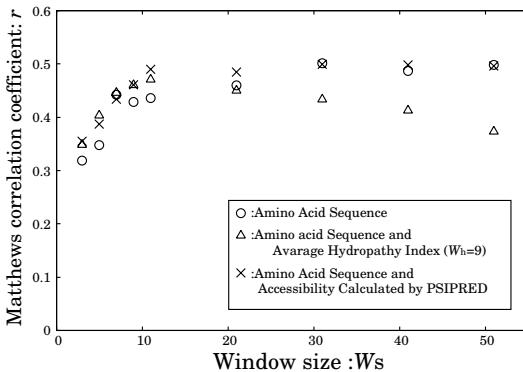


Fig. 1. Matthews correlation coefficient r for various window size W_s obtained by different input information; sequence information only, adding secondary structure estimated by PSIPRED and adding average hydropathy. 10-fold CV for 98 proteins.

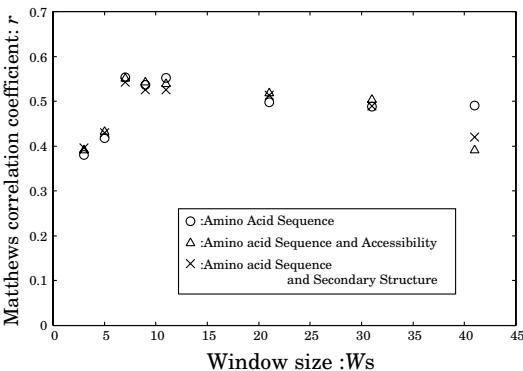


Fig. 2. Matthews correlation coefficient r for various W_s obtained by different input information; sequence information only, adding secondary structure data and accessibility obtained from PDB. 10-fold CV for 72 proteins.

performance from the figure. This difference with the previous data by PSIPRED may be caused by the limitation of the available information. That is, the estimated information by PSIPRED for all proteins gives the better results than the experimentally observed but limited amount of information in PDB. Thus the complete structure information around a target site ($W_s \leq 21$) is expected to improve the prediction.

4 Other Kernels: TSVM and Marginalized Kernel

4.1 Transductive SVM for a Semi-supervised Learning

Transductive SVM. The negative data is a site on which no glycosylation is experimentally observed until today. In other words, there is a possibility

to be found glycosylated in a future experiment. Actually, one of the present research purposes is to predict an unknown *O*-glycosylation site and verify it by the biological experiment. Then, some negative sites will turn out to be positive. Therefore in this section, a part of data which is previously classified to the negative is treated as the unlabeled, that is, the data without neither label of negative nor positive, during the learning. In the following, we removed negative labels from the sites which are always misclassified as positive in the previous experiment. The rest of the negative data and all positive data are still used as the labeled data. This learning with the unlabeled data is called semi-supervised learning, or semi-supervised classification [16].

We use Transductive SVM(TSVM) as an extension of standard SVM with unlabeled data [16]. The classification goal of TSVM is to put either label to the unlabeled data, in a way that a linear boundary has the maximum margin over all originally and newly labeled data. The addition of the unlabeled data is expected to locate the linear boundary more appropriately, if the distribution of the unlabeled data reflects the total distribution of both classes, which is sparse around the boundary in a well-posed classification task.

Unlabeled Data. To decide which data should be unlabeled, we use the results obtained by the original SVM described in the previous section. Namely, as the first step, each 452 sites with and without annotation, are used as labeled as positive and negative data, respectively. SVM is trained with 9 different window size $W_s = 7, 9, 11, 15, 21, 25, 31, 35, 41$, and the number of misclassification is counted for each negative site. Table 1 shows the number of the negative sites which is misclassified as positive for equal or more than M_p times. The misclassified negative with $M_p = 1, \dots, 9$ is used as the unlabeled data \mathcal{U}_{M_p} . The label data \mathcal{L}_{M_p} is the union of the rest negative sites $\mathcal{L}_{M_p}^n$ and 452 positive sites \mathcal{L}^p . The minimum set of labeled data is $\mathcal{L}_{M_p=1}$, and denoted \mathcal{L}_{\min} .

Table 1. Misclassification counts of 452 negative sites over 9 prediction results

M_p	0	1	2	3	4	5	6	7	8	9
Frequency (# \mathcal{U}_{M_p})	452	226	179	153	143	132	112	93	71	54

Prediction on the Labeled Data. The comparison is done by the prediction performance on \mathcal{L}_{\min} with 10-fold CV. The results are compared among (i) Original SVM using all data as labeled (no unlabeled), (ii) TSVM with unlabeled \mathcal{U}_{M_p} and labeled \mathcal{L}_{M_p} for $M_p = 1, 3, 7, 9$, and (iii) SVM using only \mathcal{L}_{\min} (no unlabeled). TSVM has a parameter R_p , a positive ratio for the unlabeled data.

From Fig. 3, TSVM with smaller M_p (larger size of \mathcal{U}) improves more from the original SVM. SVM trained with \mathcal{L}_{\min} is the best, but this is reasonable because the evaluation is done on \mathcal{L}_{\min} . The improvement by TSVM is owing to the increase of sensitivity S_n , which overcomes the decrease of specificity S_n . The linear boundary moves towards the negative region for the larger \mathcal{U} .

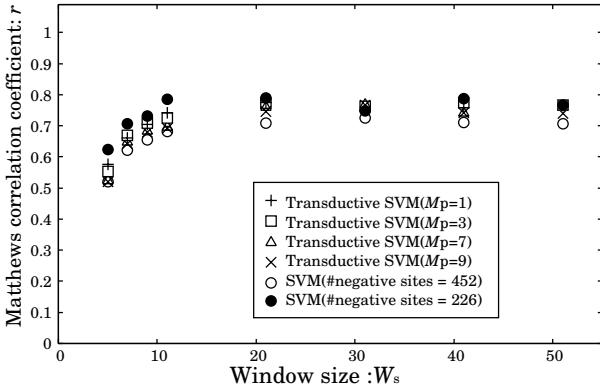


Fig. 3. Matthews correlation coefficient r obtained by TSVM($R_p = 0.5$) and SVM

Prediction on the Unlabeled Data. Let us focus on the minimum set of unlabeled data $\mathcal{U}_{M_p=9}$, and denote \mathcal{U}_{\min} . It is observed that all data in \mathcal{U}_{\min} are always classified as positive in every experiment of TSVM. \mathcal{U}_{\min} may have the highest possibility of an unobserved glycosylation, and more detailed study on any feature of them is now undergoing, before the biological experiments.

4.2 First-Order Marginalized Count Kernel with a Hidden Variable

Marginalized count kernel(MCK) proposed by Tsuda et al.[\[17\]](#) is expected to be effective especially for biological sequence classification. It assumes the data includes any hidden variables.

We use MCK with a hidden variable of three states. Compositions of amino acids including hidden variables are calculated as marginalized counts $\gamma_{kl}(\mathbf{x})$ for protein sequences, and a linear kernel is used for γ_{kl} [\[17\]](#). We also use RBF kernel for γ_{kl} . Fig. 4 compares the results obtained by the followings; (i) the original RBF kernel for a primary sequence information, (ii) RBF kernel for a relative composition of amino acids within W_s , (iii) MCK, which uses a linear kernel for γ_{kl} , and (iv) RBF kernel for γ_{kl} . For smaller window size $W_s < 21$, RBF kernel for γ_{kl} shows the best performance. It is noteworthy that the marginalized counts do not explicitly keep the position information of each amino acid, therefore this result may implies that the amino acid composition (just as counts γ_{kl}) over a certain range near a target site affects the glycosylation. This is consistent with the performance obtained by the relative composition.

Then, which does matter for the glycosylation, the detailed sequence information, or the relative composition? To answer this question, we have a following assumption. It has been known that there are two types of *O*-glycosylation sites; one is densely spaced glycosylated site and the other is sparsely glycosylated sites, and two types are supposed to have different glycosylation mechanisms. We have developed SVM for each type separately, and analyzed each characteristics. As the results, some motives are expected for the sparsely glycosylated

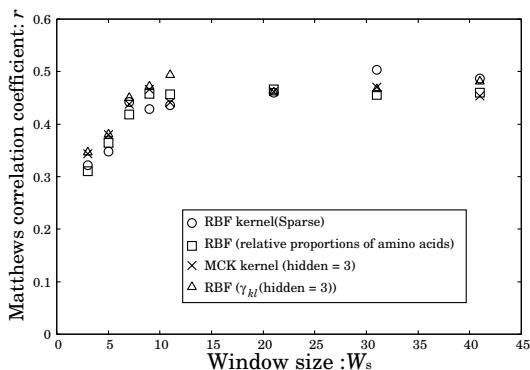


Fig. 4. Matthews correlation coefficient r obtained by MCK and original SVM with RBF kernel for various window size W_s

group, while a composition around a target site seems to affect the densely glycosylated sites. More detailed and precise report will be made in the near future.

5 Summary and Future Works

Several approaches using SVM to predict *O*-glycosylation are reported. The future works include the usage of other genetic information, other kernel methods, and biological experiments to find new glycosylation sites predicted by SVM.

Acknowledgment. This work has been supported in part by the Japanese Ministry of Education, Culture, Sports, Science and Technology, under a Grant-in-Aid for Scientific Research (B) Project No. 19300080.

References

1. Taylor, M.E., Drickamer, K.: Introduction to Glycobiology. Oxford University Press, Oxford (2003)
2. <http://www.ebi.uniprot.org>
3. Cristianini, N., Taylor, J.S.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
4. Julenius, K., Molgaard, A., Gupta, R., Brunak, S.: Prediction, conservation analysis, and structural characterization of mammalian mucin-type *O*-glycosylation sites. *Glycobiology* 15(2), 153–164 (2004)
5. <http://www.cbs.dtu.dk/databases/oglycbase/>
6. Li, S., et al.: Predicting *O*-glycosylation sites in mammalian proteins by using SVMs. *Computational Biology and Chemistry* 30, 203–208 (2006)
7. Nishikawa, I., et al.: Prediction of the *O*-glycosylation sites in protein by layered neural networks and support vector machines. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4252, pp. 953–960. Springer, Heidelberg (2006)

8. Nishikawa, I., et al.: Prediction of the *O*-Glycosylation with Secondary Structure Information by Support Vector Machines. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part II. LNCS (LNAI), vol. 4693, pp. 335–343. Springer, Heidelberg (2007)
9. Sakamoto, H., et al.: Prediction of Mucin-type *O*-glycosylation using Structure Information by Support Vector Machines. In: Proceedings of the 18th Int. Conference on Genome Informatics (December 2007)
10. <http://www.rcsb.org/pdb/home/home.do>
11. <http://www.ebi.ac.uk/Tools/clustalw/>
12. <http://spock.genes.nig.ac.jp/~genome/gtop.html>
13. <http://swift.cmbi.kun.nl/swift/dssp>
14. <http://bioinf.cs.ucl.ac.uk/psipred>
15. <http://svmlight.joachims.org>
16. Zhu, X.: Semi-Supervised Learning Literature Survey, Computer Sciences TR 1530, University of Wisconsin-Madison (2007)
17. Tsuda, K., et al.: Marginalized kernels for biological sequences. Bioinformatics 18(suppl. 1), 268–275 (2002)

Practical Approach to Outlier Detection Using Support Vector Regression

Junya Nishiguchi¹, Chosei Kaseda¹, Hirotaka Nakayama²,
Masao Arakawa³, and Yeboon Yun³

¹ Yamatake Corporation

{nishiguchi-junya, kaseda-chosei}@jp.yamatake.com

² Konan University

nakayama@konan-u.ac.jp

³ Kagawa University

{arakawa, yun}@eng.kagawa-u.ac.jp

Abstract. For precise estimation with soft sensors, it is necessary to remove outliers from the measured raw data before constructing the model. Conventionally, visualization and maximum residual error have been used for outlier detection, but they often fail to detect outliers for nonlinear function with multidimensional input. In this paper we propose a practical approach to outlier detection using Support Vector Regression, which reduces computational cost and defines outlier threshold appropriately. We apply this approach to both test and industrial datasets for validation.

1 Introduction

“Soft sensor” is an inferential model that estimates difficult-to-measure properties by using substitute properties that are easier to measure. They are widely used in industrial applications, such as online analyzers in petrochemical plants and virtual thermometers inside furnace chambers [1].

The most prominent feature of the soft sensor application is to establish nonlinear relationships of output with multidimensional inputs. However, it is usually difficult to describe the physical model precisely. Thus, the model identification approaches with historical data have been used. In the most case, noise and outlier of measured raw data cause declining performance of the resulting model. Therefore, for precise estimation with soft sensors, it is necessary to reduce the effect of noise and outliers.

Noise is generated constantly by the fluctuation of the power supply, pressure, and flow rate. Smoothing (as a pre-processing technique) and robust regression are typical methods of noise reduction.

Outliers are caused by device startup/shutdown, operational error, or transmission error. Since outliers are abnormal phenomena, it is required to remove them from the rest of the data before model construction. Visualization and maximum residual error have been used as conventional methods for detecting outliers. Although visualizing the data structure is useful for systems with one input, it would not be appropriate for

systems with multidimensional input. Also, the maximum residual error approach fails to detect data points that are far from other data points because the estimation function itself is affected by outliers. On the other hand, Robust Partial Least Squares Regression (which is less sensitive to outliers) is also used in the chemometrics application [2], but it can deal only with linear relationships.

Recently an approach using the epsilon Support Vector Regression (ε -SVR) has been proposed for outlier detection in nonlinear functions with multidimensional input [3]. However, this approach is not practical for soft sensors because of high computational costs and difficulty in defining the outlier threshold.

To solve these drawbacks, we propose a practical approach for outlier detection with μ - ε -SVR, an improvement from the original ε -SVR. The efficiency of this approach is validated by both test and industrial datasets.

2 Conventional Approach: ε -SVR Based Outlier Detection

The ε -SVR formula for a given training data set (\mathbf{x}_i, y_i) ($i = 1, \dots, l$) is

$$\begin{aligned} & \underset{\mathbf{w}, b, \xi, \xi'}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{l} \sum_{i=1}^l (\xi_i + \xi'_i) \\ & \text{subject to} \quad \mathbf{w}^T \phi(\mathbf{x}_i) + b - y_i \leq \varepsilon + \xi_i, \\ & \quad y_i - \mathbf{w}^T \phi(\mathbf{x}_i) - b \leq \varepsilon + \xi'_i, \\ & \quad \xi_i, \xi'_i \geq 0, \quad i = 1, \dots, l, \end{aligned} \quad (1)$$

where $\phi(\mathbf{x}_i)$ is a nonlinear map into some feature space, \mathbf{w} and b are the weight vector and bias of the hyperplane on the feature space, respectively; C is the trade-off parameter between complexity and losses, ε is the parameter of the ε -insensitivity loss function, and ξ_i, ξ'_i are slack variables.

When solving the optimization problem, the following dual formation of (1) is derived with the kernel function $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$.

$$\begin{aligned} & \underset{\alpha_i, \alpha'_i}{\text{maximize}} \quad -\frac{1}{2} \sum_{i,j=1}^l (\alpha'_i - \alpha_i)(\alpha'_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^l (\alpha'_i - \alpha_i)y_i - \varepsilon \sum_{i=1}^l (\alpha'_i + \alpha_i) \\ & \text{subject to} \quad \sum_{i=1}^l (\alpha'_i - \alpha_i) = 0, \\ & \quad 0 \leq \alpha_i \leq \frac{C}{l}, \quad 0 \leq \alpha'_i \leq \frac{C}{l}, \quad i = 1, \dots, l. \end{aligned} \quad (2)$$

Note that each Lagrange multiplier α_i (α'_i) has an upper bound constraint C/l .

This approach with ε -SVR utilizes the property of the Lagrange multiplier obtained by solving (2). That is, from Karush-Kuhn-Tucker (KKT) conditions (3) obtained from (1), if a data point doesn't have upper bounded Lagrange multiplier, the corresponding slack variable is zero and consequently the data point is not suspected to be the outlier.

$$\begin{aligned}
& \left\{ -w^T \phi(x_i) - b + y_i + \varepsilon + \xi_i \right\} \alpha_i = 0, \\
& \left\{ w^T \phi(x_i) + b - y_i + \varepsilon + \xi'_i \right\} \alpha'_i = 0, \\
& \xi_i \left(\frac{C}{l} - \alpha_i \right) = 0, \\
& \xi'_i \left(\frac{C}{l} - \alpha'_i \right) = 0.
\end{aligned} \tag{3}$$

Therefore, the data points with Lagrange multipliers α_i (α'_i) at their upper bounds can be considered as candidate outliers. Generally, since multiple data points have Lagrange multipliers with upper bounds, it is required to find a real outlier among several candidates. The outlier is detected when one of the candidates contains the highest frequency of the suspected outlier, which is calculated from the optimization problem (2) over several different parameters ε (which means different tolerance error). This process is repeated until no more outliers are detected or, the Root Mean Square error over several values of ε is less than the outlier threshold defined by trial and error.

The following problems arise when applying this approach with soft sensors: Firstly, it requires high computational costs for large amounts of data with multiple outliers because detection of an outlier requires several iterations of the optimization calculation. Secondly, since it is not clear how to define the outlier threshold value, trial and error is required for accurate detection.

3 Proposed Approach: μ - ε -SVR Based Outlier Detection

In order to improve the generalization ability, Nakayama et.al. [4] have proposed a SVR formula minimizing the maximum slack variable with parameter μ instead of the average slack variables. In this paper, we utilize parameter μ to overcome difficulties of conventional approaches and have thus developed the μ - ε -SVR as follows.

$$\begin{aligned}
& \underset{\mathbf{w}, b, \xi, \xi'}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \mu(\xi + \xi') \\
& \text{subject to} \quad \mathbf{w}^T \phi(\mathbf{x}_i) + b - y_i \leq \varepsilon + \xi, \\
& \quad y_i - \mathbf{w}^T \phi(\mathbf{x}_i) - b \leq \varepsilon + \xi', \\
& \quad \xi, \xi' \geq 0, \quad i = 1, \dots, l.
\end{aligned} \tag{4}$$

Dual formation of (4) is

$$\begin{aligned}
& \underset{\alpha_i, \alpha'_i}{\text{maximize}} \quad -\frac{1}{2} \sum_{i,j=1}^l (\alpha'_i - \alpha_i)(\alpha'_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^l (\alpha'_i - \alpha_i)y_i - \varepsilon \sum_{i=1}^l (\alpha'_i + \alpha_i) \\
& \text{subject to} \quad \sum_{i=1}^l (\alpha'_i - \alpha_i) = 0, \\
& \quad \sum_{i=1}^l \alpha_i \leq \mu, \quad \sum_{i=1}^l \alpha'_i \leq \mu, \\
& \quad 0 \leq \alpha_i, \quad 0 \leq \alpha'_i, \quad i = 1, \dots, l,
\end{aligned} \tag{5}$$

where insensitivity parameter ε and regularization parameter μ are defined in advance.

In the μ - ε -SVR primal problem (4), only slack variables of maximum error data points are evaluated, not the average slack variables used in ε -SVR (1). The complementary slackness condition in KKT conditions for (4) is given by

$$\begin{aligned} \left\{ -w^T \phi(x_i) - b + y_i + \varepsilon + \xi \right\} \alpha_i &= 0, \\ \left\{ w^T \phi(x_i) + b - y_i + \varepsilon + \xi \right\} \alpha'_i &= 0, \\ \xi \left(\mu - \sum_{i=1}^l \alpha_i \right) &= 0, \\ \xi \left(\mu - \sum_{i=1}^l \alpha'_i \right) &= 0. \end{aligned} \quad (6)$$

When a sum of the Lagrange multiplier reaches upper bound μ , all data points with positive (non-zero) Lagrange multiplier have the same maximum error. Therefore, these data points are likely to be the real outlier data. In optimization theory it is a well-known fact that the Lagrange multipliers indicate sensitivity of corresponding inequality constraints to the objective function [5]. Consequently, when an outlier exists, the data point with a maximum Lagrange multiplier is considered to be the most probable outlier, which is farthest from the other data. Note that, from dual problem (5), since each Lagrange multiplier isn't constrained by an upper bound, multiple data points never have the same upper bounded Lagrange multiplier.

On the other hand, when the sum of the Lagrange multipliers is under the upper bound μ , all data have an error less than the tolerance ε , or outlier does not exist. Thus, the sum of Lagrange multipliers is utilized for termination condition of the outlier detection process.

The following pseudo-code of the outlier detection procedure utilizes the features mentioned above.

Algorithm for Outlier Detection

```

Step1: Calculate  $\mu$ - $\varepsilon$ -SVR
Step2: Find the largest  $\alpha_i$ ,  $\alpha'_{i'}$ 
Step3: If  $\text{sum}(\alpha_i) = \mu$  or  $\text{sum}(\alpha'_{i'}) = \mu$ 
        Remove  $x(i)$  and  $y(i)$  from data set
        Return to Step2 using the reduced data set
    Else
        Stop
    End

```

While detecting outliers using this approach, each iteration requires only one optimization calculation. Therefore computational costs become lower than those of the conventional ε -SVR approach. Also, outlier threshold can be defined appropriately using fixed tolerance ε so that outliers are separated from other noisy data.

4 Validation with Test Data

In order to demonstrate the effectiveness of the proposed outlier detection approach, we first consider the following example problem:

$$y_i = (\sin 2\pi x_i)^2 + \eta_i + \theta_i \quad (7)$$

where x_i ($i = 1, \dots, 100$) is sampled from uniform distribution on [0,1], η_i is noise generated by normal distribution $N(0,0.05)$, and θ_i represents three outliers, that is, $\theta_i = [0.5, -0.5, 0.5, 0, \dots, 0]$

We use Gaussian kernel (8) with $r=0.05$, $\mu=1000$ and $\varepsilon=0.15$ as the parameter for this problem. Note that ε is defined as a tolerance based on 3σ of the noise distribution.

$$K(\mathbf{x}, \mathbf{x}'_i) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}'_i\|^2 / 2r^2\right) \quad (8)$$

Figure 1 shows the estimated result and residual error at each iteration. From this figure, we can see that after four iterations all three outliers are removed correctly. As the number of iterations increase, the accuracy of the estimated result improves. The maximum residual error of the final result is equal to tolerance 0.15 as given above.

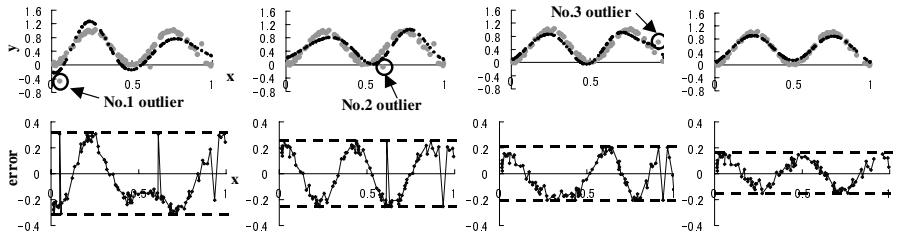


Fig. 1. Top left to right: actual data (gray dots) and estimated data (black dots) at each iteration. Bottom left to right: residual error between actual and estimated data at each iteration.

Another example is the stack loss data [6], which have been discussed in many previous studies. Stack loss data describes the operation of a plant for the oxidation of ammonia to nitric acid. This data consists of 21 samples, measured once a day. The input variables are air flow, water temperature, and acid concentration. The output variable is stack loss. In previous research, No.1, 3, 4, and 21 are considered to be outliers [7].

The proposed approach was applied for stack loss data with the following parameters: $\mu=1000$, and $\varepsilon=2.0$. As kernel function, the linear kernel (as shown by (9)) was used because many previous researches have succeeded in outlier detection with the linear approaches against the stack loss data.

$$K(\mathbf{x}_i, \mathbf{x}'_i) = \mathbf{x}_i \cdot \mathbf{x}'_i + 1 \quad (9)$$

As seen in Table 1, the proposed approach succeeded in removing each outlier after each iteration, although the conventional approach with ε -SVR required a greater number of iterations and more computation time.

From these results for the two test data sets, the proposed approach appears to remove the outliers from the nonlinear and multidimensional data correctly while taking fewer iterations than the conventional approach.

Table 1. Comparison between proposed and conventional result for stack loss data

Proposed approach (μ - ε -SVR)		Conventional approach (ε -SVR)	
Iteration count	Removed data No.	Iteration count	Removed data No.
1	No.21	20	No.3, No.4, No.21
1	No.4	20	No.1
1	No.3	20	-
1	No.1		
1	-		
Total iteration count: 5 iterations		Total iteration count: 60 iterations	
Total computation time: 0.9 sec.		Total computation time: 9.1 sec.	

5 Application for Soft Sensor

Next, we will show the efficiency of the proposed approach by implementing it with an actual soft sensor used for an air conditioning system in office building. The energy consumption in the air conditioning system deviates largely due to climate change. In order to measure the amount of energy saved after a system improves (such as a new facility installation or system maintenance) it is necessary to compare the energy consumption before and after the improvement under the same climate conditions. Therefore, the amount of the energy saved is calculated by the difference between post-improvement energy consumption and simulated pre-improvement energy consumption. The simulated energy consumption is calculated with the soft sensor where input variables represent climate conditions.

The data was obtained from an office building from April 1, 2007 to March 31, 2008 measured once a day, and consisted of 366 observations. The input variables are outside air temperature, outside air humidity, and workday/holiday indicator. Since the data contains results from abnormal operations such as power outage, these abnormal data points were removed as outliers with the proposed approach.

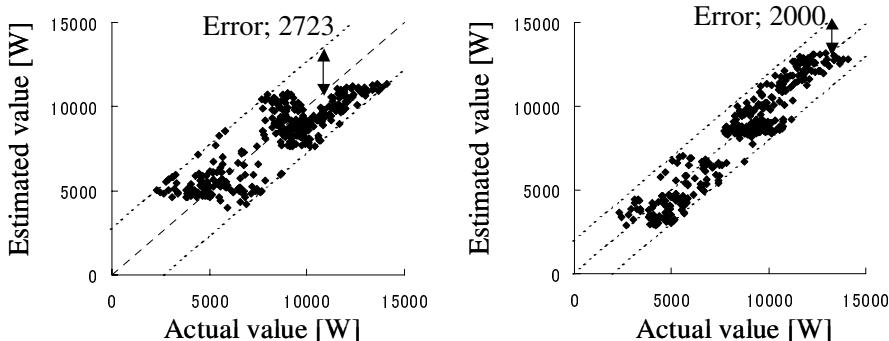
As the calculation parameters, $\mu=1000000$ and $\varepsilon=2000$ were used. Note that, ε was defined as 5% of the output range. In addition, we chose the second order polynomial kernel (10) as the kernel function by the fact that the energy consumption is largely affected by the square of the outside air temperature.

$$K(\mathbf{x}_i, \mathbf{x}'_i) = (\mathbf{x}_i \cdot \mathbf{x}'_i + 1)^2 \quad (10)$$

After six iterations, five outliers were removed from the data. Table 2 represents the removed outliers and the corresponding information from the daily report. No.294 and No.357 represent data measured during Holiday working days, and the energy consumption turned out to be larger than that of other holidays. On the other hand, the energy consumption of No.134 was relatively small because of the power outage in this office building. No.265 and No.269 were measured on the day of the control system failure. Figure 2 shows the estimated result after the removal of all outliers was accurate where the error was smaller than the tolerance ε . The result shows that the proposed approach removed the outliers effectively for the real data.

Table 2. Outliers detected in energy consumption data

Removed data No.	Date	Daily report information
No. 294, 357	Jan. 19, Mar. 22	Holiday working days
No. 134	Aug. 12	Power outage
No. 265, 269	Dec. 21, 25	System failure

**Fig. 2.** Left: Actual and estimated energy consumption before removing. Right: Actual and estimated energy consumption after removing all outliers.

6 Conclusion

In this paper we proposed a practical approach for outlier detection using μ - ϵ -SVR. This approach is advantageous over the conventional approach because it reduces computation cost and outlier thresholds can be defined more accurately. The efficiency of the proposed approach was tested on both test data sets and an industrial data set.

References

- Lin, B., Recke, B., Knudsen, J., Jørgensen, S.: A systematic approach for soft sensor development. *Computers and Chemical Engineering* 31, 419–425 (2007)
- Pell, R.J.: Multiple outlier detection for multivariate calibration using robust statistical techniques. *Chemometrics and Intelligent Laboratory Systems* 52, 87–104 (2000)
- Jordaan, E.M., Smits, G.F.: Robust Outlier Detection using SVM Regression. In: Proceeding. 2004 IEEE International Joint Conference on Neural Network (2004)
- Nakayama, H., Yun, Y.: Support Vector Regression Based on Goal Programming and Multi-objective Programming. In: IJCNN 2006, Neural Networks (2006)
- Mangasarian, O.L.: Nonlinear Programming. McGraw-Hill, New York (1969)
- Brownlee, K.A.: Statistical Theory and Methodology in Science and Engineering, pp. 491–500. Wiley, New York (1960)
- Rousseeuw, P.J., Baxter, M.A.: Robust Regression and Outlier Detection. John Wiley & Sons, Inc., New York (1987)

A Variant of Adaptive Mean Shift-Based Clustering

Fajie Li¹ and Reinhard Klette²

¹ University of Groningen, The Netherlands

² The University of Auckland, New Zealand

Abstract. This paper proposes a special adaptive mean shift clustering algorithm, especially for the case of highly overlapping clusters. Its application is demonstrated for simulated data, aiming at finding the ‘old clusters’. The obtained clustering result is actually close to an estimated upper bound, derived for those simulated data elsewhere.

1 Introduction

There are hundreds of clustering algorithms proposed in the literature; for lists of examples, see page 5 in [8], page 13 in [10] or page 130 in [13].

Clustering not only interests scientists in computing but also, for example, astronomers [3][6][9]. Given is an observed set of stars (considered to be a set of points); how to find (*recover*) clusters which are the contributing galaxies to the observed union of those clusters?

This paper discusses clustering at a general level. We illustrate by using one (fairly complex) data set of simulated astronomical data [1]. Clusters in those data are typically characterized by being highly overlapping. Obviously, the recovery of highly overlapping data is difficult, if not even (nearly) impossible.

Currently published cluster algorithms (see, for example, [4]) work neither efficiently nor correctly on such data, and upper bounds for recovery rates would allow to identify realistic goals. See [11] for the definition of recovery rate as used in this paper, and [12] for the estimation of a best possible upper bound for this recovery rate, illustrated there for the same simulated astronomical data example as also used in this paper.

This paper proposes a variant of adaptive mean shift clustering, and we illustrate it for finding old clusters in the mentioned simulated data. We compute the recovery rate of our proposed algorithm; we show that this is close to the estimated lowest possible upper bound in [12] of recovery rates for the used example of simulated data.

1.1 Adaptive Mean Shift Procedure

This subsection explains the principle of adaptive mean shift-based clustering, meaning that the data points always move to local density maxima.

¹ See www.astro.rug.nl/~ahelmi/simulations_gaia.tar.gz and Section 4.1 in [12].

We review some results of [2]. Let $k(x)$ be a symmetric univariate kernel function, where $x \in (-\infty, +\infty)$. We can construct a dD kernel function from $k(x)$ as follows:

$$K(\mathbf{x}) = c_{k,d} k(\|\mathbf{x}\|^2) \quad (1)$$

where

$$c_{k,d} = \frac{\int_{R^d} K(\mathbf{x}) d\mathbf{x}}{\int_{R^d} k(\|\mathbf{x}\|^2) d\mathbf{x}} = \frac{1}{\int_{R^d} k(\|\mathbf{x}\|^2) d\mathbf{x}} > 0$$

Function $k(x)$, where $x \in [0, +\infty)$, is called the *profile* of kernel $K(\mathbf{x})$.

By Equation (1), the kernel density estimator can be rewritten as follows:

$$\hat{f}_{h,K}(\mathbf{x}) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{\mathbf{x} - \mathbf{X}_i}{h}\right\|^2\right) \quad (2)$$

Suppose that $k(x)$ is differentiable in $[0, +\infty)$, except for a finite number of points. Let

$$g(x) = -k'(x)$$

where $x \in [0, +\infty)$, except for a finite number of points.

Construct a kernel function from $g(x)$ as follows:

$$G(\mathbf{x}) = c_{g,d} g(\|\mathbf{x}\|^2)$$

where

$$c_{g,d} = \frac{\int_{R^d} G(\mathbf{x}) d\mathbf{x}}{\int_{R^d} g(\|\mathbf{x}\|^2) d\mathbf{x}} = \frac{1}{\int_{R^d} g(\|\mathbf{x}\|^2) d\mathbf{x}} > 0$$

Denote the gradient of the density estimator of $\hat{f}_{h,K}(\mathbf{x})$ by $\nabla \hat{f}_{h,K}(\mathbf{x})$. Furthermore [see [2] for the details], let

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{1}{2} h^2 \frac{c_{g,d}}{c_{k,d}} \times \frac{\nabla \hat{f}_{h,K}(\mathbf{x})}{\hat{f}_{h,G}(\mathbf{x})} \quad (3)$$

where

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x} - \mathbf{X}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x} - \mathbf{X}_i}{h}\right\|^2\right)} - \mathbf{x} \quad (4)$$

Equation (4) is the difference between the weighted mean and \mathbf{x} , known as *mean shift vector*. Since the gradient of the density estimator always points towards that direction in which the density rises most quickly, by Equation (1), the mean shift vector always points towards the direction in which the density rises most quickly. This is the main principle of mean shift-based clustering.

[1] generalized the kernel estimator such that each data point \mathbf{x}_i is associated with a bandwidth value h_i , where $i = 1, 2, \dots, n$:

$$\hat{f}_K(\mathbf{x}) = \frac{c_{k,d}}{n} \sum_{i=1}^n \frac{1}{h_i^d} k\left(\left\|\frac{\mathbf{x} - \mathbf{X}_i}{h_i}\right\|^2\right) \quad (5)$$

Consequently, Equation (4) was generalized to

$$\mathbf{m}_G(\mathbf{x}) = \frac{\sum_{i=1}^n \frac{1}{h_i^{d+2}} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h_i}\right\|^2\right)}{\sum_{i=1}^n \frac{1}{h_i^{d+2}} g\left(\left\|\frac{\mathbf{x}-\mathbf{X}_i}{h_i}\right\|^2\right)} - \mathbf{x} \quad (6)$$

which is called the *adaptive* mean shift vector. Analogously, the adaptive mean shift vector also always points towards the direction in which the density rises most quickly, which is called the *mean shift property*. This is the basic principle of adaptive mean shift-based clustering. $-\|\cdot\|_1$ is the L_1 norm.

Procedure 1. Adaptive Mean Shift Procedure

Input: A positive integer k , n data points \mathbf{x}_i , where $i = 1, 2, \dots, n$, one of the data points, \mathbf{y}_1 , and a total number N of iterations.

Output: An approximate local maximum (mode) of the density.

- 1: $h_i = \|\mathbf{x}_i - \mathbf{x}_{i,k}\|_1$, where $\mathbf{x}_{i,k}$ is the k -nearest neighbor of \mathbf{x}_i
- 2: $j = 1$
- 3: **while** $j < N$ **do**
- 4:
$$\mathbf{y}_2 = \frac{\sum_{i=1}^n \frac{1}{h_i^{d+2}} \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}_1 - \mathbf{x}_i}{h_i}\right\|^2\right)}{\sum_{i=1}^n \frac{1}{h_i^{d+2}} g\left(\left\|\frac{\mathbf{y}_1 - \mathbf{x}_i}{h_i}\right\|^2\right)} \quad (7)$$
- 5: $\mathbf{y}_1 = \mathbf{y}_2$
- 6: $j = j + 1$
- 7: **end while**
- 8: Output \mathbf{y}_2

If h_i is fixed in Procedure 1, then this is called the *Mean Shift Procedure*, implementing the main principle of mean shift-based clustering.

1.2 Locality-Sensitive Hashing

The main computation of the adaptive mean shift procedure is in Step 4 [Equation (7)] which has time complexity $\mathcal{O}(n^2 d N)$, where n is the number of input points, d the dimension of the input points, and N the number of iterations [4]. An efficient approximate computation of Step 4 can be achieved by solving the problem of nearest neighbor search [5]: Consider points \mathbf{x}_i which are nearest to \mathbf{y}_1 to test whether the kernel of \mathbf{x}_i “covers” \mathbf{y}_1 . In other words, we have that $g\left(\left\|\frac{\mathbf{y}_1 - \mathbf{x}_i}{h_i}\right\|^2\right) \neq 0$ in Equation (7). Also, considering naive estimators, we can see that, the larger the distance between point x and \mathbf{X}_i , the more likely that the value of $w(x)$ equals 0. [4] applied an approximate nearest neighbor search technique called *locality-sensitive hashing* (LSH) (see [5] and [7]) to efficiently improve the neighborhood queries.

The main ideas of LSH are represented by the following two procedures. Procedure 2 is used to partition the given data points into $K/d + 1$ regions such that the points in each region have the same KD boolean vector. It is applied in Steps 2 and 5 of the Locality-Sensitive Hashing procedure (Procedure 3).

Procedure 2. *KD(i.e., K-dimensional) Boolean Vector*

Input: A positive integer K , n data points \mathbf{x}_i , where $i = 1, 2, \dots, n$, and a point $\mathbf{q} = (q_1, q_2, \dots, q_n)$.

Output: A KD boolean vector, denoted by (b_1, b_2, \dots, b_K) .

```

1:  $i = 1$ 
2: Randomly take an integer  $d_k \in \{1, 2, \dots, d\}$ 
3: Randomly take a real number  $v_k \in [min_{x_{d_k}}, max_{x_{d_k}}]$ , where
    $min_{x_{d_k}} = \min\{x_{i_{d_k}} : x_{i_{d_k}} \text{ is } d_k - \text{th coordinate of } \mathbf{x}_i \wedge i \in \{1, 2, \dots, n\}\}$ 
    $max_{x_{d_k}} = \max\{x_{i_{d_k}} : x_{i_{d_k}} \text{ is } d_k - \text{th coordinate of } \mathbf{x}_i \wedge i \in \{1, 2, \dots, n\}\}$ 

4: if  $q_{d_k} \leq v_k$  then
5:    $b_i = \text{true}$ 
6: else
7:    $b_i = \text{false}$ 
8: end if
9: if  $i < K$  then
10:    $i = i + 1$  and goto Step 1
11: end if
12: Output  $(b_1, b_2, \dots, b_K)$ 
```

Procedure 3. *Locality-Sensitive Hashing*

Input: Two positive integers K and L , n data points \mathbf{x}_i , where $i = 1, 2, \dots, n$, and a query point $\mathbf{q} = (q_1, q_2, \dots, q_n)$.

Output: Union (i.e., set) of points in an approximate neighborhood of \mathbf{q} .

```

1: for  $l \in \{1, 2, \dots, L\}$  do
2:   Apply Procedure 2 to compute a  $KD$  vector for  $\mathbf{q}$ , denoted by  $\mathbf{v}_q$ .
3:    $C_l = \emptyset$ 
4:   for  $i \in \{1, 2, \dots, n\}$  do
5:     Apply Procedure 2 to compute a  $KD$  vector for  $\mathbf{x}_i$ , denoted by  $\mathbf{v}_{x_i}$ 
6:     if  $\mathbf{v}_q = \mathbf{v}_{x_i}$  then
7:        $C_l = C_l \cup \{\mathbf{x}_i\}$ 
8:     end if
9:   end for
10: end for
11: Output  $\cup_{l=1}^L C_l$ 
```

C_l in this Procedure 3 may be implemented by a hash table.² – For a given K , the larger the value of L , the larger the set of points in an approximate neighborhood of \mathbf{q} . Thus, for a given radius h , there exists an optimal L such that Procedure 3 outputs an optimal approximate neighborhood of \mathbf{q} with radius h . If L is larger than the optimal value, Procedure 3 will run for more time but will not improve the output. [4] presented a learning method to select optimal values of K and L .

² See C++ source code at <http://www.caip.rutgers.edu/riul>

2 Algorithm

Based on a locality-sensitive hashing technique, [4] derived a fast adaptive mean shift-based clustering algorithm (Algorithm 1) which is applied in Step 3 of our main algorithm below.

Algorithm 1. Fast Adaptive Mean Shift [4]

Input: A positive integer k , n data point \mathbf{x}_i , where $i = 1, 2, \dots, n$, one of the data points, \mathbf{y}_1 , and a total number N of iterations.

Output: An approximate local maximum (mode) of the density.

- 1: We use a subset of the input data for learning. {For selecting optimal values of K and L , see [4].}
- 2: $h_i = \|\mathbf{x}_i - \mathbf{x}_{i,k}\|_1$, where $\mathbf{x}_{i,k}$ is the k -nearest neighbor of \mathbf{x}_i
- 3: $j = 1$
- 4: **while** $j < N$ **do**
- 5: Let K, L, n data points \mathbf{x}_i and \mathbf{y}_1 be the input for Procedure 3; compute an approximate neighbor of \mathbf{y}_1 with radius h_i , denoted by $U_{\mathbf{y}_1}$. Let

$$\mathbf{y}_2 = \frac{\sum_{\mathbf{x}_i \in U_{\mathbf{y}_1}} \frac{1}{h_i^{d+2}} \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}_1 - \mathbf{x}_i}{h_i}\right\|^2\right)}{\sum_{\mathbf{x}_i \in U_{\mathbf{y}_1}} \frac{1}{h_i^{d+2}} g\left(\left\|\frac{\mathbf{y}_1 - \mathbf{x}_i}{h_i}\right\|^2\right)} \quad (8)$$

- 6: $\mathbf{y}_1 = \mathbf{y}_2$
- 7: $j = j + 1$
- 8: **end while**
- 9: Output \mathbf{y}_2

The main difference between Algorithm 1 and Procedure 1 is defined by Equations 8 and Equations 7, having the impact that Algorithm 1 is faster (however, also “more approximate” only).

Algorithm 2. Our Main Algorithm

Input: Three positive integers k, N (number of iterations) and T (threshold of the number of merged points to apply one of the traditional clustering algorithms, such as kmeans or clusterdata implemented in MATLAB), n old clusters C_i , where $i = 1, 2, \dots, n$.

Output: m new clusters G_i , where $i = 1, 2, \dots, m$.

- 1: $C = \bigcup_{i=1}^n C_i$ and $S = \emptyset$
- 2: **for** For each $\mathbf{x} \in C$ **do**
- 3: Let k, C, \mathbf{x} and N be the input for Algorithm 1; compute an approximate local maximum of the density, denoted by \mathbf{x}' .
- 4: $S = S \cup \{\mathbf{x}'\}$
- 5: **end for**
- 6: Sort S according to lexicographic order.

- 7: Merge duplicated points in S into a single point. Denote the resulting set by S' . {Note that each point $\mathbf{x}' \in S'$ can be associated with a set, called *associated* set of \mathbf{x}' , denoted by $S'_{\mathbf{x}'}$, to store the original points in C such that they are mapped, using Equation (8), to the same point \mathbf{x}' .}
- 8: **if** $|S'| > T$ **then**
- 9: $C = S'$ and goto Step 2
- 10: **end if**
- 11: Sort S' according to the cardinalities of associated sets of points in S' .
- 12: Let the last m points in S' be the initial centers, apply *kmeans* to cluster S' ; resulting (new) clusters are denoted by G'_i , where $i = 1, 2, \dots, m$.
- 13: **for** For each $i \in \{1, 2, \dots, m\}$ **do**
- 14: Output

$$G_i = (\cup_{\mathbf{x}' \in G'_i} S'_{\mathbf{x}'}) \cup \{\mathbf{x}'\}$$

15: **end for**

In Step 12 of Algorithm 2, we may replace *kmeans* by *clusterdata*. Step 11 becomes unnecessary in this case.

Table 1. Approximate recovery rates and pseudo recovery rates of Algorithm 2 for the used example of simulated astronomical data and different values of k

k	K	L	$(k_1, p_1)(\%)$	$(k_2, p_2)(\%)$	$(c_1, p_1)(\%)$	$(c_2, p_2)(\%)$
5	28	2	(29.02, 93.94)	(23.66, 93.94)	(23.15, 69.70)	(18.19, 78.79)
6	27	2	(26.49, 93.94)		(18.47, 60.61)	
7	30	3	(26.49, 93.94)	(30.02, 90.91)	(19.47, 60.61)	(30.69, 84.85)
8	30	3	(26.49, 93.94)	(30.02, 93.94)	(19.47, 60.61)	(30.69, 84.85)
9	30	3	(28.67, 93.94)	(33.71, 90.91)	(20.71, 57.58)	(26.75, 78.79)
10	30	3	(27.43, 90.91)	(35.16, 81.82)	(22.56, 63.64)	(35.32, 84.85)
11	30	3	(27.25, 87.88)	(33.77, 84.85)	(18.51, 60.61)	(30.25, 75.76)
12	28	3	(28.55, 93.94)	(34.22, 87.88)	(21.20, 69.70)	(32.04, 81.82)
13	28	3	(28.55, 93.94)	(34.22, 87.88)	(21.20, 69.70)	(32.04, 81.82)
14	28	3	(28.55, 93.94)		(21.20, 69.70)	
15	28	3	(28.79, 90.91)	(34.87, 81.82)	(25.39, 66.67)	(33.68, 75.76)
16	30	3	(27.21, 84.85)	(29.98, 87.88)	(22.03, 63.64)	(26.00, 75.76)
17	28	3	(27.74, 93.94)		(24.05, 66.67)	
18	28	3	(30.43, 93.94)		(24.64, 69.70)	
20	27	4	(28.50, 93.94)	(31.82, 78.79)	(24.84, 75.76)	(32.53, 78.79)
25	29	3	(27.47, 84.85)	(27.85, 75.76)	(15.89, 60.61)	(28.54, 75.76)
30	30	5	(26.30, 81.82)	(37.84, 93.94)	(14.95, 54.55)	(31.28, 81.82)
40	26	5	(26.30, 81.82)		(14.95, 54.55)	
50	25	4	(29.66, 96.97)		(28.73, 69.70)	
70	28	7	(30.31, 93.94)		(26.42, 69.70)	
90	29	7	(28.41, 93.94)		(24.80, 66.67)	
110	22	6	(28.67, 81.82)		(28.13, 72.73)	
130	28	8	(24.94, 81.82)		(28.84, 6970)	
150	27	9	(26.59, 93.94)		(26.72, 69.70)	

Table 2. Approximate recovery rates and pseudo recovery rates of Algorithm 2 for the used example of simulated astronomical data and $k = 30$

j-th run	($k_1\%$, $p_1\%$)	($k_2\%$, $p_2\%$)	($c_1\%$, $p_1\%$)	($c_2\%$, $p_2\%$)
1	(28.01, 96.97)	(38.94, 81.82)	(19.19, 63.64)	(39.59, 75.76)
2	(27.66, 90.91)	(38.25, 81.82)	(15.82, 60.61)	(39.27, 78.79)
3	(26.74, 90.91)	(33.71, 84.85)	(16.84, 57.58)	(35.82, 81.82)
4	(26.18, 87.88)	(33.49, 75.76)	(25.13, 66.67)	(33.49, 75.76)
5	(28.66, 96.97)	(30.89, 81.82)	(21.48, 75.76)	(34.62, 75.76)
6	(27.28, 78.79)	(37.87, 84.85)	(18.29, 54.55)	(33.95, 75.76)
7	(24.55, 93.94)	(33.34, 75.76)	(22.50, 63.64)	(34.43, 75.76)
8	(27.24, 87.88)	(36.03, 69.70)	(22.29, 66.67)	(37.72, 72.73)
9	(26.64, 90.91)	(35.66, 72.73)	(22.86, 63.64)	(34.74, 69.70)
10	(28.47, 90.91)	(36.29, 81.82)	(21.95, 63.64)	(33.71, 75.76)
mean	(27.14, 90.61)	(35.45, 79.09)	(20.64, 63.64)	(35.73, 75.76)

3 Experimental Results

Table 1 lists approximate recovery rates and pseudo recovery rates (as defined in [11]) of Algorithm 2 when applied to the previously specified sample of synthetic data (analyzed in [12] with respect to the best possible upper bound for the recovery rate). In this table, k is one of the input parameters of Algorithm 2. K and L are optimal parameters obtained in Step 1 of Algorithm 1 which is applied in Step 3 of Algorithm 2. For the used data example we used $T = 800$ in Algorithm 2. All the experiments have at most two iterations (i.e., $N \leq 2$). k_i , c_i and p_i are short for the approximate recovery rates of *kmeans*, *clusterdata*, and the pseudo recovery rates at iteration i , where $i = 1, 2$.

Table 1 shows that Algorithm 2 produces the best recovery rate if $k = 30$.

Table 2 shows approximate recovery rates and pseudo recovery rates of Algorithm 2 for $k = 30$, $K = 30$, and $L = 5$. For the selected example of input data, we used $T = 800$ in Algorithm 2. All the experiments have at most two iterations (i.e., $N \leq 2$). k_i , c_i and p_i are short for the approximate recovery rates of *kmeans*, *clusterdata*, and pseudo recovery rates at iteration i , for $i = 1, 2$.

Table 2 shows that Algorithm 2 even ensures a mean recovery rate of 35.45% (using *kmeans*) or of 35.73% (using *clusterdata*).

The best possible upper bound, estimated in Subsection 4.2 in [12] for this data set, is between 39.68% and 44.71%. This allows to conclude that the obtained mean recovery rate is a “fairly good result”.

4 Conclusion

We proposed a variant of adaptive mean shift clustering. Its efficiency was illustrated by applying it for the recovery of old clusters of a fairly complex set of simulated data. The obtained recovery rate is close to the estimated best possible upper bound of recovery rates for this data set, as papered in [12].

Acknowledgement. This research is part of the project “Astrovis”, research program STARE (STAR E-Science), funded by the Dutch National Science Foundation (NWO), project no. 643.200.501. The first author thanks project members for valuable comments on an earlier draft of this paper.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: The variable bandwidth mean shift and data-driven scale selection. In: Proc. Intl. Conf. Computer Vision, July 2001, vol. I, pp. 438–445 (2001)
2. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach toward Feature Space Analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24, 603–619 (2002)
3. Efstathiou, G., Frenk, C.S., White, S.D.M., Davis, M.: Gravitational clustering from scale-free initial conditions. Monthly Notices RAS 235, 715–748 (1988)
4. Georgescu, B., Shimshoni, I., Meer, P.: Mean shift Bbased clustering in high dimensions: a texture classification example. In: Proc. IEEE Int. Conf. Computer Vision (2003)
5. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions vis hashing. In: Proc. Int. Conf. Very Large Data Bases, pp. 518–529 (1999)
6. Helmi, A., de Zeeuw, P.T.: Mapping the substructure in the Galactic halo with the next generation of astrometric satellites. Astron. Soc. 319, 657–665 (2000)
7. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proc. Symp. Theory Computing, pp. 604–613 (1998)
8. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Computing Surveys 31(3), 264–323 (1999)
9. Knebe, A., Gill, S.P.D., Kawata, D., Gibson, B.K.: Mapping substructures in dark matter haloes. Astron. Soc. 357, 35–39 (2005)
10. Law, H.C.: Clustering, Dimensionality Reduction, and Side Information. Ph.D. Thesis, Michigan State University, the United States (2006)
11. Li, F., Klette, R.: Recovery rate of clustering algorithms. Technical Report CITR-TR-223, Computer Science Department, The University of Auckland, Auckland, New Zealand (2008), www.citr.auckland.ac.nz
12. Li, F., Klette, R.: About the calculation of upper bounds for cluster recovery rates. Technical Report CITR-TR-224, Computer Science Department, The University of Auckland, Auckland, New Zealand (2008), www.citr.auckland.ac.nz
13. Silverman, B.W.: Density Estimation. Chapman & Hall, London (1986)
14. Wu, K.L., Yang, M.S.: Mean shift-based clustering. Pattern Recognition 40, 3035–3052 (2007)

Part XII

Special Session: Neural Information Processing in Cooperative Multi-robot Systems

Using Spiking Neural Networks for the Generation of Coordinated Action Sequences in Robots

Pilar Caamaño, Jose Antonio Becerra, Francisco Bellas, and Richard J. Duro

Integrated Group for Engineering Research, Universidade da Coruña, Spain
`{pcosbrino, ronin, fran, richard}@udc.es`

Abstract. SNNs have been tested as possible candidates for the implementation of robot controllers, in particular behaviour based controllers, but in most approaches their real power, related to their inherent temporal processing, and, especially, temporal pattern generating capabilities, have been ignored. This paper is concerned with showing how SNNs in their most dynamic form can be easily evolved to provide the adaptable or sensor and context modulated pattern generating capabilities required for the generation of action sequences in robots. In fact, the objective is to have a structure that can provide a sequence of actions or a periodic pattern that extends in time from a very time limited sensorial cue.

1 Introduction

Behaviour based robotics has relied heavily on the capabilities of neural networks to model highly non linear relationships between sensed environments and contexts and the resulting actions the robots had to perform. Thus, most of the work carried out up to date has been focused on ways to acquire these models through ANNs and, much in line with the history of the field, they have been based mostly on the first two generations of NNs: a first generation based on McCulloch–Pitts threshold neurons, and a second generation of neurons where continuous activation functions are used.

There is currently a lot of activity surrounding the so called third generation of NNs, that is, spiking neural networks (SNNs). The neurons in these networks rely on pulse streams for the representation of spatial-temporal information much in the same way as biological neurons do. Based on the work by Maass [1] SNNs have been proven more powerful than first and second generation artificial neural network formalisms, requiring fewer nodes to solve the same problems. It has also been proven that they can simulate arbitrary feedforward sigmoidal neural nets and, consequently, may produce any continuous function [2][3]. In addition, compared to classical NNs, the spiking character of SNNs make them more robust to noise.

The main initial drawback of these structures is that, as the outputs of these networks are pulses rather than continuous functions, classical gradient descent based learning rules cannot be used directly. Most of the training algorithms developed are aimed at application specific architectures [4][5] and are generally based on Hebbian learning. In this context, evolution becomes an interesting alternative for obtaining SNNs as it provides a path to create a network structure and its parameters to comply with a desired attractor space.

In the field of robotics the use of SNNs is in its infancy and their full potential has certainly not been achieved yet. In fact in most approaches their real power, which is related to their inherent temporal processing and, especially, temporal pattern generating capabilities, have been ignored. Most authors have used them as substitutes for second generation NNs with a higher computational capacity (requiring smaller networks). Some examples in this line are the papers by Nielsen and Lund [6], who have designed SNN based modular blocks for the construction of robot systems that are trained through Hebbian learning, or Wang et al. [7], who hand design a SNN for an obstacle avoidance behaviour and train it using, again, a variation of Hebbian learning. Examples in the realm of evolutionary approaches are the work of Floreano et al. [4][8], who have studied the evolution of spiking neural controllers for autonomous vision-based robots or Hargas et al. [9], who presented an adaptive genetic algorithm to evolve the weights of the SNNs for the controller.

In the animal world several neural circuits in the spinal cord are in charge of generating patterns that control muscles for motion, these circuits are called Central Pattern Generators (CPG) and their function is to generate oscillatory activity and apply it to the different muscles and muscular groups related to the production of a given desired motion, in general repeated motion such as walking or crawling. In the robotics world this type of controllers has generated a great deal of interest recently in order to provide adaptable mechanisms for creating locomotion controllers for biped, quadruped, n-ped or even snake like robots [10]. Many authors have taken inspiration from the neural mechanisms underlying animal behaviour and have proposed different types of Spiking Neural Networks (SNNs) to underpin their models. However, most of these approaches have been based on hand designing the networks and their connections in accordance with the function the designer wanted for them, thus creating specific architectures for biped or quadruped walking.

However, if one looks at SNNs from another perspective, spiking neurons are inherently dynamic systems which through their membrane voltage are endowed with a modifiable internal state providing the network with a continuous internal memory [11] and through their multiple feedback connections they display the capacity of endogenously generating arbitrary temporal sequences. This ability to generate arbitrary temporal patterns is key in artificial systems such as robots that must operate in dynamic environments. In fact, as commented by Berthouze [12], some typical learning mechanisms, such as learning by observation or by imitation, imply that the system must be able to store and reproduce or recall patterns in their original order with the corresponding timing information.

Neural populations generate periodic and aperiodic patterns of activity or attractors. The stochastic firings of individual neurons give rise to cycles of activity at the population level and thus different attractors, with common properties to mathematical chaos, can be associated to different categories of stimuli or different outputs of the system [13]. The formation of such attractor landscapes in biological neural systems result in such a wealth of behaviours or patterns that, if they can be adequately harnessed in artificial SNN based systems, they could give rise to a very interesting approach for the creation of adaptable or input modulated CPG networks, that is, collections of neurons that are able to endogenously produce rhythmic or oscillatory patterned outputs [14] as a response to sensorial cues. In other words, we would produce a temporal pattern memory that would be modulated by input and

context. This is not a new concept as Mensour and Longtin [15] had already pointed in this direction more than a decade ago when studying recurrence and feedback. They concluded that multistability arising from taking delays into account in the dynamics of simple neural feedback circuits could be used to store information and that the storage capacity increases with the ratio of loop delay to response time of the cell, which in the case of SNNs is quite favourable.

In this paper we want to show how SNNs in their most dynamic form can be easily evolved to provide the adaptable or sensor and context modulated CPG capabilities required for the generation of action sequences in robots. In fact, the objective is to have a structure that can provide a sequence of actions or a periodic pattern that extends in time from a very time limited sensorial cue. In the next sections we will introduce the types of SNNs and evolutionary structures used in this approach. Section 3 is devoted to the presentation of two examples where SNNs were evolved for the generation of sensor modulated action sequences or patterns, one dealing with the control of an arm reaching motion in the presence of different obstacles and the other having to do with the motion patterns of a snake robot when travelling over different surfaces.

2 Spiking Neural Network Model and Evolution

Various spiking neural model exists, such as the Spike Response Model (hereafter, SRM) [16], the dynamic firing threshold model or the integrate-and-fire model (also known as IAF model) [17].

In this paper we implemented a simpler version of the SRM presented in [17] referred to as SRM0. The SRM model uses kernel-based methods to approximate the evolution of the neuron membrane equation. This model describes the state of a neuron as a variable called membrane potential V_i . Each incoming spike that reaches the neuron will perturb the value of V_i and, if after the summation of the inputs the value of V_i reaches the value of the threshold θ , then an output spike is generated. After the neuron has fired, the membrane potential returns to a low value, which is described by the refractory period function η . During this refractory period, it is more difficult to generate an output spike. The evolution of the membrane potential of a neuron is given by the equation:

$$V_i(t) = \eta_i(t - t_i) + \sum_{j \in \Gamma_i} w_{ij} \sum_{t_j^{(f)} \in F_j} \varepsilon_0(t - t_j^{(f)}) \quad (1)$$

In the SRM0, the behaviour of a neuron in the refractory period depends only on the last spike. There are two types of refractory periods, the absolutely refractory period and the relatively refractory period. In the first one, the neuron never generates a spike. During the relatively refractory period is difficult but not impossible to generate a spike. The refractory period behaviour is given by the next equation:

$$\begin{aligned} \eta_i(u) &= -\exp(-u^m + n) H'(u) \theta \\ u &\neq -t_i \end{aligned} \quad H'(u) = \begin{cases} -\infty, & \text{if } 0 \leq u < a \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

Where, θ is the neuron threshold, m and n are the parameters that adjust the slope of the function that defines the relatively refractory period and a is the length of the absolutely refractory period.

The second term of the equation (V_i) represents the postsynaptic potential function (PSP). When a pre-synaptic spike reaches a postsynaptic unit, the postsynaptic potential jumps to the maximum value (in this case 1). Afterwards, it decays exponentially to the resting value, with a rate given by the time constant τ_m . The postsynaptic potential is given by the next equation, where t is the current time, $t_i^{(f)}$ is the firing time of the presynaptic neuron j and d is the length of the synapse or connection delay.

$$\mathcal{E}_0(s) = \exp\left(-\frac{s}{\tau_m}\right) H(s) \quad H(s) = \begin{cases} 1, & \text{if } s \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$s = t - t_j^{(f)} - d$$

The Heaviside function H sets to 0 the value of the kernel for any time moment before the arrival of a pre-synaptic spike.

The Spiking Neural Network implemented in this work is a fully connected network with several parameters to adjust in order to obtain the desired behaviour of each example. These parameters, which are different for each neuron / synapses, are the threshold of the neuron (θ), the length of the absolutely refractory period (a), the parameters m and n that control the length of the relatively refractory period, the length of the synapses (d), the time constant (τ_m) and the strength of the synapses (w). We will adjust these parameters through an evolutionary process using an evolutionary algorithm (EA), specifically a Differential Evolution (DE) algorithm.

The DE algorithm was introduced in 1995 by Rainer Storn and Kenneth Price as a very simple population based, stochastic function minimizer [18]. The basic strategy consists in generating a new chromosome vector, named trial vector, by adding the weighted difference of two randomly selected chromosome vectors to a third one, hereafter the target vector. If the resulting chromosome vector yields a better objective function value than a target, the newly generated chromosome, the trial vector, replaces the original one to which it was compared. In order to increase the diversity of the population, it applies a differential evolution crossover operator to the trial vector, which consists in choosing some genes from the target vector and some from the trial vector. The algorithm has two parameters: the CR parameter that controls the execution of the crossover operator and, consequently, the influence of the parent over the offspring, and the F parameter that scales the influence of the set of pairs of solutions selected to calculate the mutation value.

Currently, there exist several variants of the original DE algorithm. In this paper, we will apply the DE/rand/1/bin scheme, where “rand” indicates that the individuals to compute the mutation values are chosen at random, “1” is the number of pairs of solutions chosen and “bin” means that binomial crossover is considered.

3 Two Examples

In what follows, a couple of examples of how SNNs can produce several complex sequences of outputs (corresponding to robot actions) that cycle continuously and

change depending on just one sensorial input. The first example consists moving a robot arm to a given position in order to grasp an object. In the second example a snake robot chooses a motion pattern depending on the ground surface.

3.1 Arm Reaching for an Object

In this first example, a robot arm has to move from a starting relaxed position in order to reach an object, which can be only in three different positions. The example is very simplified for the sake of clarity. This fictitious robot arm has two links and two joints with DC motors that can turn the links. These DC motors, the actuators, can be instructed to turn left, to turn right (steps of 15°) or to block the position. There is one sensor (virtual) that indicates if the object is in front, to the left or to the right.

In order to implement the controller, a SNN for each actuator is used with 2 inputs encoding the position of the object (01=> object on the right, 10 => object on the left, 11 => object in front) and a total of 8 neurons. The outputs are taken from 1 neuron of each SNN that encode the possible values for the 2 DC motors (00 => maintain the previous command, 01 => turn 15° right, 10 => turn 15° left, 11 => fix the joint) The three possible positions for the object and the final position for the robotic arm in each case are shown in Fig. 1.

The results obtained for this experiment show that depending on where the object is located the SNN is able to learn the sequences of motions for the two joints in the arm in a coordinated fashion so that the objects are reached without hitting any other object. It is important to point out that, using the commented actuators' encoding, the motion patterns are very long and present different lengths (8 bits for an object in front and 20 bits for an object to the right or to the left, see Table 1). In spite of this, the SNN is able to learn and to switch between motion patterns with just one sensorial input. The best fitness and the average fitness of one evolution for one actuator are shown in Fig. 2.

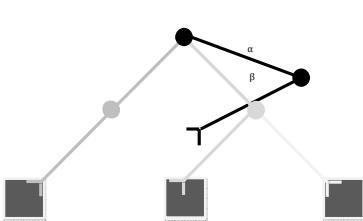


Fig. 1. Initial (black lines) and final (grey lines) positions of the robotic arm depending on object position (squares)

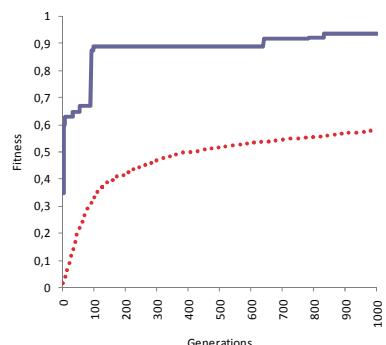


Fig. 2. Average fitness (dotted) and the best individual (solid) fitness of one evolution

3.2 Snake

The second experiment consisted in generating three different motion patterns for a snake robot using a SNN to show that the kind of temporal patterns needed for snake robot motion can be easily obtained using SNNs. The aim is that the SNN outputs the most appropriate motion pattern depending only on the type of surface and, therefore, there is just one input to the system that indicates that.

There are several snake robots used in the bibliography. We have chosen the one described in [19] although any other could be used. In this model, the snake has 7 segments linked by 6 DC motors that control angles between segments. Each segment has two balls in the bottom and, in the middle, a solenoid with a sharp tin pin that can be pressed against the ground and, that way, fix the segment.

For implementing the controller, as indicated before, we have used SNNs to produce three (predefined) different motion patterns, each one adequate for a different ground type. One is presented in [0] and corresponds to a lateral undulation motion. The other two correspond to a slide-pushing motion and the 2D projection of a concertina motion. We can see the motion patterns employed in Fig. 3.

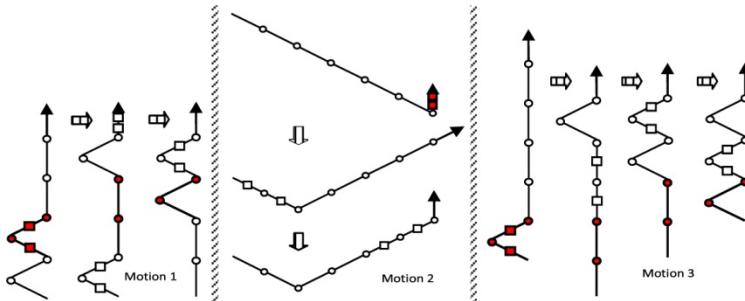


Fig. 3. Motion patterns for the snake robot

Table 1. Actuator values for motion patterns of the robot arm

	Object on the right	Object on the left	Object in front
α	1000110000000000000000	10000000000000001100	10001100
β	01000000000000000000	01000000000000000000	01000000

Table 2. Actuator values for the motion patterns of the snake robot

	Motion 1	Motion 2	Motion 3
t_1 / t_2	010	010	1000
t_3 / t_4	100	000	0101
t_5 / t_6	001	001	0010
t_7 / t_8	010	100	0000
θ_1	000001	010101	00010100
θ_2	111010	011010	10011011
θ_3	000100	010101	01010000
θ_4	101011	010101	01101111
θ_5	010000	010101	01000000
θ_6	011010	110110	01101010

Again, for the sake of easing things for the EA, we have split the controller so there is a different SNN for each actuator (6 neurons each one): 4 for producing the solenoid values (0 => do not fix, 1 => fix to the ground) and 6 for obtaining motor values (00 => -120°, 01 => 0°, 10 => 60°, 11 => 120°). These SNNs have 2 inputs that represent the 3 different possible ground types (01 => ground 1, 10=> ground 2, 11

=> ground 3). One output is taken from each SNN to control either the solenoid or the motor to which the SNN is linked.

Fig. 4 shows that the EA has no problem in finding the solution even if the motion patterns do not have the same temporal length (see in Table 2 that the lengths of the patterns are different for different snake motions). The most important thing here, again, is that the complete motion pattern changes with just one input in a faster and much more elegant way than solutions obtained with second generation ANNs.

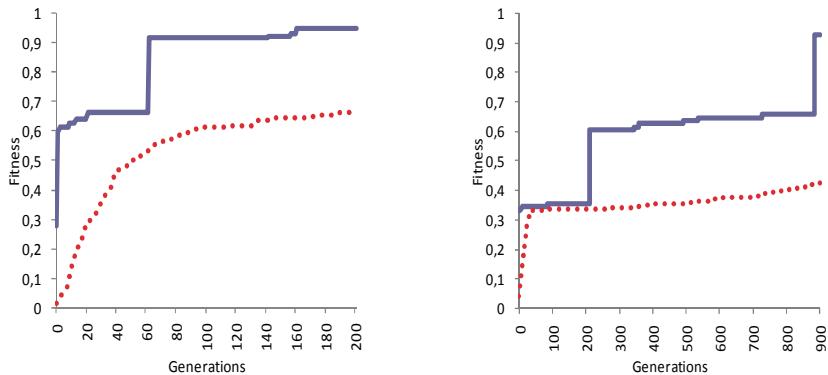


Fig. 4. Average fitness and the best individual fitness of the evolution of one of the solenoid controllers (left) and one of the motor controllers (bottom)

4 Conclusions

SNNs can be easily evolved to provide the adaptable or sensor and context modulated pattern generating capabilities required for the generation of action sequences in robots. In fact, we have shown that SNNs can provide a sequence of actions or a periodic pattern that extends in time from a very time limited sensorial cue. In the examples presented here, using completely connected SNNs a time limited input value could change the patterns generated at the output neurons in a fast and efficient way thus leading to a controlled pattern generator that using an input cue would switch between different complex patterns in one or more outputs which, applied to real hardware (snake or arm) would lead to different coordinated motion sequences.

Acknowledgements. This work was supported by the MEC of Spain through project CIT-370300-2007-18, DPI2006-15346-C03-01 and DEP2006-56158-C03-02.

References

1. Maass, W.: On the computational complexity of networks of spiking neurons. In: Advances in Neural Information Processing Systems, vol. 7, pp. 183–190 (1995)
2. Gerstner, W.: Time structure of the activity in neural network models. Phys. Rev. E 51, 738–758 (1995)

3. Maass, W.: Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons. In: *Advances in Neural Information Processing Systems*, vol. 9, The MIT Press, Cambridge (1997)
4. Floreano, D., Mattiussi, C.: Evolution of spiking neural controllers for autonomous vision-based robots. In: *Proceedings of the International Symposium on Evolutionary Robotics*, pp. 38–61 (2001)
5. Burgsteiner, H.: Training networks of biological realistic spiking neurons for real-time robot control. In: *Proceedings of the 9th International Conference on Engineering Applications of Neural Networks*, Lile, France (2005)
6. Nielsen, J., Lund, H.H.: Spiking neural building block robot with Hebbian learning. In: *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 2, pp. 1363–1369 (2003)
7. Wang, X., Hou, Z.-G., Zou, A., Tan, M., Cheng, L.: A behavior controller based on spiking neural networks for mobile robots. *Neurocomputing* 71, 655–666 (2008)
8. Floreano, D., Zufferey, J.C., Nicoud, J.D.: From wheels to wings with evolutionary spiking neurons. *Artif. Life* 11, 121–138 (2005)
9. Hagras, H., Pounds-Cornish, A., Colley, M.: Evolving spiking neural network controllers for autonomous robots. In: *The Proceeding of IEEE Int. Conf. on Robotics & Automation 2004*, New Orleans, LA, USA, pp. 4620–4626 (2004)
10. Russell, A., Orchard, G., Etienne-Cummings, R.: Configuring of Spiking Central Pattern Generator Networks for Bipedal Walking Using Genetic Algorithms. In: *IEEE International Symposium on Circuits and Systems, 2007, ISCAS 2007*, pp. 1525–1528 (2007)
11. Koopman, A., Van Leeuwen, M., Vreeken, J.: Dynamic neural networks, comparing spiking circuits and LSTM. Tech. Report UU-CS-2003-007, Institute for Information and Computing Sciences, Utrecht University (2003)
12. Berthouze, L.: An episodic memory model using spiking neurons. *IEEE Int. Conf. on Sytems, Man, and Cybernetics* 1, 86–91 (2000)
13. Harter, D.: Evolving neurodynamic controllers for autonomous robots. In: *Proc. 2005 Int. Joint Conference on Neural Networks (IJCNN 2005)*, pp. 137–142 (2005)
14. Stein, P.S.G., Grillner, S., Selverston, A.I., Stuart, D.G.: *Neurons Networks and Behaviour*. MIT Press, Cambridge (1997)
15. Mensour, B., Longtin, A.: Encoding information in neural feedback circuits. In: *IEEE 17th Annual Conference on Engineering in Medicine and Biology Society*, vol. 2, pp. 1463–1464 (1995)
16. Gerstner, W., van Hemmen, J.L.: Associative memory in a network of “spiking” neurons. *Network* 3, 139–164 (1992)
17. Maass, W., Bishop, C.M.: *Pulsed Neural Networks*. MIT Press, Cambridge (1999)
18. Storn, R., Price, K.: Differential Evolution - a Simple and Efficient Adaptive Scheme for Optimization over Continuous Spaces TR-95-012, ICSI (1995)
19. Shan, Y., Koren, Y.: Design and Motion Planning of a Mechanical Snake. *IEEE Trans. Systems, Man and Cybernetics* 23(4), 1091–1100 (1993)

Neuro-Evolutive System for Ego-Motion Estimation with a 3D Camera

Ivan Villaverde*, Zelmar Echegoyen, and Manuel Graña**

Computational Intelligence Group

University of the Basque Country

<http://www.ehu.es/ccwintco>

Abstract. A neuro-evolutive system for mobile robot ego-motion estimation using time-of-flight (TOF) 3D camera readings is presented in this paper. It is composed of two modules. First, a Neural Gas adaptative algorithm is used to obtain a set of codevectors quantizing the preprocessed 3D measurements provided by the camera. Second, codevector sets from consecutive frames are matched using an evolutive algorithm in order to estimate the motion of the robot between those two positions.

1 Introduction

Recently, a wide new research area in mobile robotics has been opened by the commercialization of a new generation of time-of-flight (TOF) 3D cameras. Those affordable, small and lightweight cameras have made possible that such devices could be used not only in high budget projects or in static or off-line processing systems, but also as part of mobile robotic platforms as on board sensors. Previous 3D cameras were very expensive, bulky, require high power consumption and were very slow for mobile robotics applications.

Working on the broad area of multi-robot systems, we are focusing our efforts on the use of TOF 3D cameras to perform Simultaneous Localization and Mapping (SLAM) [1,2]. As a previous step toward this objective, we are currently working on ego-motion estimation from the 3D camera readings. In this paper, we introduce some modifications to the system described in [3] improving the system's ego-motion estimation.

The neuro-evolutive system is composed of two main modules plus a preprocessing step. The first module computes the approximation of the preprocessed camera 3D data. This approximation is a vector quantization of the 3D data given by a set of 3D codevectors. In this paper we use the Neural Gas (NG) [8] algorithm instead of the Self Organizing Map (SOM) used in [3]. The NG obtains better quantizations of the 3D data. The second module is an Evolutive Algorithm [4] which performs the task of estimating the motion parameters by searching on the space of linear transformations restricted to the translation and

* Ivan Villaverde holds a predoctoral grant from the Basque Country University.

** This work was supported in part by the Spanish Ministerio de Educación y Ciencia under grant DPI2006-15346-C03-03.

rotation, applied on the codevector sets obtained by the NG for successive camera readings. The fitness function is the matching error between the transformed last set of codevectors and the codevector set corresponding to the next camera readings. Matching of codevector sets is based in the euclidean distance. It is performed searching for the closest codevector and is a potentially many-to-one correspondence.

In section 2 we will explain briefly the sensor characteristics in order to give a proper context to the data and understand the need of its preprocessing. In section 3 we present the neuro-evolutive system that performs the ego-motion parameter estimation. In section 4 we present some experimental results on a Pioneer robot incorporating the camera. Finally, section 5 provides some conclusions and further research efforts.

2 Sensor Data and Preprocessing

In our work, we use a Swiss Ranger SR-3000 time-of-flight range camera [5]. Their full specs and capabilities are not within the scope of this paper, but some of its characteristics determine how its data must be processed. The SR-3000 is based on phase-measuring Time of Flight principle [6]. The SR-3000 camera uses an array of near-infrared leds in order to illuminate the scene. Knowing the amplitude of the wavelength of the light emitted, it can determine the distance to objects measuring the phase delay of the reflected light, within a range of non-ambiguity determined by the wavelength. Figure 1 shows the conventional visible wavelength image obtained from a specific robot position and orientation. This position will be used as reference for following illustrations. Figure 2 shows the raw data provided by the 3D camera, which consists of two matrices storing at each pixel the measured distance and the intensity of the reflected infrared light. Since each pixel corresponds to fixed azimuth and zenith angles in spherical coordinates, we can transform the distance image to more appropriate Cartesian coordinates (shown in figure 3) and obtain a cloud of 3D points, representing the camera measurements.

This 3D point cloud is extremely noisy and it's necessary a strong filtering in order to retain only relevant points. As can be seen in figures 2 and 3, most of the noise comes from measurements beyond the non-ambiguity range: far away measurements (from space that is through the door) appear to be very close to the camera, forming a dense cone near the vertex of the cloud of points. Also, pixels without any object in their line of sight present measurements spread along all distances, caused by specular reflections. As those points are supposed to be far away, the reflection intensity should be very low. Taking this into account, we define a confidence value C for each pixel i , calculated proportionally to its measured distance and intensity:

$$C_i = I_i \times D \quad (1)$$

This confidence value allows filtering noisy points, while keeping relevant distant points with low intensity that could be discarded if filtering is done only by intensity.



Fig. 1. Reference visible wavelength image from the optical camera view

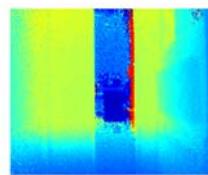
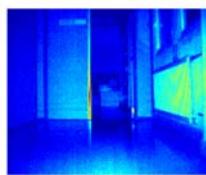


Fig. 2. Intensity (left) and distance (right) images from the 3D camera. Note that the distance measured beyond the door is very low (dark blue) due ambiguity range.

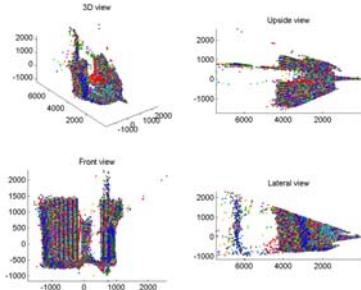


Fig. 3. Plotting of the cloud of points in 3D Cartesian coordinates extracted from the distance image provided by the TOF 3D camera

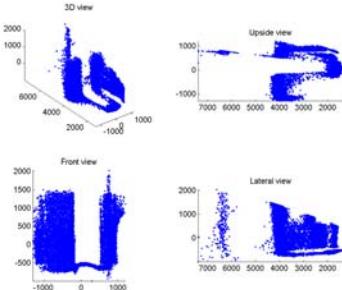


Fig. 4. Plotting of the cloud of points in 3D Cartesian coordinates extracted from the distance image provided by the TOF 3D camera after filtering out ambiguous readings. Relevant surfaces can be appreciated.

3 Neuro-Evolutive System

The filtered cloud of points (figure 4) still presents two big problems, namely, the big size of the data set (usually more than 15.000 points per 3D frame) and the uncertainty in the measurements of the surfaces. In order to cope with both problems, in previous works [3] we trained a Self Organizing Map [7] to fit the cloud of points. The objective of this step was to obtain from the SOM a grid that kept the spatial shape of the points cloud and, hopefully, of the objects in the environment, and at the same time reduces dramatically the size of the data set. We also worked in the assumption that the same point in the SOMs extracted from two consecutive frames will correspond approximately to the same spatial point, which make a lot easier the computation of the distance between grids. However, the assumption that every point in one frame has a corresponding point in the next one will only be true if the movement is smooth

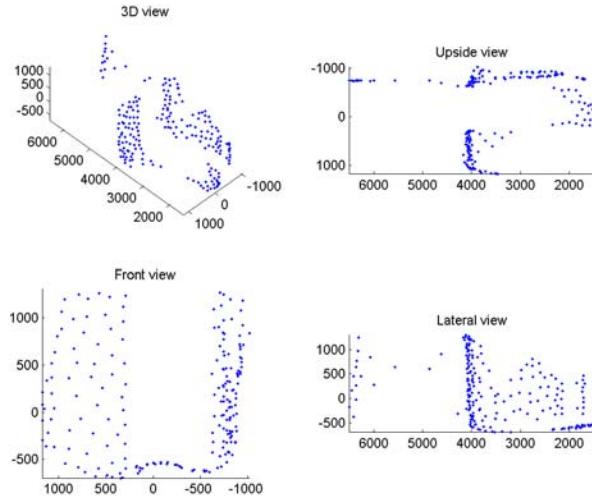


Fig. 5. A codevector set of 200 nodes trained by NG to quantize the reference position filtered point cloud shown in figure 4

enough. In fact, it should be so smooth that it makes it useless in the practice. Since we are acquiring the 3D measurements from a mobile robot, traveling at a reasonable speed, in each frame some of the measurements corresponds to areas of the environment that were out of range in the previous frame, while some of the measurements in the previous one would fall behind in the new frame. That is, the physical surfaces imaged in two consecutive frames may overlap only partially, thus the TOF 3D Camera reading may have optimal partial matches.

This problem is better tackled with the use of Neural Gas [8]. The NG computes a vector quantization of the data as SOM, but with the advantage (in our case) that its points are not constricted to a fixed grid. This reduces considerably the problem of the existence of nodes with no correspondence to the input data, lying in 3D space regions without input camera data shown by the SOM trained grids, due to the topological preservation properties of SOM. In figure 5 can be seen the point cloud generated by fitting a 200 nodes NG to the reference position measures. It can be appreciated that physical surfaces can be easily identified.

So, the input data for the ego-motion parameter estimation consists of a sequence of codevector sets S corresponding to TOF 3D camera frames, computed by the NG, that approximate the shape of the environment in front of the robot at each time instant corresponding to a 3D camera frame. The robot is described by its position at each time instant t given by $P_t = (x_t, y_t, \theta_t)$, and the codevector set S_t fitted over the observed data. At the next time instant $t+1$ we obtain S_{t+1} from the camera. Our objective is to estimate the position P_{t+1} from the knowledge of the codevector set S_{t+1} and the previous estimation of the position at time t . We assume that, from two consecutive positions, the view of the

environment is approximately the same, but from a slightly different point of view (i.e., the robot is viewing the same things, but from other position). Since most of the objects in S_t are expected to be present also in S_{t+1} , the way to calculate this new position is to calculate the transformation T that S_t requires to match S_{t+1} . So, our objective will be to search for the parameters of T which minimize the matching distance between S_{t+1} and the transformed codevector set $\hat{S}_{t+1} = T \times S_t$.

In our previous work, we applied a very basic Evolution Strategy (ES) to search for the new position of the robot. Since new generations were generated by merely adding Gaussian perturbations to best fitted individuals, convergence was very slow. Since quick response time is crucial in mobile robotics, we had to establish very light stopping conditions which made the ES very prone to fall in suboptimal solutions. For the experiments presented in this paper, we have developed a more complete genetic algorithm, using heredity and crossover to keep the traits of the best fitted in the new generations.

Since we are looking for the transformation matrix T , the traits of the ES individuals will encode the parameters of T . Although the data consists in 3D point clouds, the robot is moving only along the plane of the floor, so we only need the parameters necessary for the transformation within that plane. So, each individual would be the hypothesis $h_i = (x_i, y_i, \theta_i)$, where x , y and θ are the parameters of the transformation matrix T , and also correspond to the relative position between P_t and P_{t+1} .

$$T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & x_i \\ \sin(\theta_i) & \cos(\theta_i) & y_i \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

For each hypothesis h_i encoded by an ES individual we have a prediction

$$(\hat{S}_{t+1})_i = T_i \times S_t \quad (3)$$

which is used to calculate the fitness function as a matching distance between codevector sets, computed as the sum of the euclidean distances from the points of $(\hat{S}_{t+1})_i$ to its closest point in S_{t+1} .

Initial population is built from one initial hypothesis assumed to be the origin $h_0 = (0, 0, 0)$ (i.e., no transformation: the robot has not moved from previous position). Each generation a new population is built from the one third of the previous population with the best fitness function values. Those best fitted pass directly to the next population, and the rest of the individuals are generated by crossing them. Each crossing is made selecting randomly two individuals (h_1 , h_2) from the best fitted and two descendants (d_1 , d_2) are generated following the crossing pattern:

$$\begin{aligned} d_1 &= [x_1, y_2, \theta_1] \\ d_2 &= [x_2, y_1, \theta_2] \end{aligned} \quad (4)$$

Each trait of the descendants is then mutated, with a 50% probability, by adding gaussian perturbations from a normal distribution with mean zero and standard deviation with default value of 10.

With the obtained transformations, we can estimate the ego-motion of the robot. Starting from initial position x_0 , and given a calculated transformations sequence $T = T_1, \dots, T_t$, robot's position at time step t can be calculated applying consecutively the transformations to the starting position:

$$x_t = T_t \times \dots \times T_1 \times x_0 \quad (5)$$

4 Experimental Settings and Results

To test the approach, we have collected data from a series of pre-recorded walks across the corridors of our building in which data (3D readings, optical camera images and odometry) was recorded in approximately 20 cm. steps. Several NG configurations were tested, mostly testing the balance between efficiency (number of nodes) and the ability to accurately fit the data, highlighting the physical surfaces. Results presented in this section are based on an NG with 100 nodes.

The ES populations are composed of 20 individuals and are evolved as described in section 3. The matching distance used as fitness function is calculated also as described in the same section. Convergence condition is reached when the distance between the one third best fitted individuals is below a minimum euclidean distance (as they represent a point in the plane) and have the same orientation. Waiting for full convergence do not provide much better results, so, since fast operation is a critical issue in mobile robotics, this threshold is



Fig. 6. Path estimation following a turning trajectory and its correspondent camera images

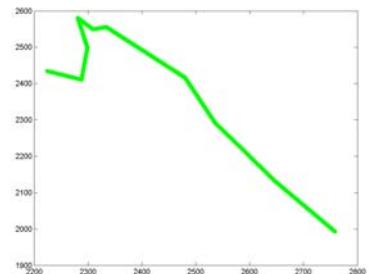


Fig. 7. Erroneous path estimation of a turning trajectory and its correspondent camera images

typically set to a distance around 2. The best fitted individual is selected as result.

Experimental results have shown that our ES provides a good surface matching, given that is some good feature to match inside the field of view of the camera. In figure 6 we can see how a turning trajectory is correctly estimated by the algorithm.

As other purely sensor based localization/ego-motion/servoing methods, the capacity to make correct estimations is highly dependent on the presence of good features in the measurements. That means that if there isn't "something" (typically high point density areas which usually represent objects in the environment of the robots) to match within the codevector sets, the estimation is going to be wrong with a high probability. This problem is similar to occlusion problems in vision based systems, where occlusion of the visual features used as landmarks leads to error in localization. In figure 7 can be seen an example where in absence of vertical surfaces, which provide the best matching features, the path followed is wrongly estimated. Solution for this problem should come, as in other methods, by the embedding of this technique within predictive corrective techniques, like Kalman filters.

5 Conclusions and Further Work

In this paper, we have presented our advances toward a new mobile robot ego motion estimation neuro-evolutionary algorithm, based on information obtained from a 3D camera and a hybrid intelligent system composed of a NG and an ES. The 3D camera provides information about the distances to the objects in the robot environment. This information is translated to a cloud of points, which is filtered and fitted with a NG, in order to obtain a codevector set with the shape of the objects in the near environment of the robot. Those codevector sets are used to estimate the next position, evolving a population of hypothesised positions using a matching distance as fitness function. Some experimental results have also been presented and its problems discussed.

We are currently working on identifying the advantages and flaws of the method, in order to, as our most immediate future work plan, integrate this technique into a Kalman or particle filter SLAM architecture. There are also space for improvements in several parts of the algorithm, as different surface matching techniques can be developed and tested and fine tuning of the parameters of the ES (like population size, stopping condition, adaptive mutation, etc.) can be performed. Another option is the substitution of the genetic algorithm by a well-known point cloud registration algorithm, like ICP or LS3D, and compare its results and performance. Finally, we plan to fuse the information from the 3D camera with the optical camera and other sensors to enhance the robustness of SLAM systems on the Pioneer robot.

References

1. Dissanayake, G.: A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation* 17(3), 229–241 (2001)
2. Thrun, S.: Robotic Mapping: A Survey. In: Lakemeyer, G., Nebel, B. (eds.) *Exploring Artificial Intelligence in the New Millennium* (2002)
3. Villaverde, I., Graña, M.: A Hybrid Intelligent System for Robot Ego-Motion Estimation with a 3D Camera. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) *HAIS 2008. LNCS*, vol. 5271, Springer, Heidelberg (2008)
4. Randy, L.H., Sue, E.H.: *Practical Genetic algorithms*, 2nd edn. Wiley Interscience, Hoboken (2004)
5. Oggier, T., Lehmann, M., Kaufmann, R., Schweizer, M., Richter, M., Metzler, P., Lang, G., Lustenberger, F., Blanc, N.: An all-solid-state optical range camera for 3D-real-time imaging with sub-centimeter depth-resolution (SwissRanger). In: Proc. SPIE, vol. 5249, pp. 634–545 (2003)
6. Lange, R., Seitz, P.: Solid-State Time-of-Flight Range Camera. *IEEE J. Quantum Electronics* 37(3), 390–397 (2001)
7. Kohonen, T.: *Self-Organizing Maps*, 3rd edn. Springer, Heidelberg (2000)
8. Martinetz, T.M., Schulten, K.J.: A neural-gas network learns topologies. In: Kohonen, T., Mäkisara, K., Simula, O., Kangas, J. (eds.) *Proc. International Conference on Artificial Neural Networks*, pp. 397–402. North-Holland, Amsterdam (1991)

Neuro Granular Networks with Self-learning Stochastic Connections: Fusion of Neuro Granular Networks and Learning Automata Theory^{*}

Darío Maravall and Javier de Lope

Perception for Computers and Robots

Universidad Politécnica de Madrid

dmaravall@fi.upm.es, javier.delope@upm.es

Abstract. In this paper the fusion of artificial neural networks, granular computing and learning automata theory is proposed and we present as a final result ANLAGIS, an adaptive neuron-like network based on learning automata and granular inference systems. ANLAGIS can be applied to both pattern recognition and learning control problems. Another interesting contribution of this paper is the distinction between pre-synaptic and post-synaptic learning in artificial neural networks. To illustrate the capabilities of ANLAGIS some experiments with multi-robot systems are also presented.

1 Knowledge-Based Production Rules with Self-learning and Adaptive Capabilities

Knowledge-based production rules of the type *If {condition} Then {Action}* are very powerful and widely used computational instruments to encode the knowledge needed for the implementation of intelligent systems. Not surprisingly they are the basic building blocks of expert systems, probably the most significant and successful contribution of the early “symbolic Artificial Intelligence” [1]. However, production rules-based expert systems have been criticized of being brittle [2] in the sense that they are unable to generalize from narrow, specialized domains, where they usually perform quite satisfactorily. Therefore, big efforts have been devoted to endow production rules with self-learning and adaptive capabilities [3,4].

In this paper we propose an artificial Neuro Granular Network with self-learning and adaptive stochastic connections which is based on the fusion of knowledge-based production rules, implemented by a Neuro Granular Network, with stochastic connections that is able to learn and adapt to novel domains and environments. These learning and adaptation capabilities are based on Learning Automata Theory [5].

* This work has been partially funded by the Spanish Ministry of Science and Technology, project: DPI2006-15346-C03-02.

2 Adaptive Neural Network Based on Learning Automata and Granular Inference Systems: ANLAGIS

Summarizing, our proposed Adaptive Neuron-like network of Granular and Learning Automata System (ANLAGIS) is composed of two principal parts or subnets: (1) the granulation part or classification subnet which plays the role of partitioning the state space into undistinguishable atomic information granules or states or, as we propose, to call them *ballungen*¹; (2) the second part of ANLAGIS is the action-selection subnet which is in charge of choosing for each particular information granule or state or ballungen, the appropriate available action. This part of ANLAGIS is based on a learning automaton  that chooses each possible action according to a probability distribution that varies as a function of the responses (either punishment or reward) of the environment. This learning automaton governs the self-learning and self-adaptive stochastic connections of ANLAGIS. In Fig.  it is shown the block diagram of ANLAGIS.

As commented a few lines above and in the caption of Fig. , ANLAGIS consists of two main parts: the first one is in charge of partitioning the state space in the appropriate information granules or basic states or clusters or *ballungen* and the second part of ANLAGIS (the right side of Fig. ) is devoted to choose the most appropriate action for each state or atomic granule: this choice is based on a learning automaton that updates its probability distribution as a function of the responses given by the environment (either a reward or a punishment). Thanks to the existence of a learning automaton for each state or information granule ANLAGIS is able to learn in real-time with unsupervised guidance the proper actions.

3 The Granulation Process of ANLAGIS

We briefly describe the above mentioned granulation process as formed by the following steps.

Step 1. Selection of the most representative or principal state variables of the system x_1, x_2, \dots, x_n , which are the n principal state variables that optimize the representation of all the undistinguishable states or *ballungen* or clusters.

¹ In tribute to the Austrian philosopher and outstanding member of the Vienna Circle Otto Neurath who in his master piece philosophical paper *Protokol Sätze* (Protocol Sentences)  introduced the term *ballungen* (balloons, clots, conglomerates) to designate *verbal clusters*, a concept precursor to the idea of Wittgenstein's private language and also close to Michael Polanyi's tacit knowledge. Our concept of granulation process in ANLAGIS also attempts to formalize internal numerical sensory information clusters and we think that Neurath's *ballungen* is one of the first attempts in this direction. The final step of anchoring or grounding these internal clusters into symbols in the case of a pattern recognition problem or into actions in the case of a learning control problem we propose to be implemented by means of a probability distribution giving an additional degree of freedom to the self learning capability of ANLAGIS.

Step 2. Granulation or partition of all the state variables. In this paper we propose to apply a conventional Boxes-like [7] based granulation of the state variables. Roughly speaking, the interval granulation is a hard open problem in every particular application; here we propose to apply evolutionary methods so that the granulation subnet of ANLAGIS requires an evolutionary training process.

4 Self-learning and Adaptive Behavior of ANLAGIS: Reinforcement Learning Based on Learning Automata Theory

Once the principal state variables have been selected and partitioned in the corresponding intervals then the atomic information granules or states or clusters or *ballungen* emerge as a consequence of the interaction with the environment, so that the next step is to map the *ballungen* space $\mathcal{X} = X_1, X_2, \dots$ to the action space $\mathcal{A} = a_1, a_2, \dots, a_r$. This mapping process is known in the reinforcement learning parlance as the policy mapping or planning problem. In control theory parlance is known as the controller design problem.

One of our main contributions in this paper is to propose a probabilistic policy mapping based on learning automata theory so that for each individual state the available actions are selected by means of a probabilistic distribution which is

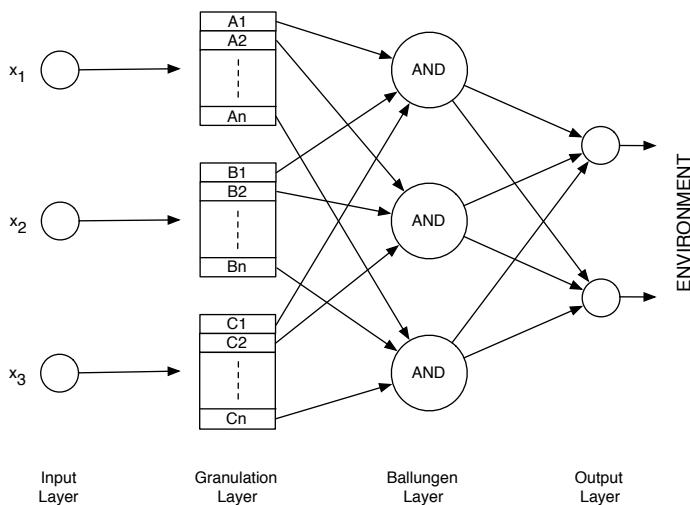


Fig. 1. Block diagram of ANLAGIS. The left-side of the whole network consists of the granulation process of the state variables x_i , so that after the previous interval granulation of each state variable there appears the corresponding AND-neuron that computes the logical conjunction of several interval granulations. Each AND-neuron has $1, 2, \dots, r$ possible outputs (which are chosen according to a probability distribution that changes in function of the environment responses: see the text for details).

updated depending on the environment responses (reward or punishment). In other words, in this paper we propose to apply a learning automaton for each individual *ballungen*. As the *ballungen* in ANLAGIS are implemented by means of an artificial neural network (i.e. the left side of Fig. 1 or the granulation process described above) and more specifically as the *ballungen* are detected by the AND-neurons in Fig. 1 which, at the same time, are connected to every existing output neurons through probabilistic connections (links) then we propose a novel approach to artificial neurons as explained in the sequel.

5 A Novel Approach to Artificial Neural Networks: Pre-synaptic Learning *vs.* Post-synaptic Learning

The conventional “input-oriented” artificial neuron model with pre-synaptic learning is depicted in Fig. 2 below. The activation of neuron’s output y can be expressed as:

$$y_i = f(\tau, \sigma w_i, x_i) \quad (1)$$

where τ is the threshold of neuron i and w_i are its pre-synaptic weights; similarly f is the activation function usually implemented as the sigmoid function. As is well known in this conventional neuron model learning is based on the variation of the pre-synaptic weights as a function of the training error.

Now we propose to consider another perspective and to focus on the neurons output that in principle can be connected to several target neurons as depicted in Fig. 3.

In the “output-oriented” artificial neuron model the central idea is based on the connections of a generic neuron through its axonal output to what can be called target neurons. In this paper we propose to define a probability p_i for each possible connection so that we can use learning automata theory to introduce a learning capability for the output links of a generic neuron. In this novel proposal each neuron has an associated learning automaton in which the action probability vector $P = \{p_i\}$ is defined over the output connections of the corresponding neuron. For this reason we propose the term post-synaptic learning in which, as depicted in Fig. 3, the pre-synaptic weight of target output neuron, w_{i_1} , is also the post-synaptic probability of neuron N_i as regards to its probability connection to output target neuron. In other words, this pre- and

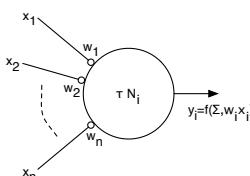


Fig. 2. Input-oriented artificial neuron model

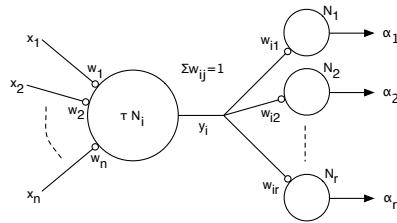


Fig. 3. The “output-oriented” artificial neural network in which a generic neuron n_i is connected to r different target neurons: $N_{i1}, N_{i2}, \dots, N_{ir}$. In this case learning takes place in the probability of connecting the neuron N_i to the different target neurons (see text for details).

post-synaptic weights $w_{ij}(j = 1, \dots, r)$ are the probabilities of the associated learning automaton so that they hold:

$$0 \leq w_{ij} \leq 1 \quad ; \quad j = 1, \dots, r \quad ; \quad \sum_{j=1}^r w_{ij} = 1 \quad (2)$$

Obviously the post-synaptic learning is based on the reinforcement received from the environment as it happens in learning automata theory.

6 Multi-robot Testing System Environment

We have defined a multi-robot system to illustrate the capabilities of the proposed system in which we have run several experiments. As the experiments as the environment are inspired by soccer, football, rugby, basketball or any game played between two teams. The multi-robot system is composed by four robots and they set up two teams, being each team formed by two robots. A possible configuration of the teams is shown in Fig. 4.

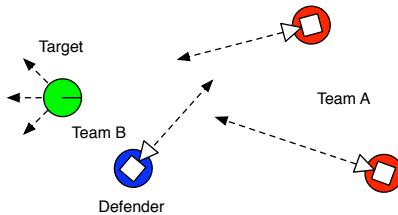


Fig. 4. Scheme of the experimental scenario defined for testing. It is composed by robots teams with cooperative and competitive goals.

The A-team’s goal is to touch one of the B-team robots which we named as *target*. Each A-team robot has its own controller based on ANLAGIS but the goal is the same for each robot. That is, each robot has its own set of rules which

must be learned. They do not share any kind of memory. The coordination, if exists, is established as an emergent behavior.

The B-team's goal is to avoid the contact of the target robot with one of the opposite team robots. In this team, each robot has a different role: the target robot can be immobile or perform random or reactive movements based on its range sensors. The goal of the second robot which we call the *defender*, is to avoid the contact between the opposite team robots and the target robot.

If the B-team defender touches one of the A-team attackers, the turn is finished and the point, the *reward*² is for the B-team. If one of the A-team robots reaches the target, the point or reward is for the A-team. Otherwise, if there is not contact between robots, the turn is finished after a determined time and is considered as a draw.

The A-team robots and the defender are equipped with an onboard camera. Each robot can detect and recognize the color of the other robots and identify them as the target, a friend or an enemy. As we have previously commented, the target robot only is equipped with a ring of sonar sensors. All the robots have collision detectors.

7 Designing the Control Systems

We have stated that the first step for applying ANLAGIS is to define the *bal-lungen* or clusters. Previously the state variable must be identified.

The robots have a similar sensorial system composed by a camera and a bumper. The task to be accomplished by both teams can be reduced to reach a position defined by a visual landmarks, trying to avoid the contact with the other elements that also are present in the image. The landmark and the other elements are determined by colors. Thus, several well-known algorithms can be applied in order to localize the target and elements into the digital image and to determine theirs relative positions in the world.

We can reformulate the task in such a way that the controller must keep the visual target into a specific image window while it also tries to elude the contact with other visual elements in that window.

Therefore, the state variables are defined by the image coordinates of the other robots. The granulation of the state variables is made by using the horizontal coordinate. When the goal is to reach the target, the value must be near to the center of the image, assuming that the camera axis is parallel to the robot movement direction. The window width is a parameter to be determined and also the number of intervals in the window. We have considered three intervals, so that the target can be on the left, in front or on the right of the target. We have considered several window widths. When the goal is to reach the target the use of narrow windows is preferred. When the goal is to avoid the contact, it is better to use greater values.

² Depending on the learning strategy used, the reward in a Reinforcement Learning sense is also determined during the turn or episode and not only when it is finished.

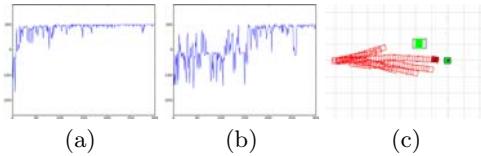


Fig. 5. (a) Incremental algorithm reward, (b) Q-learning reward and (c) robot runs

We have considered three actions for the defender and attackers robots: turn left, turn right or move ahead. The speeds have been determined by hand and we use a greater values for the attackers than for the defender in order to give them some chance to win the game. The optimum values for all these parameters, the robots speeds, the windows widths and number of intervals, can be easily obtained by using evolutionary strategies.

We have designed a reinforcement learning algorithm based on [9] for determining what actions must be performed in every situation, i.e. if the robot must turn or not in order to reach the target depending on the presence of the other robots. We have also used a conventional Q-learning algorithm but we found that the incremental learning was much faster with our approach. Fig. 5a and Fig. 5b show a typical reward learning curves followed by the incremental and Q-learning algorithms, respectively, in the simple case that a robot must reach a visual landmark. After several initial runs, the robot learns to maintain a orientation that guides it to the target. Fig. 5c shows these initial runs. Also it can be observed as the target robot is maintained practically in the center of the image which is represented by an small window over the robot.

8 Experimental Results

Fig. 6 shows several cases that can be commented. In Fig. 6a it is shown the camera images. It can be observed that each robot keeps centered its target. Fig. 6b illustrates a situation in which the B-team was the winner: the defender collides with the robots and it closes the path to the target robot. This is a very usual case. Fig. 6c shows a very interesting situation on which the A team employs a tactic in which one of the attacker robots takes the attention of the defender, allowing the second attacker to reach the target easily. Fig. 6d represents another situation in which both attackers get stuck.

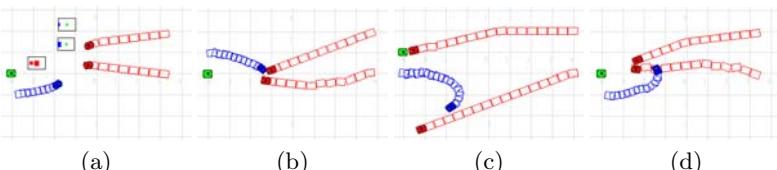


Fig. 6. Some experiment results (see text for details)

9 Conclusions and Further Work

A novel approach for solving complex pattern recognition and learning control problems based on the fusion of artificial neural networks, granular computing and learning automata theory is proposed. Also we discuss the distinction between pre-synaptic and post-synaptic learning in artificial neural networks. This distinction adds probabilistic terms to the neuron output connections which allow a certain degree of nondeterministic results to the output. This fact favors the general system performance.

To illustrate the capabilities of ANLAGIS some experiments with multi-robot systems are also presented. The multi-robot system for testing is inspired on a game between two teams composed by several players. This allows verify the system behaviour in a multi-agent system with competitive and cooperative objectives.

The results presented in this paper are made in a simulated environment. We are currently working in the creation of an environemnt with real robotic and sensing systems.

References

1. Kasabov, N.K.: Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering, ch. 2. MIT Press, Cambridge (1996)
2. Holland, J.H.: Escaping brittleness: The possibilities of general purpose learning algorithms applied tn paralle rule-based systems. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) Machine Learning II, pp. 593–623. Morgan Kaufmann, San Francisco (1996)
3. Butz, M.V.: Learning Classifier Systems. In: Proc. GECCO 2008, pp. 2367–2388. ACM Press, New York (2008)
4. Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.): IWLCS 2003. LNCS, vol. 2661. Springer, Heidelberg (2003)
5. Thathachar, M., Sastry, P.: Varieties of Learning automata: An Overview. IEEE Trans. on Systems, Man and Cybernetics, Part B 32(6), 711–722 (2002)
6. Neurath, O.: Protocol sentences, Logical Positivism (The Library of Philosophical Movements), pp. 199–208. A.J. Ayer Free Press (1959)
7. Michie, D., Chambers, R.A.: Boxes: An experiment in adaptive control. In: Dale, E., Michie, D. (eds.) Machine Intelligence, vol. 2, pp. 137–152. Oliver & Boyd (1968)
8. Narendra, K., Thathachar, M.: Learning Automata - A Survey. IEEE Transactions on Systems, Man, and Cybernetics 4(4), 323–334 (1974)
9. Maravall, D.: New reinforcement algorithm in learning automata. IEEE Electronic Letters 16(16), 623–624 (1980)

An Incremental Learning Algorithm for Optimizing High-Dimensional ANN-Based Classification Systems

Abraham Prieto, Francisco Bellas, Richard J. Duro, and Fernando Lopez-Peña

Grupo Integrado de Ingeniería, Universidade da Coruña, Spain

abprieto@udc.es, fran@udc.es, richard@udc.es, flop@udc.es

Abstract. This paper presents an incremental learning algorithm primarily intended for adjusting and optimizing high dimensional backpropagation ANN-based supervised classification systems. The algorithm avoids the highly time-consuming pre-processing stage used to reduce dimensionality through the deletion or averaging of redundant information and the establishment of an appropriate processing window. The proposed algorithm acts during the training process of the ANN by automatically obtaining the optimal window size and transformation parameter values needed for a given set of classification requirements. During this process, it changes the network topology on line by adaptively appending input units and their corresponding connections to the existing network. The proposed process allows the ANN to learn incrementally by adapting to the new topology without forgetting what had been learnt earlier. This process could also be used as an incremental learning system in, for instance, robotic systems; when new instances are fed, as it would not need to perform a whole training stage. Instead, the knowledge encoded in the new instances could be learnt through the proposed adjustment of network topology.

1 Introduction

Artificial Neural Networks (ANNs) are widely used for supervised classification or categorization of high dimensional datasets [1]. There are some well known problems arising when analyzing high dimensional data which are related to the nature of the data itself, independently of the methods used for classification; this is mainly due to “the curse of dimensionality”, that basically states that, in any function approximation and optimization, the number of evaluations of the function needed to obtain a given level of approximation grows exponentially with the number of dimensions [2]. Nevertheless, the characteristics of high dimensional feature spaces point out that multivariate data in general are located on a submanifold of much lower dimensionality [3], consequently, any high dimensional dataset could be projected into a lower dimensional space without significant loss of information [4]. Dimensionality reduction methods have been extensively applied in unsupervised applications [5] but not quite as much in supervised ones [6], partially due to the presence of class labels which notably complicate the reduction process. Dimensionality reduction is widely applied to autonomous robots in applications such as perception [7] or map building [8] among others.

In this paper, we propose an incremental learning algorithm that automatically determines the dimensional reduction parameters and the processing window size required by a backpropagation ANN-based supervised classification system. The adjustment of the system parameters by this incremental learning algorithm takes place during the training of the ANN. The algorithm not only acts on the signal under analysis but also on the network topology by increasing its input dimension. This increase corresponds to a progressive refinement of the input data introduced by re-encoding the dataset of multidimensional vectors in any selected data window into a new incremental representation. As the backpropagation learning network is not incremental in nature, in general it should be completely retrained after suffering this sort of modifications. However, the proposed incremental learning algorithm allows learning the new instances while preserving old memory. The proposed algorithm is tested here on high dimensional image processing applications, mainly hyperspectral imaging classification, due to the availability of real data of very high dimensionality in this realm, but it could be used in other applications as well.

2 The ANN-Based Classification System

The system core consists of three main interacting parts that can be seen in the block diagram of Fig. 1. The left block represents the training dataset. The second and third blocks make up *the first processing stage* where the transformation of the portion of the raw data under consideration into the feature space description takes place through the determination of the window size and the dimensional transformation parameters. The fourth block corresponds to the ANN-based classifier and constitutes the *second stage* of the system, where the parts of the dataset are classified into the categories provided by the expert's labeling. Finally, an adviser (*third stage*) using the variation trends of the error throughout the training process adjusts the parameters of the first stage so as to better adapt them to the production of the best possible classification.

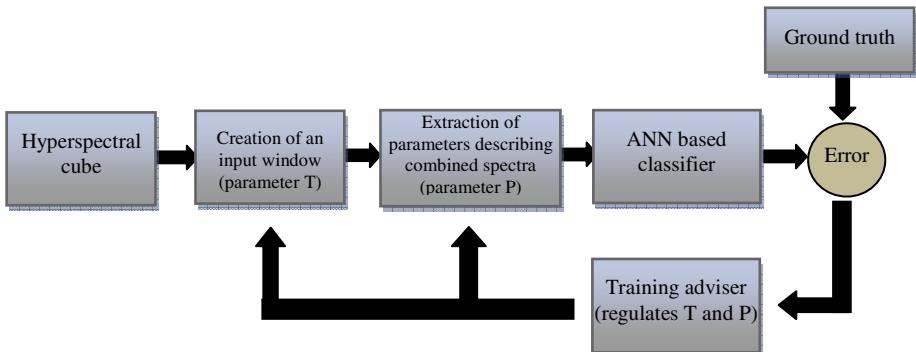


Fig. 1. Simple block diagram of the system

2.1 Data Window Size Selection and Dimensional Transformation of the Input Data

In more detail, the first step in the algorithm is a transformation of the set of multidimensional vectors from the selected data window into a new incremental representation. The vector elements resulting from this transformation are arranged so that the training procedure used in the second stage of the system can choose how many of these parameters are really necessary for the ANNs to produce an appropriate classification, that is, how much of the incrementally encoded vector is necessary.

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ be a set of vectors \mathbf{x} containing data measured at m different spatial locations sampled over a given domain under analysis. Each data vector $\mathbf{x}_i \in \mathbb{R}^B$, where B is the number of magnitudes measured at each sampled point. For our purposes, the B components of each vector can be seen also as a discrete representation of a continuous data function. We are interested in cases where both the number of samples M and the vector dimension B are high. Thus, the dimension $D = M \times B$ of the data matrix \mathbf{X} conforming the observation space is very large. Generally, a high dimensionality of the observation space arises in many applications such as climatology, image processing, multisensorial data in robotics, or hyperspectral imaging among others. Nevertheless, the features or processes under analysis can often be described by a set of parameters having a smaller dimensionality. The present approach deals with cases where dimensionality can be reduced in both the spatial distribution of samples (thus reducing dimension M) and in the vectors size (reducing B).

The classification process is an $R^D \rightarrow R$ mapping over the domain under analysis into different classes, each one described by a set of parameters and/or features. This application is defined through a training procedure that makes use of a window of T vectors taken out of the full set of M vectors in the observation space. Our first goal is to reduce the cost of the classification process by making use of a lower-dimensional embedding $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\}$ (where $\mathbf{y}_i \in \mathbb{R}^A$ with $A < B$) preserving local information so that processing the lower dimensional embedding \mathbf{Y} yields a similar classification to that from processing the original data vectors of \mathbf{X} . Generally, the embedding dimension A is not known a priori and must be guessed during the dimensionality reduction process. In the present case the reduction of the vectors' dimensionality is achieved through an incremental procedure which starts by projecting the vectors into a J dimensional with an initial value for J that is much lower than the one of the parameter space. The components of these vectors are used as inputs to ANNs that are then trained to perform the classification. The analysis of the results achieved will tell if the input vectors' dimension needs to be increased or not. Each increasing step doubles the dimension of the vectors, but preserves the existing components and just adds new ones, which is consistent with an incremental description. The procedure is explained in the following.

Let be \mathbf{F}_J a vector obtained by projecting the original vector \mathbf{x} of dimension B into a new space of lower dimension (J). Without loss of generality, we will take as starting value $J = 4$, because less than four parameters are generally too few to really characterize a vector. A component of the projected vector is $F_{i,4}$ ($i=1, \dots, 4$). An increase in vector dimension should double J , that is $J=8$, having components $F_{j,8}$ ($j=1, \dots, 8$). Nevertheless, the four components of $F_{i,4}$ are preserved in $F_{j,8}$. Thus the components of this new vector \mathbf{F}_8 should be defined as $\mathbf{F}_8 = [F_{i,4}, F_{k,8}]$ with $k=5, \dots, 8$.

For a general incremental step $\mathbf{F}_n = [\mathbf{F}_{n-1}, F_{l,n}]$ with ($l = 1+n/2, \dots, n$) and with $n \mid 2^n \in (4, B)$. The result of this process is a reduced training matrix of TxB^* (being $B^*=2^n$).

After the reduction over the vector dimension some new definitions should be introduced. Let $S(V, n)$ be a vector obtained by selecting a subvector of V such that this subvector has a size n : $S(V, n) = S / S \subset V \cap \dim(S) = n$. Let $\text{Var}(V)$ be the difference between the largest and smallest values of a vector, that is: $\text{Var}(V) = \max(V) - \min(V)$.

Let $S^*(V, n)$ be the minimal subvector of size n of a vector V , the subvector with the minimum vector variability in V of size n : $S^*(V, n) = S(V, n) / \min\{\text{Var}(S(V, n))\}$

Let $\text{Diff}(V, V')$ be the vector resulting from removing the elements of a subvector from the original vector, and let $V (S^p(V, k))$ be the k partial subvector of a vector:

$$S^p(V, k) = S^*(D_{k-2}, T/k) \text{ with } \begin{cases} D_0 = V; \\ D_n = \text{Diff}(D_{n-1}, S^*(D_{n-1}, T/k)); \\ T = \dim(V); \end{cases}$$

Vector V_b represents T values for each of the B^* and will be reduced using T incremental descriptors:

$$[s_1, s_2, s_3, s_4, s_5, \dots, s_n, \dots, s_T] \text{ with } \begin{cases} s_1 = O(V_b); \\ s_n = O(S^p(V_b, n)); \end{cases}$$

After applying this stage to different datasets, we found the need to generalize the processing through the creation of operator $O(V)$. In the case of clearly different signals, operator O performs the mean of the elements of the vector, and in the case of very similar signals it produces the typical deviation. As a result of this process, a TxB^* incremental matrix is obtained.

Finally, to arrange the parameters to used as inputs to the ANNs we establish an order to introduce or remove blocks of inputs based on a tuning parameter P , when it is necessary to increase or decrease the number of descriptors used by the classifier. Thus, after this first stage, the initial data matrices have been reduced to a more manageable dimension through a reduction based on parameters P and T .

2.2 Classification of the Categories through an ANN Structure

In order to perform the classification of the data into the categories provided by the expert, a modified RBF network was used. The objective was to allow the system to modify the number of inputs during the training procedure without the need of restarting the learning process from scratch. Consequently, if the number of inputs increases or decreases, the new resulting network benefits from the previous training, thus reducing the learning time and smoothing the transients.

Any increase in the number of inputs (number of descriptors used to describe the features of the window), implies introducing new input neurons and new synapses in the classification network. To avoid retraining from scratch, the descriptors have been chosen so as to provide a gradually more detailed view of the data. Whenever an increase of the number of parameters takes place (new inputs to the network), these parameters derive from previous lower resolution descriptors and, consequently, their influence can be smoothed by introducing neurons and weights for them that are the same as for their originators (Fig 2).

Nevertheless, in order to smooth even further the consequences of these modifications, several coefficients are introduced to gradually increase the effect of the new inputs on the hidden layers. These new attenuated inputs need to be trained in the same way as the old ones, and, consequently, the training algorithm must be modified in this direction. When the number of neurons is decreased, the redundant inputs and synaptic connections are smoothly removed and the training is adapted to its new configuration. The rate of increase or decrease of the transition coefficients values has been regulated to reduce the sharp transitory increase of the error.

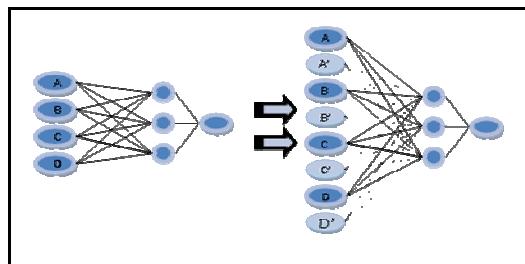


Fig. 2. Dimensionality increase resulting in new inputs to the classification network

An added bonus of using this variable size network is that it is even possible to decrease the convergence time in high dimensional search spaces. As shown in Fig. 3, we have compared the evolution of the discrimination error level with two different training modes for a hyperspectral classification problem. One uses a 12 input network during whole search process (traditional training strategy) and the second one increases the number of inputs gradually (incremental evol.). The inputs are generated using the preprocessing stage already described over a set of synthetic spectra grouped in different proportions into regular mosaics. The advantages of using the incremental approach may be seen in terms of convergence, which is much faster in the incremental case. Even though a slight worsening of the error level takes place just after changes in network, the system recovers in about 20 training steps.

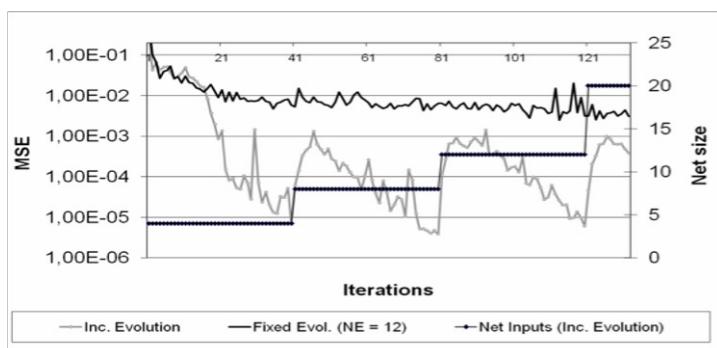


Fig. 3. Comparison of evolution of the MSE during training when using a fixed 12 input network (solid black line) or a growing network (solid light line) that starts from 4 inputs and its inputs increase as shown by the dotted line

2.3 On Line Tuning of the Window Size and Dimensional Transformation Parameters

The tuning or advisory module modifies the parameters of the first stage after evaluating the evolution of the classification error. This tuning system operates in the classification space and not in terms of other more classical parameters such as mean squared error or others. The basic ideas behind this way of evaluating are, first to measure the absolute value of the error, second to consider the stability of the error values to assure a reliable classification, and third, to make the input data set size (the parameter controlled by the adviser), as small as possible. That is, aim to use the least possible amount of information from a data set to identify the category it belongs to.

Specifically, we have defined the following measuring parameters:

- The *discrimination level* measures the average proportion of misclassified samples for a set of samples.
- The *reliability* represents the ratio of discrimination values that are lower than a particular discrimination level. Reliability is based on a discrimination level and on a given number of discrimination values, called the size of the “use window”.
- The *reliable discrimination level* (D_r) is defined as the maximum discrimination level obtained with a given reliability from a set of discrimination level data. This is basically the percentile “reliability” in the discrimination level data set. Thus, two parameters are merged into a single one which establishes the required reliability level, and smoothes the search process.
- The *spatial discrimination level* is obtained based on the input data set size (T), and the required size (T^*) and using a geometric analogy for weighing the right or wrong classification we obtain:

$$P = \min \left\{ \frac{T^*}{T} \cdot 1 + \frac{T - T^*}{T} \frac{1}{N_c}, 1 \right\} \quad \text{being } N_c \text{ the number of classification categories}$$

- Consequently, the global reliable discrimination level will be: $P_G = D_r \cdot P$ and this will be the parameter we are using to control training. So, the search will stop when this global discrimination reaches the desired level.

Based on the previous evaluation, a search algorithm is used to guide the modification of the dimensional reduction parameters and the input window size. This search is structured as a set of rules: it uses a random initial search direction and maintains it while the global error level decreases. If the error level increases, a new random advance direction is selected. In addition, in order to add some extra exploratory capabilities, a small probability of changing the direction selected for the algorithm is added. This simple search method appears to be enough for the experiments presented in the following sections and is surely enough to display the basic capabilities of the approach proposed in this paper.

3 Experimental Results

We now present one of the experimental results obtained using the whole system. In this case we are analyzing hyperspectral images. This kind of images have been widely used in different fields like non invasive inspection and control methods in industrial processes [9] or homeland security [10]. In this kind of images, the spectral information of every pixel is collected in dozens or hundreds of contiguous discrete spectral bands. The problem is extracting the spatio-spectral features required for the classification of material samples given some ground truth when the spatial resolution is high enough for each grain of a single component to be larger than the pixel size.

For these experiments we were provided with samples of granite extracted at different locations. They were scanned in a single image by a hyperspectrometer delivering data cubes with a depth of 144 spectral bands. The tests were carried out with the materials in different positions and the results were equivalent. Three different minerals occur in different proportions in granite, giving each type of granite its own color, texture and structural characteristics. In these experiments, the training was carried out using a set of ground truth labels provided by an expert for the different types of granite and the results were obtained for a required *reliability* of 85%, a *use window* of 10x10 and several values for the optimal window size. After the training and tuning process, the classifications obtained for the test sets were very satisfactory, leading to successful classifications with the required parameters.

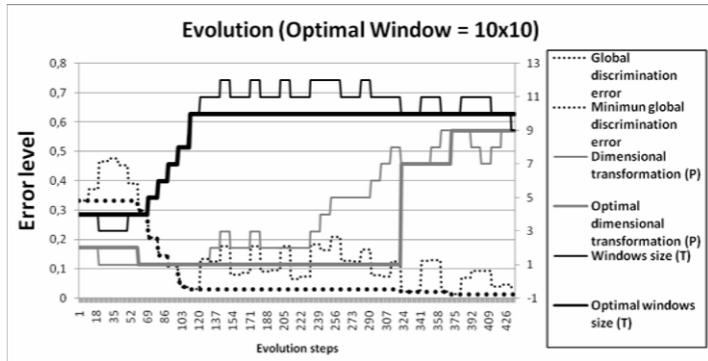


Fig. 4. Training for a process seeking an optimal window size of 10x10

In Fig. 4, a search process for which an optimal window size of 10x10 is desired is depicted. The left axis corresponds to the global discrimination error. The right axis corresponds to the values of parameters P and T. We are also including in the figure the minimum error achieved up to each evolution step and the values of P and T used to obtain this minimal value. In the first 100 steps the low initial values of T and P (4 and 2 respectively) make the discrimination error levels very high. Take into account that the discrimination values show the percentage of misclassified samples, this is, the lower the better. After this initial stage the value of T increases to a value slightly higher than the required window size and P also increases a little bit. As a consequence the discrimination improves to an average misclassification value of 1% and the global discrimination varies between values of 10 and 20% due to variations

in the window size. In the last stage the increase in the input parameters and a small decrease of the window size to the optimal value makes the global discrimination achieve lower values. The minimum global discrimination error level obtained is 1.3% corresponding to a window size of 10, a spatial discrimination level of 1.0, a parameter P of 9 (56 input neurons), an average discrimination level of 0.42% and a reliable discrimination level of 1.3% are obtained. Notice that this is a search process, and the exploratory component creates fluctuations trying to continue the search after finding good solutions in order to determine if better solutions are available.

4 Conclusions

An incremental learning algorithm for optimizing high dimensional backpropagation ANN-based classification systems has been presented. The algorithm acts during the ANN training process by automatically determining the minimum dimensions after reducing to a minimum and then progressively increasing the dimensionality of the dataset. In parallel, the ANN topology changes to adapt the input size to the data dimensions. The proposed incremental learning algorithm allows learning the new instances introduced by the changes in data and input size while preserving old memory. The algorithm makes use of a training advisor that regulates two parameters. The resulting classification system is of application to high dimensional sensorial datasets as those of hyperspetrometry or mobile robots.

Acknowledgments. This work was supported by the MEC of Spain through project CIT-370300-2007-18, DPI2006-15346-C03-01 and DEP2006-56158-C03-02.

References

1. Benediktsson, J.A., Swain, P.H., Ersoy, O.K., Hong, D.: Classification of High Dimensional Data Using Neural Networks. In: Geoscience and Remote Sensing Symp. (1990)
2. Donoho, D.: High dimensional data analysis - the curses and blessings of dimensionality. In: AMS conference Math Challenges of the 21st Century, Los Angeles, August 6-11 (2000)
3. Verleysen, M.: Learning high-dimensional data. In: NATO Advanced Research Workshop on Limitations and Future Trends in Neural Computing, Siena, Italy, pp. 22–24 (2001)
4. Jimenez, L.O., Landgrebe, D.A.: Supervised classification in high-dimensional space: geometrical,statistical, and asymptotical properties of multivariate data. IEEE Transactions on Systems, Man and Cybernetics 28(1), 39–54 (1998)
5. Ravi Kanth, K.V., Agrawal, D., Singh, A.: Dimensionality Reduction for Similarity Searching in Dynamic Databases. In: SIGMOD Conference, pp. 166–176 (1998)
6. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley Interscience, Hoboken (2000)
7. Grollman, D.H., Jenkins, O.C., Wood, F.: Discovering natural kinds of robot sensory experiences in unstructured environments. J. Field Robotics 23(11-12), 1077–1089 (2007)
8. Yairi, T.: Map Building without Localization by Dimensionality Reduction Techniques. In: Proceedings of the 24th International Conf. on Machine Learning, Corvallis, OR (2007)
9. Willoughby, C.T., Folkman, M.A., Figueroa, M.A.: Application of hyperspectral-imaging spectrometer systems to industrial inspection. In: Proc. SPIE, vol. 2599, pp. 264–272 (1996)
10. Pan, Z., Healey, G.E., Prasad, M., Tromberg, B.J.: Hyperspectral face recognition for homeland security. In: Proc. Infrared Tech. and Applications XXIX, vol. 5074, pp. 767–776 (2003)

Towards the Adaptive Control of a Multirobot System for an Elastic Hose

Zelmar Echegoyen, Alicia d'Anjou, Ivan Villaverde, and Manuel Graña*

Grupo de Inteligencia computacional,
University of the Basque Country
www.ehu.es/ccwintco

Abstract. This paper reports initial steps in the study of control strategies for a multi-robot system trying to move a flexible hose. Our starting point is the hose geometry modeling using cubic splines. The control problem is then stated as the problem of reaching a desired configuration of the spline control points from an initial configuration. The control of the hose by the multi-robot system is first solved neglecting the hose internal dynamics. This is the simplest ideal case. Then we give a model for the internal dynamics. Future and on going work is addressing the definition of a control system that takes into account the hose internal dynamic constraints.

1 Introduction

Nowadays robotic systems are facing the challenge of working in very unstructured environments, such as shipyards or construction sites. In these environments, the tasks are non repetitive, the working conditions are difficult to be modeled or predicted, and the size of the spaces is huge. In these environments, a common task is the displacement of some kind of flexible hose. It can be a water hose or a power line, or other. We are interested here in the design of a control architecture for a multi-robot system dealing with this problem. A collection of cooperating robots attached to the hose must be able to displace it to a desired configuration. We have identified the following sub-problems: modeling a flexible elongated object, distributed sensing on the robots to obtain information of the environment and/or of the configuration of the system including robots and the hose, inverse kinematics of the whole system, stable structural design, highly adaptive control via high level cognitive mechanisms. Here we focus on the hose modeling and the generation of control strategies for a collection of autonomous robots attached to it. Section 2 contains the description of the hose geometrical model. Section 3 gives the solution for the hose control in the simplest case, neglecting its internal dynamics. In section 4 we introduce the model of the hose internal dynamics that may be the base for our future work described in section 5.

* The Spanish Ministerio de Educacion y Ciencia supports this work through grant DPI2006-15346-C03-03.

2 Hose Geometrical Model

Modeling uni-dimensional objects has great application for the representation of wires in industry and medicine. The most popular models use differential equations [2], rigid body chains [6] and spring-mass systems [27]. Spring-mass systems and rigid body chains allow to simulate a broad spectrum of flexible objects, and they are rather versatile when simulating deformations. They are very fast to compute. However they are imprecise for uni-dimensional object modeling. The combination of spline geometrical modeling and physical constrains was introduced by [3]. We have chosen the Geometrically Exact Dynamic Splines (GEDS) [1], because they allow a continuous definition of the hose that accounts for the rotation of the transverse section at each point in the curve, and that an exhaustive and rigorous mechanical analysis has been developed. A spline is a piecewise polynomial function. See figure 1 for an illustration.

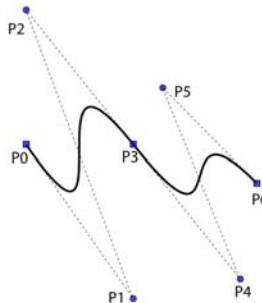


Fig. 1. Cubic spline

Splines define a curve by means of a collection of Control Points, which define a function that allows to compute the whole curve. In order to reduce the interpolation error, the number of Control Points can be increased. When modeling a hose, we assume that it has a constant sectional diameter, and that the transverse sections are not deformed in any way. If we do not take into account the hose internal dynamics, an spline passing through all the transverse section centers suffices to define the hose, as can be appreciated in figure 2. If we want to take into account the hose internal dynamics, we need also to include the hose twisting at each point given by the rotation of the transverse section around the axis normal to its center point, in order to compute the hose potential energy induced forces. In the GEDS model, the hose is described by the collection of transverse sections. To characterize them it suffices to have: The curve given by the transverse section centers $c(t) = (x(t), y(t), z(t))$, and the orientation of each transverse section $\theta(t)$. This description can be summarized by the following notation: $q(t) = (c, \theta) = (x, y, z, \theta)$. In figure 2, vector t represents the tangent to the curve at point $c(t)$, and vectors n and b determine the angle angle $\theta(t)$ of the transverse section at point $c(t)$.

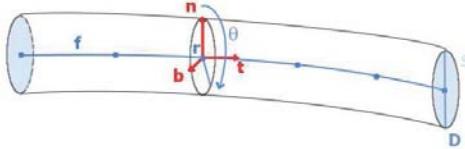


Fig. 2. Hose section

The hose mathematical representation is given by a collection of polynomial splines, where each spline is defined as:

$$q(u) = \sum_{i=1}^n b_i(u).P_i \quad (1)$$

where $b_i(u)$ are the basis function associated to the control point P_i , $u \in [0, L]$ and s the arc-length.

3 Basic Control Problem: No Internal Dynamics

The goal is the positioning of the hose by the positioning of several autonomous robots $\{r_1, \dots, r_m\}$ attached to it. Initial and final hose positions are given, as well as the initial robot positions. In figure 3 it can be appreciated the problem configuration, with a hose described by parametric cubic splines with control points P_i and a collection of robots r_j attached to it. Let it be:

- $q_0(u)$ the initial spline representing the hose, specified by the positions of the control points P_0 .
- $q_*(u)$ the desired spline configuration, specified by the positions of the control points P_* .
- $R_0 = \{q_0(u_{r_1}), \dots, q_0(u_{r_m})\}$ the robot initial positions.
- $l(u)$ The hose internal dynamics constraints.

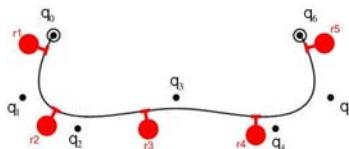


Fig. 3. Control points and positions of the robots

We are interested in obtaining the motion of the attached robots, given by instantaneous velocities of the hose attachment points \dot{R} , that will bring the hose from the initial configuration $q_0(u)$ to a desired configuration $q_*(u)$ with an initial configuration of the robots R_0 so that the $l(u)$ constraints hold at all times during this transition.

3.1 Robots Attachment Points Velocity

We compute the partial derivative of a point $q(u)$ in the curve as a function of the control point P_i :

$$\frac{dq(u)}{dP_i} = b_i(u) \frac{dP_i}{dP_i} = b_i(u) \quad (2)$$

Defining the Jacobian matrix J_{rq} as the robots contact points with the hose as a function of the control points, we have:

$$J_{RP} = \begin{pmatrix} \frac{\partial q(u_{r_1})}{\partial P_1} & \dots & \frac{\partial q(u_{r_m})}{\partial P_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial q(u_{r_1})}{\partial P_n} & \dots & \frac{\partial q(u_{r_m})}{\partial P_n} \end{pmatrix} = \begin{pmatrix} b_1(u_{r_1}) & \dots & b_1(u_{r_m}) \\ \vdots & \ddots & \vdots \\ b_n(u_{r_1}) & \dots & b_n(u_{r_m}) \end{pmatrix} \quad (3)$$

Being u_{r_j} the attachment point of the robot r_j to the hose. We obtain the following expression which gives the relation between the control points velocity \dot{P} and the robot attachment points velocity \dot{R} :

$$\dot{R} = J_{RP} \cdot \dot{P} \quad (4)$$

To obtain the evolution $P^{k+1} = f(P^k)$ that decreases gradually the distance between the control points and their desired positions $\|P - P_*\|$, we minimize the following Lagrangian function:

$$L(P, \lambda) = \sum \|P - P_*\| + \lambda \cdot l(u)$$

We define the following iterations that implement the gradient descent of the Lagrangian function:

$$\begin{aligned} P^{k+1} &= P^k + \Delta P^k \\ \lambda^{k+1} &= \lambda^k + \Delta \lambda^k \end{aligned}$$

The increments Δq^k and $\Delta \lambda^k$ for the step $k + 1$ are obtained solving the following equations system:

$$\nabla^2 L(P^k, \lambda^k) \begin{pmatrix} \Delta P^k \\ \Delta \lambda^k \end{pmatrix} = -\nabla L(P^k, \lambda^k) \quad (5)$$

If we use the jacobian J_{rq} defined at equation 4, we can determine the motion of the robots contact points in the spline. We approximate at each step the robots contact points variations:

$$\Delta R^k = J_{rq} \cdot \Delta P^k.$$

4 Modeling the Hose Internal Dynamics

The control equation 4 does not take into account neither the hose internal energy nor the external forces acting on it. It is necessary to determine the force that

will be generated in the hose as a consequence of its energy configuration. We need also to determine the external forces that try to predict the results of the interaction. Work in this section is based on the paper on Geometrically Exact Dynamic Splines [1]. The relation between the energy and the force, it is defined by the Lagrange equations [6] using the control points as the degrees of freedom, because they define completely the spline curve and the transverse section orientation:

$$\frac{d}{dt} \left(\frac{\delta T}{\delta \dot{P}_i} \right) = F_i - \frac{\delta U}{\delta P_i}, i \in 1, \dots, n \quad (6)$$

The Lagrange equations use the potential energy U and the system's kinetic energy T . The kinetic energy is the motion energy, while the potential energy is the energy stored because of the hose position. F is the model of the external forces acting on the hose. We can assume that the mass and stress are homogeneously distribute among the n degrees of freedom of the hose, where P_i , $i \in 1, \dots, n$, are the spline control points.

4.1 Potential Energy

In figure 4 we can appreciate the forces and torques $\mathcal{F} = (\mathcal{F}_S, \mathcal{F}_T, \mathcal{F}_B)^t$ that deformate the hose and perform some influence on its potential energy. The stretching force, \mathcal{F}_s , is the force normal to the hose transverse section and its application results in its lengthening. The tension torque, \mathcal{F}_T , makes the transverse section to rotate around the kernel curve. The curve torquing, \mathcal{F}_B , modifies the orientation of the transverse section. The forces acting on the transverse section plane are neglected, because we accept the Kirchhoff assumption that considers that the transverse sections are rigid and that only the hose curvature may be distorted. Forces \mathcal{F} are proportional to the tension ϵ , where ϵ_0 is the tension in a repose state. If we consider an small tension, appropriate for a curvature radius relatively big compared against the radius of the transverse section, we can assume that there is linear elasticity, so that the computations are simplified and we can state the next equation, which derives from the Hooke law:

$$\mathcal{F} = H(\epsilon - \epsilon_0) = \begin{pmatrix} ES & 0 & 0 \\ 0 & GI_0 & 0 \\ 0 & 0 & EI_s \end{pmatrix} (\epsilon - \epsilon_0)$$

where I_0 is the inertial polar motion, I_s the inertia transverse section moment, ES stretching rigidity, GI_0 torsion rigidity, EI_s curve rigidity and H the Hooke matrix. The potential energy, U , is composed of the tension and gravitational energies, and has references to elasticity. The tension vector ϵ , is composed of stretching tension ϵ_S , the torsion tension ϵ_T and the curve tension ϵ_b . Besides, we define a Hooke matrix, H , which is derived from the Hooke's situation. Assuming that the transverse section is circular and diameter curve, D , is constant, the potential energy is very determined by the following expression,

$$U = \frac{1}{2} \int_0^L \epsilon^t H \epsilon ds.$$

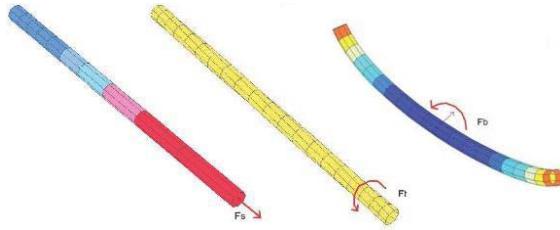


Fig. 4. Forces induced by Potential energy of the hose

4.2 Kinetic Energy

Because the hose is defined by its position and rotation at each curve point, the kinetic energy T includes the translation and rotation energies. The translation energy corresponds to the control points displacement, while the rotation energy is due to the rotation of the transverse sections. Defining J as the inertia matrix, invariant over all spline points, because the hose diameter is constant everywhere.

$$J = \begin{pmatrix} \mu & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & I_0 \end{pmatrix}$$

The spline kinetic energy T is defined by the following equation:

$$T = \frac{1}{2} \int_0^L \frac{dq^t}{dt} J \frac{dq}{dt} ds$$

Where μ is the lineal density and I_0 is the polar inertia moment.

4.3 External Forces on the Control Points

Taking into account the potential and kinetic energy in the Lagrange equations, and we substitute q by the expression in eq. 11, we obtain:

$$\frac{d}{dt} \frac{\partial T}{\partial P_i} = \sum_{j=1}^n J \int_0^L (b_i(s) b_j(s)) ds \frac{d^2 P_j}{dt^2} \quad (7)$$

To simplify the expression, we define $M = J \int_0^L (b_i(s) b_j(s)) ds$ and $A = \frac{d^2 q_j}{dt^2}$, so that the expression in eq. 6 becomes:

$$\frac{d}{dt} \frac{\partial T}{\partial P_i} = \sum_{j=1}^n M_{i,j} A_j \quad (8)$$

From the hose energy, we aim to determine the component forces of the right term of the Lagrange equation 6, finding the the potential energy derivative as a function of the control points:

$$P^i = -\frac{\partial U}{\partial P_i} = -\frac{1}{2} \int_0^L \frac{\partial(\epsilon - \epsilon_0)^t H(\epsilon - \epsilon_0)}{\partial P_i} ds \quad (9)$$

Using equations 8 and 9 we can write the Lagrange equation 6 as a matrix equation.

$$MA = F + P \quad (10)$$

The four subsystems for x , y , z and θ are independent.

4.4 External Forces on the Robot Contact Points

We use the equation 5 to determine the motion of the spline control points ΔP^k at each gradient minimization step. We differentiate the spline control points relative to time $\frac{\partial(\Delta P^k)}{\partial t}$ and we introduce it in the equation 10 in order to get the forces F_i that must be applied on each control point to reach the desired hose configuration. At each hose control step it is needed to obtain the forces that the robots must apply so that the hose has the desired acceleration at the control points. Because the dynamic is continuously defined over all the spline, a force F applied on a particular point produces the generalized forces F_i over the spline control points. When differentiating the power $W = Fq$ respect to the spline control point P_i , the corresponding generalized force F_i is obtained:

$$\frac{\partial W}{\partial P_i} = F \frac{\partial q}{\partial P_i} = F b_i$$

So, after obtaining the forces F_i that must be applied over the control points, we have to determine the forces that the robots must exert over the hose.

$$F_r = J_{RP} F_i \quad (11)$$

After having the robot required forces, F_r , the control task is to obtain the robot accelerations allowing these forces.

5 Conclusions and Future Work

The work done up to now has been focused on the study and modeling of the hose, with special emphasis to determine the internal dynamic of the hose and the application of external forces. Our future work is to extend the work on the hose control taking into account the internal energy and induced dynamics, and the consequent constraints. Work must be done on the study of the control system stability. We will perform simulations before trying the implementation of the system in real robots under controlled circumstances. For that step a calibration technique to estimate the hose parameters will be needed. The work up to this point assumes the existence of a global observer and that the computational process and decision is made in a centralized server or unit. As a long term research endeavor we aim to study distributed sensing and control realizations of the system.

References

1. Theetten, A., Grisoni, L., Andriot, C., Barsky, B.: Geometrically Exact Dynamic Splines, Institut National de Recherche en Informatique et en Automatique - INRIA (2006)
2. Interactive simulation of one-dimensional flexible parts, DaimlerChrysler (2007)
3. Qin, H., Terzopoulos, D.: D-NURBS: A Physics-Based Geometric Design, Department of Computer Science, University of Toronto (1996)
4. Smooth constraints for spline variational modeling, University of Lille, France (2004)
5. Pai, D.K.: STRANDS: Interactive simulation of thin solids using Cosserat models. In: Computer Graphics Forum 2002 (2002)
6. Real-time Virtual Cables Based On Kinematic Simulation. In: Proceedings of the WSCG 2000 (2000)
7. A Virtual Environment for Interactive Assembly Simulation: From Rigid Bodies to Deformable Cables. In: 5th World Multiconference on Systemics, Cybernetics and Informatics (2001)
8. Peterson, J.W.: Arc Length Parameterization of Spline Curves. Research report, Taligent, Inc.

Economical Implementation of Control Loops for Multi-robot Systems

Jose Manuel Lopez-Gude^{1,*}, Manuel Graña², Ekaitz Zulueta¹,
and Oscar Barambones¹

Computational Intelligence Group UPV/EHU

¹ System Engineering and Automatic Dept., University College of Engineering of Vitoria,
Basque Country University, C/Nieves Cano 12, 00106 Vitoria, Spain

² Computer Science and Artificial Intelligence Dept., Faculty of Computer Engineering, Basque
Country University, C/Paseo Manuel de Lizardizábal 1, 20018 San Sebastián, Spain
{jm.lopez,manuel.grana,ekaitz.zulueta,oscar.barambones}@ehu.es
www.ehu.es/ccwintco

Abstract. In spite of the multiple advantages that multi-robot systems offer, to turn them into a realistic option and to get their proliferation, they must be economically attractive. Multi-robot systems are composed of several robots that generally are similar, so if an economic optimization is done in one of them, such optimization can be replicated in each member. In this paper we deal with the economic optimization of each control loops of the subsystems that each robot must control individually. As the subsystems can be complex, we propose to use a Predictive Control modeled by Time Delayed Neural Networks and implemented using very low cost Field Programmable Gate Arrays.

1 Introduction

In this paper we show a work to implement low level controllers with small computational needs that can be used in each of the subsystems that must be controlled in each of the robots that belongs to a multi-robot system. If a robot is in a multi-robot system that robot needs bigger computational capacity, because it has to do some tasks derived from being in the team, for example, coordination and communication with the remaining members of the team. Besides, occasionally, it has to deduce cooperatively the global strategy of the team. One of the theoretical advantage of multi-robot systems is that the cost of the team must be lower than the cost of a single robot with the same capabilities. To become this idea true it is mandatory that the cost of each member was under a certain value, what we can get if each of them is equipped with very cheap computational systems. One of the cheapest and more flexible devices for control systems implementation are Field Programmable Gate Arrays (FPGAs). If we could implement a control loop using a very simple FPGA structure, the economic cost of each of them could be about 10 dollars.

* This work was supported in part by the Spanish Ministerio de Educación y Ciencia under grant DPI2006-15346-C03-03.

On the other hand, and under a pessimistic vision, the subsystems to control could have problems to be controlled using classic and well known control schemas as PID controllers. In this situation we can use other advanced control systems which try to emulate the human brain, as Predictive Control. This kind of control works using a world model and calculating some predictions about the response that it will show under some stimulus, and it obtains the better way of control the subsystem knowing which is the desired behavior from this moment until a certain instant later. The predictive controller tuning is a process that is done using analytical and manual methods. Such tuning process is expensive in computational terms, but it is done one time and in this paper we don't deal with this problem. However, in spite of the great advantage of predictive control, which contributes to control systems that the classic control is unable to do, it has a great drawback: it is very computationally expensive while it is working. In section 4 we will revise the cause of this problem. A way of avoiding this drawback is to model the predictive controller using neural networks, because once these devices are trained they perform the calculus at great speed and with very small computational requirements, and at the same time, we can implement them using very cheap FPGA devices. In this paper we propose a learning model to be used with Time Delayed Neural Networks, so once the neural network is trained, the neuronal predictive controller is ready and it responds properly showing its generalization capabilities in environments that it hasn't seen in the training phase. This way we could get a very cheap implementation of each of the control loops that each robot of the multi-robot team needs, avoiding the rise of the total cost of the team. In the literature there are several sources indicating that each robot of a multi-robot system must be as cheap as possible. There is a quantitative support for the argument that larger teams of less-reliable and cheaper robots can perform certain missions more reliably than smaller teams of more-reliable robots [1]. There are examples of using very cheap discrete components. In [2] very cheap sensorless nodes are used to support a complex multi-robot foraging task. On the other hand, in [3] a kind of sensors is used because they became cheaper than others. In [4], the components of the developed system consume energy provided by microcontroller's I/O ports, are cheap and available on micro-component market. Besides the use of individual components, [5] integrate economic and technical issues into an unified engineering design framework for the manufacturers of robots.

Section 2 gives a summary of the objective of the work of this paper. Section 3 gives background information about Predictive Control and a technique called Dynamic Matrix Control, and about a kind of neural nets called Time Delayed Neural Networks. Section 4 discusses a concrete case study and the results that we obtain. Finally, conclusions are covered in section 5.

2 Objective

The main objective of this paper is to get cheap implementation of low level control loops that could be used by each member of a multi-robot system. To get this objective Time Delayed Neural Networks are used to model predictive controllers, because these can control subsystems that classics controllers can't.

3 Background

This section gives a brief introduction about a general technique called Model Predictive Control, and about a concrete technique called Dynamic Matrix Control. We also present a brief introduction about Time Delayed Neural Networks.

3.1 Model Predictive Control (MPC) and Dynamic Matrix Control (DMC)

Model Predictive Control (MPC) is an advanced control technique used to deal with systems that are not controllable using classic control schemas as PID. This kind of controllers works like the human brain in the sense that instead of using the past error between the output of the system and the desired value, it controls the system predicting the value of the output in a sort time, so the system output is as closer as possible to its desired value for these moments. Predictive Control isn't a concrete technique. It's a set of techniques that have several common characteristics: there is a world model that is used to predict the system output from the actual moment until p samples, a objective function that must be minimized and a control law that minimizes the objective function. The predictive controllers follow these steeps:

- Each sampling time, through the system model, the controller calculates the system output from now until p sampling times (prediction horizon), which depends on the future control signals that the controller will generate.
- A set of m control signals is calculated optimizing the objective function to be used along m sampling times (control horizon).
- In each sampling time only the first of the set of m control signals is used, and in the next sampling time, all the process is repeated again.

The technique called Dynamic Matrix Control (DMC) is a concrete MPC algorithm that uses:

- As subsystem model, the step response of the subsystem,
- As objective function, it measures the difference between the reference signal and the subsystem output,
- As control law, $\Delta u = (G' G + \lambda I)^{-1} G' (w - f)$, being G a matrix that contains the systems dynamics, λ a parameter about the following capacity of the subsystem, w the reference signal and f the free response of the subsystem.

To learn more about Predictive Control in general, and about Dynamic Matrix Control in particular, see [6-9].

3.2 Time Delayed Neural Networks

Time Delayed Neural Networks (TDNN) are a kind of multi-layer perceptron neural networks. The TDNN special feature is that they are a kind of dynamic neural networks, because delayed versions of the input signals are introduced to the input layer. Due to this, the outputs don't depend only on the actual values of the signals, they depend on the past value of the signals too. This kind of neural network can be trained

using the Backpropagation algorithm or the Generalized Delta Rule. In the experiments that we show in this paper the Levenberg-Marquardt method has been used. To learn more about neural networks in general see [10-12]. To learn more about Time Delayed Neural Networks, see [13].

4 Case Study

In this case study we will show an example of how we can get an advanced control of robot subsystems using neural networks. We are going to suppose that the model of a subsystem is described by the following discrete transfer function:

$$H(z) = \frac{1}{z - 0.5} . \quad (1)$$

Although it is a stable subsystem, its response is unstable if we try to control it using a PID controller tuned through classic and well-known methods as Ziegler-Nichols.

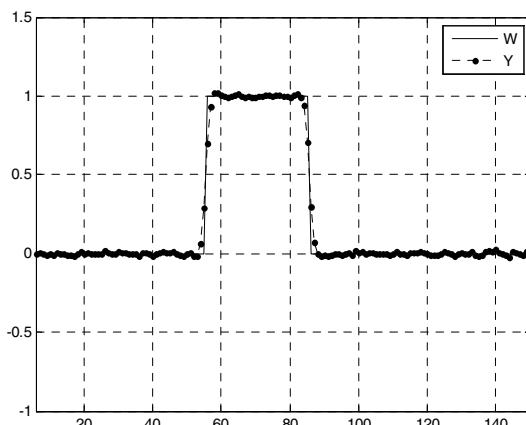


Fig. 1. Control of a robot subsystem using Predictive Control when the reference is a step, with the values of the parameters $p = 5$, $m = 3$ y $\lambda = 1$

However, using a properly tuned DMC predictive controller, for example, with the values for its parameters $p = 5$, $m = 3$ y $\lambda = 1$, a right control is obtained, as it is shown in Fig. 1.

To get this control it has been mandatory to tune the DMC controller. This phase is very expensive in computationally terms, but it's carried out only one time. However, the computational requirements of DMC controller are great when it's in its working phase, due to the operations that it must perform to get the control law, and although it obtains set of m control signals, only first of them is used in this sample time, the rest are ignored. Because of this, it would be convenient to have a mechanism that could implement such controller requiring less computational power. Besides, it may

be necessary to control several subsystems of this kind in each robot of the multi-robot team. An alternative to get this is to use neural networks, and more precisely, Time Delayed Neural Networks, because, as the rest of neural networks, they are very fast and they have the ability of generalizing their responses.

To implement a predictive controller using a neural network we have done training experiments with multiple structures, varying two structural parameters: the number of the hidden layer neurons h and the number of delays of the time delay line d . It has been used the Levenberg-Marquardt method to carry out the training of each structure, and the training model has consisted of a target vector $P = [w(k), y(k), \Delta u(k-1)]'$ and an output $\Delta u(k)$. As it has been shown in Fig. 2 and Fig. 3, the control of the neuronal controller is right even with noisy references that hadn't been used in the training phase.

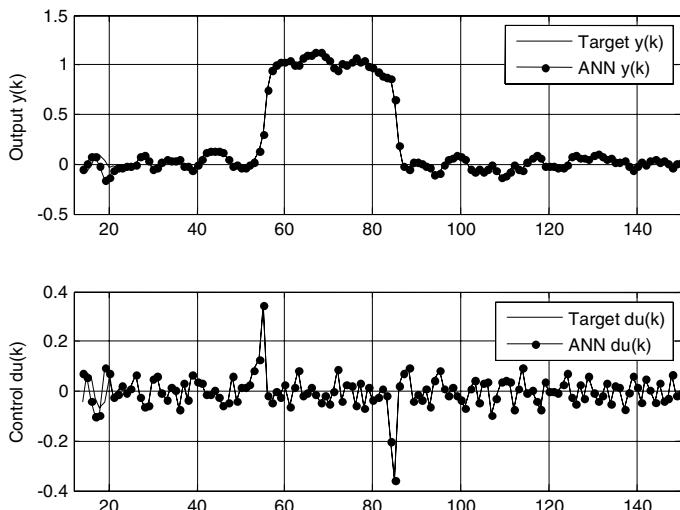


Fig. 2. Control of a robot subsystem with a Time Delayed Neural Network with a time delay line of $d = 7$ delays in the input, and $h = 5$ neurons in the hidden layer. The reference to follow is a signal that the neural network hasn't seen in the training phase.

To implement these predictive controllers using neural networks we have chosen FPGA devices. We have used a device commercialized by Altera Corporation, the EPF10K70 device, in a 240-pin power quad flat pack (RQFP) package. We way that we have used to implement the neural network in this device is to describe the behavior of that neural network using VHDL language, including in the entity that is in this description the same inputs and outputs that the neural network has. VHDL is a description language used to describe the desired behavior of circuits and to automatically synthesize them through specific tools.

To learn more about identification and control of dynamical systems, see [14-15], and about neural identification applied to predictive control, see [16-17].

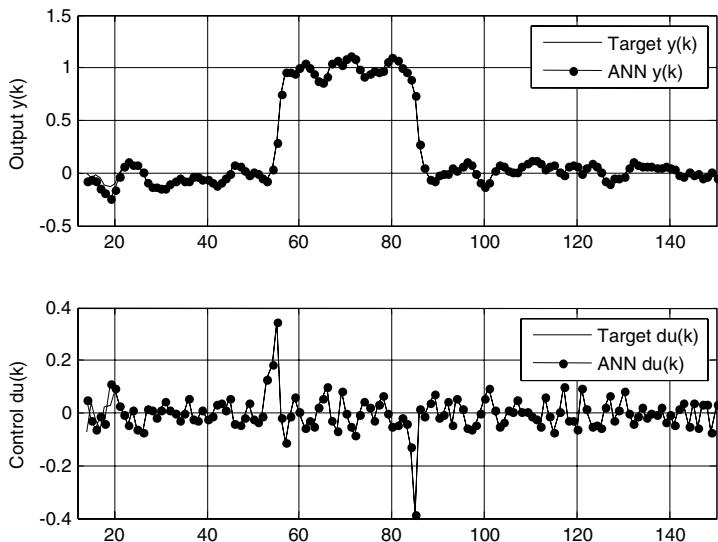


Fig. 3. Control of a robot subsystem with a Time Delayed Neural Network with a time delay line of $d = 7$ delays in the input, and $h = 5$ neurons in the hidden layer. The reference to follow is a signal that the neural network hasn't seen in the training phase.

5 Conclusions

This paper has started thinking about the convenience that the computational capacity of robots that belong to multi-robot systems was devoted exclusively to high level functions they have to perform due to being a member of such system. However, each robot must have so many internal control loops as subsystems, and in some cases they aren't controllable through classic techniques. In these cases, predictive control is a good option, but it's computationally expensive. In this paper it has been shown how the predictive controllers can be modeled using Time Delayed Neural Networks, which implementation is very cheap using very low cost FPGAs. This way we can reduce de price of each member of multi-robot system, because the investment in computational capacity must cover only the high level functions, ignoring the subsystems that it had, which are solved with very low cost FPGAs.

References

1. Stancliff, S.B., Dolan, J.M., Trebi-Ollennu, A.: Mission Reliability Estimation for Multi-robot Team Design. In: IEEE International Conference on Intelligent Robots and Systems, pp. 2206–2211 (2006)
2. O'Hara, K.J., Balch, T.R.: Pervasive Sensor-less networks for cooperative multi-robot tasks. In: 7th International Symposium on Distributed Autonomous Robotic Systems (DARS 2004), pp. 305–314 (2007)

3. Wu, H., Tian, G., Huang, B.: Multi-robot collaborative localization methods based on Wireless Sensor Network. In: IEEE International Conference on Automation and Logistics, pp. 2053–2058 (2008)
4. Kornienko, S., Kornienko, O., Levi, P.: Minimalistic approach towards communication and perception in microrobotic swarms. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2228–2234 (2005)
5. Andrews, B.W., Passino, K.M., Waite, T.A.: Social Foraging Theory for Robust Multi-agent System Desing. *IEEE Transactions on Automation Science and Engineering* 4(1), 79–86 (2007)
6. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer, London (2004)
7. Camacho, E.F., Bordons, C.: Model Predictive Control in the Process Industry. Springer, London (1995)
8. Maciejowski, J.M.: Predictive Control with Constraints. Prentice Hall, London (2002)
9. Sunan, H., Kok, T., Tong, L.: Applied Predictive Control. Springer, London (2002)
10. Braspennning, P.J., Thuijsman, F., Weijters, A.J.M.M.: Artificial Neural Networks. Springer, Berlin (1995)
11. Chester, M.: Neural Networks. Prentice Hall, New Jersey (1993)
12. Widrow, B., Lehr, M.A.: 30 Years of Adaptative Neural Networks: Perceptron, Madaline, and Backpropagation. *Proceedings of IEEE* 78(9), 1415–1441 (1990)
13. Huang, B.Q., Rashid, T., Kechadi, M.T.: Multi-Context Recurrent Neural Network for Time Series Applications. *International Journal of Computational Intelligence* 3(1), 45–54 (2006)
14. Narendra, K.S., Parthasarathy, K.: Indentification and Control of Dynamical Systems Using Neural Networks. *IEEE Tran. Neural Networks* 1(1), 491–513 (1990)
15. Norgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural Networks for Modelling and Control of Dynamic Systems. Springer, London (2003)
16. Arahal, M.R., Berenguel, M., Camacho, E.F.: Neural identification applied to predictive control of a solar plant. *Control Engineering Practice* 6, 333–344 (1998)
17. Huang, J.Q., Lewis, F.L., Liu, K.: A Neural Net Predictive Control for Telerobots with Time Delay. *Journal of Intelligent and Robotic Systems* 29, 1–25 (2000)

Part XIII

**WORKSHOP: Hybrid and Adaptive
Systems for Real-Time Robotics
Vision and Control**

An Improved Modular Neural Network Model for Adaptive Trajectory Tracking Control of Robot Manipulators

Dimitrios Alexios Karras

Chalkis Institute of Technology, Dept. Automation and Hellenic Open University,
Rodu 2, Ano Iliupolis, Athens 16342, Greece
dakarras@ieee.org, dakarras@teihal.gr, dakarras@usa.net

Abstract. A novel approach is presented for adaptive trajectory tracking of robot manipulators using a three-stage hierarchical neural network model involving Support Vector Machines (SVM) and an adaptive unsupervised Neural Network. It involves a novel adaptive Self Organizing feature map (SOFM) in the first stage which aims at clustering the input variable space into smaller sub-spaces representative of the input space probability distribution and preserving its original topology, while rapidly increasing, on the other hand, cluster distances. Moreover, its codebook vector adaptation rule involves m-winning neurons dynamics and not the winner takes all approach. During convergence phase of the map a group of Support Vector Machines, associated with its codebook vectors, is simultaneously trained in an online fashion so that each SVM learns to positively respond when the input data belong to the topological sub-space represented by its corresponding codebook vector, taking into account similarity with that codebook vector. Moreover, it learns to negatively respond to input data not belonging to such a previously mentioned corresponding topological sub-space. The proposed methodology is applied, with promising results, to the design of a neural-adaptive trajectory tracking controller, by involving the computer-torque approach, which combines the proposed three-stage neural network model with a classical servo PD feedback controller. The results achieved by the suggested hierarchical SVM approach are favorably compared to the ones obtained by traditional (PD) and non-hierarchical neural network based controllers.

1 Introduction

A basic feature of Self-Organizing Feature Maps (SOFM) is their capability to cluster functional spaces into subspaces with preservation of their topological attributes [1]. It is, also, known that Artificial Neural Networks (ANNs) and especially Multilayer Perceptrons (MLP) and Radial Basis Functions (RBF) have the theoretical ability to approximate arbitrary nonlinear functions of many input variables [1]. Although MLPs and RBFs have been successfully employed in function approximation tasks, on the other hand, several drawbacks have been revealed in their application, such as slow convergence in difficult tasks and non guaranteed generalization capability. Therefore, an improvement could be expected in such ANN performance if task

decomposition and simplification could be achieved by clustering difficult tasks functional spaces through involving properly trained SOFM models. This principle underlies the herein presented novel nonlinear controller architecture. In addition, in order to overcome the above discussed drawbacks, and especially the problem of non guaranteed generalization capability, more powerful ANN models have emerged, namely, Support Vector Machines (SVMs). SVM models, introduced by Vapnik in 1992 [1], are now involved in many different classification tasks with success. Few research efforts, however, have employed them in nonlinear regression and control tasks.

Robot manipulators have become increasingly important in the field of flexible automation but they are subject to structured and/or unstructured uncertainties. ANN models and especially MLPs and RBFs have been used for the construction of Neural - Adaptive Controllers to cope with both types of uncertainty [2] following different approaches. The research line herein followed for designing a hierarchical nonlinear compensator using ANNs is in principle based on the computed torque method proposed in [3], where the ANNs were used to compensate for nonlinearities of the robotic manipulator rather than to learn its inverse dynamics. Another method for the direct control of robot manipulators using ANNs was proposed in [4]. In that reference, the control system consists of an inverse model of the robot dynamics which produces the forces/torques to be applied to the robot, given desired positions, velocities and accelerations and a neural controller generating a correcting signal. Also, another approach is to combine an ANN model with a servo PD feedback controller [5]. This approach, illustrated in Figure 1, provides on - line learning of the inverse dynamics of the robot. In this scheme the manipulator's inverse - dynamics model is replaced by generic neural network models, one per joint, each neural network model adaptively approximating the corresponding joint's inverse dynamics. Other similar approaches are presented in [6-8]. The herein presented paper follows the line of research of [5] by investigating a novel hierarchical neural network based nonlinear controller which acts as a PD torque estimation compensator

2 A Three Stage Hierarchical Model Based on an Adaptive SOFM Scheme and Positive/Negative Learning SVM in the Computed Torque Method

It is herein proposed that ANN performance in control tasks could be improved through a three-stage process employing an adaptive regularized SOFM and a group of SVMs associated with its codebook vectors, which learn to positively respond to input vectors associated with these codebook vectors and to negatively respond otherwise. The SOFM algorithm could be applied to the available training input vectors of the function approximation/ control problem in order to quantize its corresponding input space defined by the input variables in smaller topology preserving regions by involving suitable codebook vectors as their representatives, whose distribution is representative of the joint probability distribution of these input variables.

This unsupervised process results in determining the associated codebook vectors, whose number has been predefined by the user, let's say $M \times M$. During convergence of this process and at its (t) iteration, the $m(t) \times m(t)$ codebook vectors associated to the number $NE(t)$ of neighbouring output units at instance t are updated in accordance

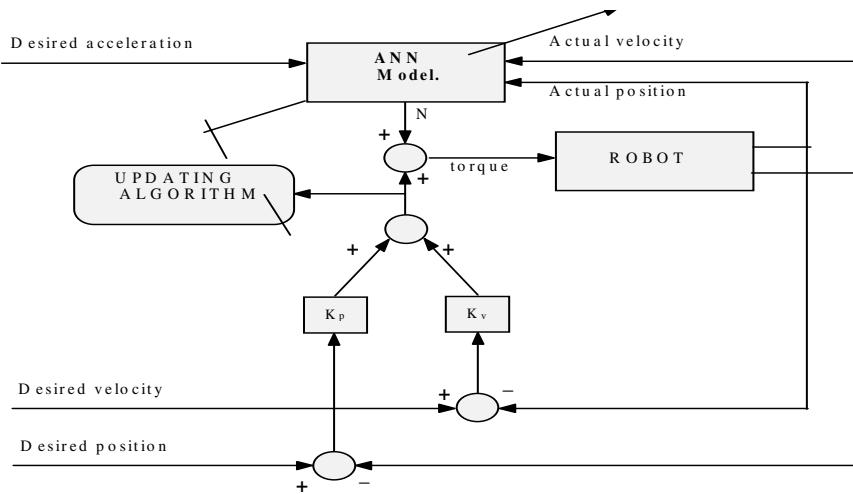


Fig. 1. Neuro-controller based computed torque architecture

with the classical SOFM algorithm. Then, the weights of their corresponding $NE(t)$ group of $m(t) \times m(t)$ SVMs are, also, updated. Thus, each such SVM is iteratively being specialized, simultaneously with its associated codebook vector, to respond when the input vector belongs to the topological subspace represented, at the end of the convergence process, by this codebook vector.

Such an attempt to specialization, however, would be incomplete since any of the above mentioned SVM learns to respond to any input vector and not only to the specifically wanted input vectors coming from the SOFM based quantized sub-space associated with the corresponding codebook vector. It is herein, proposed that the required SVM specialization trait could be achieved by inducing, in the SVM training process, input vectors coming from different subspaces and imposing a large negative number $D(t)$ as their respective desired outputs. Such a concept makes possible that the corresponding SVM learns to respond negatively to input vectors not coming from its associated specialized input space. It should be herein noted that as the SOFM convergence moves forward and $NE(t)$ decreases, then, $D(t)$ becomes larger in order to illustrate such a specialization process. This could be achieved by considering, at the t iteration $-e^{Dt}$ as such a desired output (D is a user defined constant) during the corresponding SVM training process involving irrelevant input vectors (the ones not coming from the specific corresponding input sub-space).

Apart from the previously depicted positive/ negative SVM learning concept, a novel variation of the SOFM algorithm is herein proposed, where we attempt to increase the distances between different clusters of the input space probability distribution so as to facilitate convergence of the map and achieve higher resolution of SOFM input space in order to feed the second/third stage SVMs with groups of input vectors of better quality, in terms of homogeneity. It is herein suggested that such a goal could be attained through adapting not only the winning neuron and its neighboring neurons $NE(t)$ weights but, also, loosing neurons weights. Loosing neurons weights are adapted in a manner similar to that of LVQ, by increasing the distance between

these weights vectors and the corresponding input data vectors. More specifically, all neurons j outside the winning neuron neighborhood $NE_i(t)$, with associated codebook vector $\mathbf{W}_{win}(t)$ at iteration t , are updated by the following formula:

$$\mathbf{W}_j(t+1) = \mathbf{W}_j(t) - [b(t) \exp\|(\mathbf{W}_{win}(t) - \mathbf{W}_j(t))\|] (\mathbf{X} - \mathbf{W}_j(t)),$$

for all codebook-vectors j not belonging in $NE_i(t)$ at the t iteration of the algorithm, when an input vector \mathbf{X} is applied to the SOFM model. All codebook-vectors i belonging in $NE_i(t)$ are updated through the following rule

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + [a(t)] (\mathbf{X} - \mathbf{W}_{m_winners}(t)),$$

where $\mathbf{W}_{m_winners}(t)$ is a codebook vector calculated from the m -winning neurons at time t and not the single winning neuron. The calculation in our case involves weighted average of the m -winning vectors,

$\mathbf{W}_{m_winners}(t) = d_j \mathbf{W}_j(t)$, where d_j is the normalized distance $\|(\mathbf{W}_j(t) - \mathbf{W}_i(t))\|$, of winning neuron i from neuron j belonging to the m -winners at time t .

This update law could be compared to the conventional SOFM update,

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + [a(t)] (\mathbf{X} - \mathbf{W}_i(t))$$

The term $[b(t) \exp\|(\mathbf{X} - \mathbf{W}_j(t))\|]$ is similar to Kohonen's conventional learning rate $a(t)$ but now this parameter depends not only on time but, also, on distance due to the term $\exp\|(\mathbf{W}_{win}(t) - \mathbf{W}_j(t))\|$. That is, the larger the distance between winning codebook vector $\mathbf{W}_{win}(t)$ and codebook-vector j the larger the learning parameter so that the updated codebook-vector j increases dramatically its distance from the winning codebook vector as well as the input vector \mathbf{X} , in order to have faster convergence. This is the reason we call this new SOFM update scheme adaptive SOFM.

With regards to the second stage involved group of SVMs, the task of nonlinear regression/ controller compensation could be defined as follows. Let $f(\mathbf{X})$ be a multi-dimensional scalar valued function to be approximated. Then, a suitable regression model to be considered is: $D = f(\mathbf{X}) + n$, where \mathbf{X} is the input vector, n is a random variable representing the noise and d denoting a random variable representing the outcome of the regression process. Given, also, the training sample set $\{(X_i, D_i)\}$ ($i=1,..,N$) then, the SVM training can be formulated as next outlined:

Find the Lagrange Multipliers $\{\lambda_i\}$ ($i=1,..,N$) and $\{\lambda'_i\}$ ($i=1,..,N$) that maximize the objective function,

$$Q(\lambda_i, \lambda'_i) = \sum_{i=1..N} D_i (\lambda_i - \lambda'_i) - e \sum_{i=1..N} (\lambda_i + \lambda'_i) - \frac{1}{2} \sum_{i=1..N} \sum_{j=1..N} (\lambda_i - \lambda'_i)(\lambda_j - \lambda'_{j'}) K(\mathbf{X}_i, \mathbf{X}_j) \text{ subject to the constraints: } \sum_{i=1..N} (\lambda_i - \lambda'_i) = 0 \text{ and } 0 \leq \lambda_i \leq C, 0 \leq \lambda'_i \leq C \text{ for } i=1..N, \text{ where } C \text{ is a user defined constant.}$$

In the above definition, $K(X_i, X_j)$ are the kernel functions. In the problem at hand we have employed the radial basis kernel $K(X_i, X_j) = \exp(-1/(2\sigma^2) \|X_i - X_j\|^2)$. Taking into account all the previous definitions we can then, fully determine the approximating function as $F(\mathbf{X}) = \sum_{i=1..N} (\lambda_i - \lambda'_i) K(\mathbf{X}, \mathbf{X}_i)$

In addition, in order to accommodate the concept of positive/ negative response based specialized SVM training process previously illustrated, we should note that the corresponding training sample set $\{(X_i, D_i)\}$ ($i=1,..,N$) contains not only the X_i vectors coming from the associated SOFM defined cluster (sub-space) but, also, all other known input vectors X_j coming from the other subspaces, associated with the relevant SOFM codebook vectors. In order for such a concept to be implemented it is required

that our system memorizes all input vectors presented to the system prior to training iteration t, along with their corresponding winning codebook vectors. Therefore, it is then, easy to formulate the needed training set $\{(X_i, D_i)\}$ ($i=1, \dots, N$) at every iteration t.

Finally, regarding the third stage of the presented novel ANN controller, it employs a simple SVM model combining the outputs of the $m \times m$ nearest to the winning codebook vector second stage SVMs. Such a third stage SVM model attempts to compensate the estimation error of the second stage SVM associated to the relevant winning SOFM codebook vector. m is a user defined constant, and in our case has been selected to be equal to 3. To be more specific, this third stage SVM takes as inputs the $m \times m$ second stage SVMs, previously depicted, while its desired outputs are the same as the ones of the second stage SVM associated to the winning codebook vector. Thus, this third stage SVM implements function $SVM_{-3} = f(SVM_{-21}, SVM_{-22}, \dots, SVM_{-2k}, \dots, SVM_{-2m})$, where, SVM_{-2k} is the k_{th} nearest to the winning codebook vector second stage SVM above defined model.

This paper investigates the generalization accuracy of the suggested three stage system in estimating a robotic manipulator's inverse dynamics model in combination with a servo PD feedback controller as in fig. 1. Assuming that structural uncertainty occurs in our n-link robot manipulator, the correct model of its inverse dynamics is given by the differential equation in vectorial form [3],

$$\mathbf{T} = \mathbf{R}^{-1}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \quad (1)$$

where, \mathbf{T} is the joint torque, \mathbf{R}^{-1} is a nonlinear mapping from the joint coordinate space to the joint torque space and \mathbf{q} , $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ are the robot arm motion parameters: joint trajectory, velocity and acceleration variables. The correct model of its direct dynamics is [3],

$$\ddot{\mathbf{q}} = \mathbf{R}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{T}) \quad (2)$$

Robot dynamics, however, cannot be modelled exactly. An estimated model $\hat{\mathbf{R}}^{-1}$ is used to predict the feedforward torques and a servo-feedback control scheme is involved to improve robustness. To this end, the approach adopted here involves a PD servo-controller to compensate for the linear changes and the three-stage ANN based controller to compensate for the intrinsic nonlinearities encountered due to parameters uncertainties. Therefore, the supposed correct inverse dynamics model of the robot arm of figure 1 is defined as follows,

$$\mathbf{T} = \hat{\mathbf{R}}^{-1}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + \mathbf{T}_{pd} \quad (3)$$

$$\mathbf{T}_{pd} = \mathbf{K}_p (\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_v (\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \quad (4)$$

where, \mathbf{T}_{pd} is the joint torque estimated by the PD-controller, \mathbf{T} is the total joint torque, \mathbf{q}_d and $\dot{\mathbf{q}}_d$ are the desired trajectory and velocity curves and finally, \mathbf{K}_p , \mathbf{K}_v are the gains of the PD controller. From equations (3-4) it is clear that the proposed ANN model should be trained to predict joint torque $\mathbf{N} = \mathbf{T} - \mathbf{T}_{pd}$ provided, the curves \mathbf{q} , $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}_d$ of actual trajectory, velocity and desired acceleration are given as inputs. It is a common practice to use $\ddot{\mathbf{q}}_d$ instead of differentiating the velocity $\dot{\mathbf{q}}$ to get $\ddot{\mathbf{q}}$. In a

simulated version of the system, it is clear that equispaced samples of these three curves should be given as inputs to the ANN model. The algorithm of the simulation of the above control scheme can be depicted as follows: At every instance t the desired curves \mathbf{q}_d , $\dot{\mathbf{q}}_d$, and $\ddot{\mathbf{q}}_d$ as well as the curves of the actual trajectory and velocity ($\mathbf{q}(t)$ and $\dot{\mathbf{q}}(t)$) are given. Applying them as inputs to equation (1) when the true masses are involved we have the desired joint torque $T = T(\text{true})$. Then, if the false link masses are used in equation (4) a $\hat{T} = T_{\text{pd}}(\text{false})$ joint torque is computed. Subsequently, this false torque is applied to equation (4) involving the true parameters in order to derive the instance $t+1$ curves of the actual trajectory and velocity ($\mathbf{q}(t+1)$ and $\dot{\mathbf{q}}(t+1)$) of the robot arm. And so on. Therefore, we can take equispaced samples of the curves $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ as well as $\ddot{\mathbf{q}}_d(t)$ to form the suggested ANN's input patterns. In addition, we can take $N = T (= T(\text{true})) - \hat{T} (= T_{\text{pd}}(\text{false}))$ to form its desired output. By varying link masses within given intervals, we can obtain a large set of training, validation and test set patterns.

3 Experimental Study and Discussion of the Results

A simple two-link planar elbow arm was used to test the performance of the proposed three-stage methodology comparing the application of SVMs in its second and third stage either without involving the concept of positive/ negative response training or by involving it in order to achieve specialized sub-space training, as well as the merits offered by involving a hierarchical architecture as the one herein presented. The manipulator was modeled as a two rigid links of lengths $l_1 = 1\text{m}$ and $l_2 = 1\text{m}$ with point masses $m_1 = 0.8\text{kg}$ and $m_2 = 2.3\text{kg}$ at the distal end of the links corresponding to the false PD model, while the true masses varied within the 10% confidence interval of these values. Twelve variations have been considered for each such link mass. The simulation was carried out invoking a fourth order Runge-Kutta algorithm, with step size $h = 0.01$. The desired position trajectory has the components

$$\theta_1 = g_1 \sin(2\pi t/T) \quad \text{and} \quad \theta_2 = g_2 \sin(2\pi t/T) \quad (5)$$

with period $T = 2\text{s}$ and amplitudes $g_i = 0.1 \text{ rad}$. For good tracking the time constant of the closed-loop system was selected as 0.1s . For critical damping this means that for the PD Outer-Loop design $\mathbf{K}_v = \text{diag}\{k_v\}$, $\mathbf{K}_p = \text{diag}\{k_p\}$ with $k_p = 100$ and $k_v = 20$.

Application of equations (1), (3), (4) employing these definitions leads, for every one of the 144 total pair masses variations, to obtaining the curves of desired acceleration, actual trajectory and actual velocity along with the associated torque N to be modelled by the proposed three-stage ANN. Each such curve has been sampled into 30 points, from which, using sliding windows of length lw , we have formed training and test patterns for the ANN as follows. Each ANN input pattern contains lw points for the desired acceleration, the corresponding lw points for the actual trajectory and finally, the lw corresponding points for the actual velocity. The desired output value is the associated torque N of the lw -th sample of the sliding window under consideration. In our simulations $lw = 6$ and we have used 2000 patterns for ANN training and

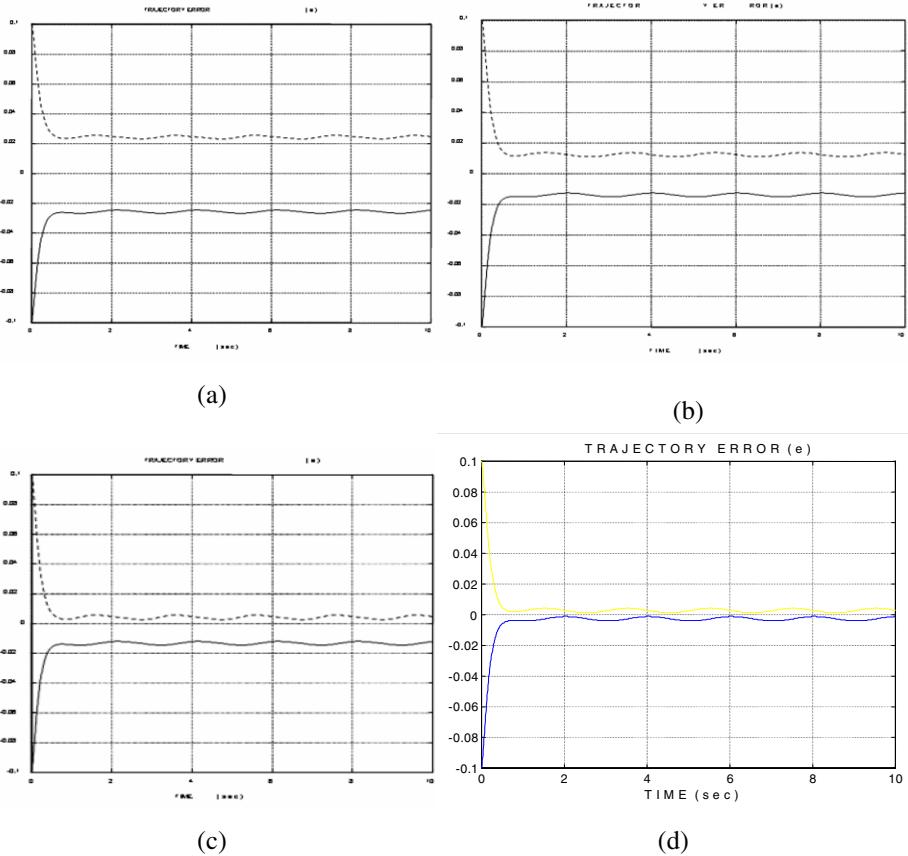


Fig. 2. (Regarding the above curves the dashed line represents first joint trajectory errors while the continuous line represents the second joint trajectory errors) (a) Trajectory tracking errors for the two joints when the true values of masses are $\pm 10\%$ of the false masses, when involving a PD only controller. (b) The same experiment using the correcting torque (N) obtained after using a simple SVM ANN based controller. (c) The same experiment using the correcting torque (N) obtained after using a similar to the proposed three stage methodology, but without involving the concept of positive/ negative response learning for the second stage SVMs specialization. (d) The same experiment using the correcting torque (N) obtained after using the proposed three stage hierarchical methodology, i.e involving SVMs in the second and third stage and employing the concept of positive/ negative response training for second stage SVMs specialization.

the rest 1600 patterns for testing. Finally, the resulting network was combined with the PD robotic controller to test the performance of the method. Figure 2(a) shows the trajectory tracking error when the true masses are well apart from the false masses, without additional correcting torque. Only the PD controller is involved. In figure 2(b) the same experiment is repeated but this time an additional correcting torque, produced by the ANN involving a single SVM, has been conducted. Its performance is better than the

PD controller respective performance. In figure 2(c)/2(d) the same experiment is repeated but this time an additional correcting torque, produced by the proposed three-stage ANN scheme either not involving (2(c)) the concept of positive/ negative response training for second stage SVMs specialization or involving it (2(d)), is applied to improve the performance of the controller. It is shown that the SVM based three-stage controller, with training specialization, outperforms all other rival controllers. This experiment is trajectory dependent. Further study is needed to make the method trajectory independent.

References

1. Haykin, S.: Neural Networks: A comprehensive foundation, 2nd edn. Prentice-Hall, Englewood Cliffs (1999)
2. Kawato, M., et al.: Hierarchical neural network model for voluntary movement with applicationn to robotics. IEEE Contr. Syst. Mag., 8–16 (1988)
3. Zalzala, M.S., Morris, A.S.: A neural network approach to adaptive robot control. Int. J. Neural Networks 2, 17–35 (1989)
4. Zomaya, Y.H., Suddaby, M.E., Morris, A.S.: Direct neuro - adaptive control of robot manipulators. In: Proc. IEEE Int. Conf. On Robotics and Automation, Nice, France, May 1992, pp. 1902–1907 (1992)
5. Khemaissia, S., Morris, A.S.: Neuro-adaptive control of robotic manipulators. Robotica 11, 465–473 (1988)
6. Wang, Z., Zhang, J., Chen, Z., He, Y.: Neural network_based on adaptive discrete-time global sliding mode control scheme. In: Intelligent Control and Automation. Lecture Notes in Control and Information Science, vol. 344, Springer, New York (2006)
7. Stepanenko, Y., Gao, Y., Su, C.Y.: Variable structure control of robots with PID sliding surfaces. Int. J. Robust Nonlinear and Control 8, 79–90 (1998)
8. Parra-Vega, V., Arimoto, S., Liu, Y.H., Hirzinger, G., Akella, P.: Dynamic sliding PID control for tracking of robot manipulators: theory and experiments. IEEE Trans. Robotic Automa. 19, 967–976 (2003)

Variable Colour Depth Look-Up Table Based on Fuzzy Colour Processing

Heesang Shin and Napoleon H. Reyes

Institute of Information and Mathematical Sciences,
Massey University, Auckland, New Zealand
shinheesang@gmail.com, n.h.reyes@massey.ac.nz

Abstract. This paper presents an application of a Fuzzy Colour Contrast Fusion (FFCF) algorithm in compensating for reduced colour depth representation of a colour image while maintaining efficient colour sensitivity that suffices for accurate real-time colour-based object recognition. We investigate the effects of applying fuzzy colour contrast rules to varying colour depth as we extract the optimal rule combination. The experiments were performed using the robot soccer game set-up with spatially varying illumination intensities on the scene. Interestingly, our results show that for most cases, colour depth reduction could actually improve colour classification via a pie-slice technique, in a modified rg-chromaticity colour space. For 6 different colours, the algorithm was able to yield 6.5% higher overall accuracy with only one-twelfth of LUT size than the full colour depth LUT.

1 Introduction

The compendium of algorithms presented in this paper work in cohesion, with the main purpose of more accurately classifying colour pixels depicting moving target objects under varying illumination intensities, while cutting-down memory space by finding the most efficient colour depth for each target colour. The approach described here is suitable for vision systems that employ region segmentation by colour. Such systems are deemed crucial in real-time mobile robot applications [1] like Robocup [2] and FIRA [3].

Altogether, the beauty of the techniques is that all possible colour combinations can be pre-processed, classified and stored into a variable colour-depth look-up table for real-time object tracking use. The system is calibrated by using exemplars of the colour objects in different illumination settings in an indoor environment. These exemplars are selected based on human visual judgment of which region is relatively dim, dark and bright. In turn, the gradient between these illumination conditions are automatically worked out by the algorithms. The algorithms work together to compensate for the effects of light intensity at every pixel of an object. That is, light intensity at each point in the image that is a product of the reflectance at the corresponding object point and the illumination intensity at the same location, apart from a constant factor contributed by the optical arrangement of the imaging device [4]. A fuzzy colour

contrast fusion algorithm (FCCF) [5] is employed to perform colour correction via contrast enhance or degrade operations [5] on the RGB colour channels independently. This is handled by a synthesis of optimal colour contrast rules which are extracted automatically by ranking all 343 (7 operations per channel) possible rule combinations according to an algorithm presented in [6]. Previously, FCCF in full colour depth has been successfully applied to real-time colour-object recognition [5] [7], and also been tested previously to work with the YUV, HSI and rg-chromaticity spaces with improved colour classification results [7]. In this work we explore the efficacy of FCCF when the colour depth is reduced to save memory space for colour classification, allow for portability to vision systems with limited resources and parallel and distributed processing adaptability. We analyse the clustering of the combination of fuzzy colour contrast rules at differing amount of reductions in colour depth, in search for the optimal ones. The ultimate goal is to find the minimum colour depth for each of the colour channels, while maintaining at least the same level of accuracy as the full colour depth LUT. We performed the experiments on the same illumination set-up and target colours used in [6] for direct comparison, and the colour classification results attained by the variable colour depth LUT show superior performance.

2 The Algorithms

2.1 Colour Descriptors from rg-Chromaticity Colour Space

The colour descriptors are extracted from a modified rg-chromaticity colour space. The original rg-chromaticity space is already known to remove ambiguities due to illumination or surface pose [8]. On top of that the colour descriptors we are using also lend themselves amenable for pie-slice colour classification [9].

$$\begin{aligned} \text{rg-chromaticities: } r &= \frac{R}{R+G+B}, g = \frac{G}{R+G+B} \\ \text{rg-Saturation} &= \sqrt{(r - 0.333)^2 + (g - 0.333)^2} \\ \text{rg-Hue} &= \tan^{-1}\left(\frac{g-0.333}{r-0.333}\right) \end{aligned}$$

2.2 Fuzzy Colour Contrast Fusion (FCCF)

Chromaticity distortions are compensated for via a fusion of colour contrast operations employed through fuzzy rules. The inputs are the original colour tristimulus in RGB form, as well as the calculated rg-Hue and rg-Saturation values. With reference to the set of optimal colour contrast rules detailing the contrast operations, levels of operations (i.e. 1x, 2x, 3x), as well as the colour contrast constraint angles, the appropriate rules are applied independently on the colour channels. Consequently, FCCF produces the colour-corrected RGB values for more accurate colour classification.

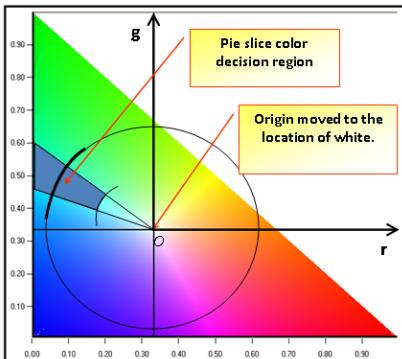


Fig. 1. Modified rg-chromaticity space and the pie-slice decision region

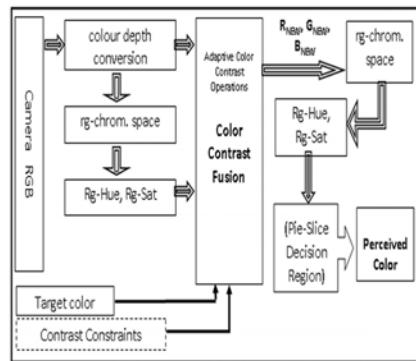


Fig. 2. FCCF with Reduced Colour Depth

Example 1. Sample colour contrast rule

If(pixel depicts LIGHTBLUE according to colour contrast constraint angles) Then apply colour contrast enhance on the Red channel 2 times, apply colour contrast degrade on the Green channel and apply colour contrast enhance on the Blue channel 3 times.

The fuzzy colour contrast rules employ contrast enhance and contrast degrade operations on the colour channels independently through Equations (1) and (2).

Contrast Enhance Operator:

$$\alpha = \begin{cases} 2\mu_\alpha^2(y) & 0 \leq \mu_\alpha(y) < 0.5 \\ 1 - 2[1 - \mu_\alpha(y)]^2 & 0.5 \leq \mu_\alpha(y) \leq 1 \end{cases} \quad (1)$$

Contrast Degrade Operator:

$$\alpha = \begin{cases} 0.5 + 2[\mu_\alpha(y) - 0.5]^2 & 0 \leq \mu_\alpha(y) < 0.5 \\ 0.5 - 2(1 - [\mu_\alpha(y) + 0.5]^2) & 0.5 \leq \mu_\alpha(y) \leq 1 \end{cases} \quad (2)$$

For each colour channel, there are 3 possible manipulations: enhance, degrade or no operation. Moreover, enhance and degrade operations could also be applied multiple times; that is, 1, 2, or 3 times. Therefore, for each colour channel, there are 7 possible combinations of operations. Altogether, for three colour channels, there are 343 possible rules. Note that the optimal rules for each target colour are extracted automatically through a scoring algorithm described in [6].

The following algorithms describe how to construct and search a variable colour depth Look-Up-Table.

Algorithm 1. Variable Colour Depth LUT build algorithm

```

foreach  $t \leftarrow$  every target  $n$  colours do
    for  $R \leftarrow 0$  to  $2^{ddrn} - 1$  do                                // Every possible Red values
        for  $G \leftarrow 0$  to  $2^{ddgn} - 1$  do                                // Every possible Green values
            for  $B \leftarrow 0$  to  $2^{ddbn} - 1$  do                                // Every possible Blue values
                if colour value is classified as colour  $t$  then
                     $L = (R \ll (ddgn+ddbn)) + (G \ll ddbn) + B;$ 
                     $LB = L \gg (b \log_2);$ 
                     $Lb = 1 \ll (L \bmod b);$ 
                     $LUT[t]/LB] = LUT[t]/LB] \cup Lb;$ 

```

dsr, dsg, dsb are the colour depth values of each colour channel in source image
 $ddrn, ddgn$ and $ddbn$ are the colour depth values of each colour channel in each target colour LUT

LB is an index to the LUT that corresponds to the target colour

b is the size of data type of LUT optimised for the system architecture (e.g. 8 for byte-aligned, and 16 for word-aligned)

2.3 Variable Colour Depth LUT Build and Search

After the optimum set of colour contrast rules, pie-slice decision region parameters and optimum colour depth for each channel has been determined, the variable colour depth LUT is generated. Algorithm 1 builds an LUT for each target colour t , scanning every possible colour values. If the colour value is classified as a target colour, a bit is set in the LUT at a calculated location to indicate membership to that target colour. Given the source colour value of a pixel, Algorithm 2 searches the LUTs of each possible target colours t to classify its colour. The corresponding LUT location for each target colour depth is calculated and a bit mask AND operation is used to extract the target query bit. Note that the LUT location calculation requires shift-left as well as shift-right operations in order to discard excessive bits in the source colour value.

There's a possibility that a single pixel could be classified as multiple colours, as returned by each LUT. As illustrated in Fig. 4, this could actually help resolve ambiguities when the object is under extreme illumination condition, the pixel may be classified as either pink, or blue or violet, but not yellow or green. With reference to the neighboring cluster of pixels, the pixel gets the same classification as the cluster. Moreover, the algorithms lend themselves amenable for parallel and distributed implementations since each LUT could be generated and searched independently of the others.

3 Experiments

The experiments were performed on the same robot soccer test bed used in [6] for comparison purposes. However, the calibration set up is non-typical, as it is plagued with spatially varying illumination intensities, with 6 target colours,

Algorithm 2. Variable Colour Depth LUT query algorithm

```

foreach  $t \leftarrow$  every target  $n$  colours do
     $R =$  Red component value of target pixel;
     $G =$  Green component value of target pixel;
     $B =$  Blue component value of target pixel;
     $L = ((R \gg (dsr - ddrn)) \ll (ddgn + ddbn)) + ((G \gg (dsg - ddgn)) \ll ddbn) + (B \gg (dsb - ddbn));$ 
     $LB = L \gg (b \log_2);$ 
     $Lb = 1 \ll (L \bmod b);$ 
    if (LUT[t][LB]  $\cap$  Lb)  $\neq 0$  then
        Given pixel is qualified for target colour  $t$ 

```

represented by 40 colour patches, strategically positioned to be exposed under different illumination conditions (i.e. dim, dark, bright). The focus of the experiments is to compare colour classification results when the full colour depth LUT (24 bits) is used vs. variable colour depth LUTs. The classification performance is gauged based on a scoring formula proposed in [6]. The formula takes into account the number of true positives, false positives, as well as the area of the target colour objects, and has proven to identify the superior colour contrast rule combination.

3.1 Colour Classification Results of Full 24-Bit Colour Depth vs. Variable Colour Depth

We employed the same colour classification definition for the 6 target object colours tested in [5] for direct comparison of algorithm performances. The previous research used a 24-bit colour depth LUT for each target colour, and utilised an algorithm for automatic extraction of the angles and radii values, and colour contrast rules. Table II shows comparisons between the best scores from the previous research and this research.

As observed from the table, it is clear that the application of the variable colour depth approach resulted to better scores than the full 24-bit colour depth LUT in all 6 target object colours. It is evident that the misclassifications have been significantly reduced down to at least 50% for all target colours.

3.2 Optimised Colour Contrast Rules

The clustering of the best set of colour contrast rules for each possible colour depth value, and for each target colour is shown in Fig. 3. The colour depth values were varied from 5 to 8, considering a total of 64 possible permutations inspected for each target colour. The actual numeric figures are presented in Table II.

On the other hand, Fig. 5 shows a mapping of the best colour contrast rule combinations for the optimal colour depth values for each target colour. It also reflects the number of occurrences of the same colour contrast rule combination

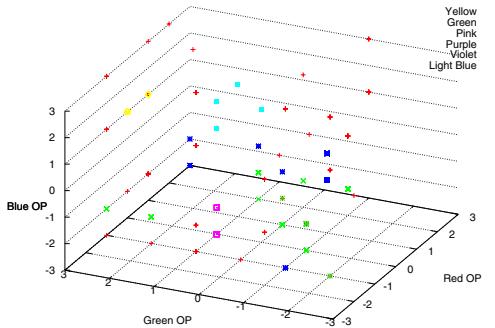


Fig. 3. Mapping of all the Best Colour Contrast Rule Combinations for all Colour Depth Values and for each Target Colour

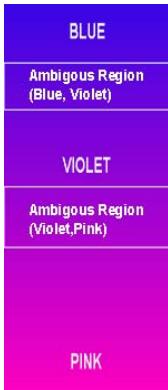


Fig. 4. Colour Gradiation From Blue to Pink

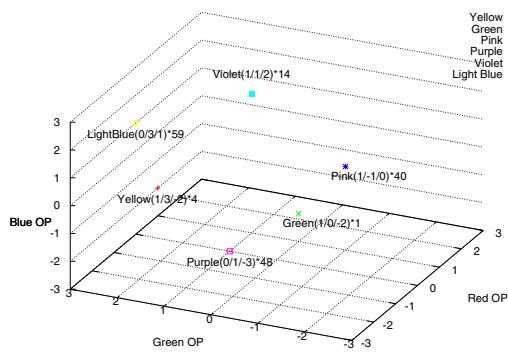


Fig. 5. Mapping of the Best Colour Contrast Rule Combinations for the Optimal Colour Depths for each Target Colour. Positive number indicates contrast enhancement and level of contrast application; 0 for No Operation, while a negative number denotes contrast degradation.

for each target colour. As indicated in the figures, the optimal colour contrast rule combinations for LightBlue, Pink and Purple adhere with the majority of colour contrast rule combinations found at different colour depth values. There are some observable patterns in the rule combinations as well. For instance, Green always requires the color contrast degrade operation on the Blue channel. In contrast, there is no observable pattern on Yellow's contrast rule combinations; they are scattered all over the rule space. The scattering of the rules is due to the fact that the decision region for Yellow is defined with very narrow bounding angles. Consequently, these only proves that FCCF effectively employs colour contrast rules to compensate for the escaping pixels from the narrow pie-slice decision region for Yellow.

3.3 Colour Contrast Rules and Scores

Table II details the results of the experiments on optimised colour contrast rule extraction at varying colour depth. The table reflects the scores garnered by the rule combinations in classifying the 6 target colours (represented by 40 colour patches). The table indicates the colour depth and colour contrast operation used for each of the colour channel, the performance of the rule combination in terms of the number of hits, misclassifications, storage space requirement, improvement over the full-colour-depth LUT and rule combination's relative ranking. The best results show that FCCF increased colour classification even when there were lost bits in the colour depth.

Table 1. Colour Contrast Rules and Scores

Colour	Depth R , G ,B	Contrast Rule R ,G ,B	Score	Hits	Misses	LUT Size	Improvement Rate	Rank
Yellow	8, 8, 8	3, 1, -2	0.648530	2104	68	2048KB	0%	3
	7, 8, 6	1, 3, -2	0.655979	2261	96	256KB	1.149%	1
	6, 8, 6	-1, 1, -3	0.609815	2258	172	128KB	-5.97%	4
	7, 8, 8	1, 3, -2	0.655374	2261	97	1024KB	1.055%	2
Green	8, 8, 8	0, -1, -3	0.552422	3313	383	2048KB	0%	4
	6, 5, 8	1, 0, -2	0.639059	3137	127	64KB	15.683%	1
	6, 8, 6	2, 1, -2	0.587168	3266	288	128KB	6.29%	3
	7, 8, 8	0, -1, -3	0.615805	3239	206	1024KB	11.474%	2
Pink	8, 8, 8	1, -1, 0	0.586446	1714	99	2048KB	0%	4
	7, 8, 7	1, -1, 0	0.622773	1679	46	512KB	6.194%	1
	6, 8, 6	1, -1, 0	0.603303	1623	58	128KB	2.874%	3
	7, 8, 8	1, -1, 0	0.616230	1684	55	1024KB	5.079%	2
Purple	8, 8, 8	0, 1, -3	0.572888	2777	314	2048KB	0%	2
	6, 7, 7	0, 1, -3	0.576178	2782	309	128KB	0.574%	1
	6, 8, 6	0, 1, -3	0.565163	2729	313	128KB	-1.348%	4
	7, 8, 8	0, 1, -3	0.572094	2773	314	1024KB	-0.139%	3
Violet	8, 8, 8	1, 1, 2	0.526654	2535	497	2048KB	0%	4
	5, 7, 5	0, 1, 1	0.602979	1802	101	16KB	14.492%	1
	6, 8, 6	1, 1, 2	0.545970	2502	442	128KB	3.668%	2
	7, 8, 8	0, 0, 2	0.529645	2400	425	1024KB	0.568%	3
Light Blue	8, 8, 8	0, 3, 1	0.668808	2758	68	2048KB	0%	4
	5, 5, 6	0, 3, 1	0.690887	2786	30	64KB	3.301%	1
	6, 8, 6	0, 3, 1	0.671966	2703	48	128KB	0.472%	2
	7, 8, 8	0, 3, 1	0.671255	2758	63	1024KB	0.366%	3

3.4 Reductions in Memory Usage

As depicted in Table II, the memory storage requirement for the reduced colour depth LUT optimized for colour classification varies between 16KB and 512KB. Altogether, for the classification of all 6 target colours, the total memory requirement for the variable colour depth LUT is only 984KB. In contrast, a 24-bit colour depth LUT requires 12MB for all 6 colours.

4 Conclusions and Future Work

The FCCF algorithm proves to be effective in compensating for the effects of reduction in colour depth by influencing the formation of colour clusters depicting colour objects within their pie-slice decision region in the colour space. This is

particularly useful for colour classification. The experiments' results show that different colour contrast rule combinations were derived for different colour depth configurations. The best colour classification scores all came from the utilisation of variable colour depth LUTs, and therefore with less precise colour values. These colour depth values varied between each colour channel. In general, it can be deduced that the lower colour-depth values eliminated selectively some colour information that paved the way for more effective colour discrimination. This resulted to better scores, either by achieving higher true positives in the colour classification process, or reducing false positives. For all 6 target colour objects, memory consumption is reduced from 12MB down to only 984KB, or 92% reduction from the full 24 bit colour space. While achieving reduced memory consumption, on the average, the colour classification scores for all 6 colours increased from 0.592625 up to 0.631309, or with 6.5% overall improvement. All colour classification scores improved for each target colour, as compared to using the full 24-bit colour space. For future works, we intend to fuse Genetic Algorithms with FCCF in automatically extracting the optimal colour descriptors, fuzzy rules and colour depth values for each colour channel to further improve the colour calibration process.

References

1. Bruce, J., Balch, T., Veloso, M.: Fast and inexpensive color image segmentation for interactive robots. In: Proceedings of IROS 2000, pp. 2061–2066 (2000)
2. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H.: Robocup: A challenging problem for AI. *AI Magazine* 18, 73–85 (1997)
3. Kim, J.H., Seow, K.T.: Soccer Robotics. Springer, Heidelberg (2004)
4. Horn, B.K.P.: Determining lightness from an image. *Computer Graphics and Image Processing* 3, 277–299 (1974)
5. Reyes, N.H., Dadios, P.E.: Dynamic color object recognition using fuzzy logic. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 8, 29–38
6. Playne, D.P., Mehta, V.D., Reyes, N.H., Barczak, A.L.C.: Hybrid fuzzy colour processing and learning. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) *ICONIP 2007, Part II. LNCS*, vol. 4985, pp. 386–395. Springer, Heidelberg (2008)
7. Reyes, N.H., Messom, C.: Identifying colour objects with fuzzy colour contrast fusion. In: 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, and FIRA RoboWorld Congress (2005)
8. Finlayson, G.D., Schiele, B., Crowley, J.L.: Using color for image indexing. In: Proceedings of Challenge of Image Retrieval (1998)
9. Thomas, P., Stonier, R., Wolfs, P.: Robustness of color detection for robot soccer. In: Proceedings of the Seventh International Conference on Control, Automation, Robotics and Vision, pp. 1245–1249 (2002)

Towards a Generalised Hybrid Path-Planning and Motion Control System with Auto-calibration for Animated Characters in 3D Environments

Antony P. Gerdelen and Napoleon H. Reyes

Institute of Information and Mathematical Sciences, Massey University,
North Shore 102-904, Auckland, New Zealand
`gerdelan@massey.ac.nz, n.h.reyes@massey.ac.nz`

Abstract. Intelligent navigation and path-finding for computer-animated characters in graphical 3D environments is a major design challenge facing programmers of simulations, games, and cinematic productions. Designing agents for computer-animated characters that are required to both move intelligently around obstacles in the environment, and do so in a psycho-visually realistic way with smooth motion is often a too-difficult challenge - designers generally sacrifice intelligent navigation for realistic movement or vice-versa. We present here a specially adapted hybrid fuzzy A* algorithm as a viable solution to meet both of these challenges simultaneously. We discuss the application of this algorithm to animated characters and outline our proposed architecture for automatic tuning of this system.

1 Introduction

Our previous works have focused on the development of next generation navigation algorithms for soccer robots by combining existing navigation and motion control algorithms into hybrids which exploit the benefits of the individual algorithms but also negate their limitations. To this end we have developed the Hybrid Fuzzy-A* [1] algorithm which can perform both efficient and dynamic route planning in environments with static and moving obstacles, and also move using smooth-path motion and avoid obstacles at very high speeds, requiring only a very small allocation of CPU resources as robot soccer teams are typically controlled by one commodity desktop computer which also has to process computer vision and direct 3-11 other robots every 1/30th of a second. Several variants of this hybrid were created which enabled us to incorporate other properties into robot behaviour [2] [3] - in particular, an ability for the ball-carrying robot to *evade* hostile robots on its path to the opposition goal and pre-empt their path of movement as well [4]. The success of these works, and also the impressive appearance of this algorithm operating in 3D graphical simulation (see Figure [1]), lead us to speculate whether the algorithm could be applied to a



Fig. 1. Our 3D robot soccer simulator with robots controlled by the Fuzzy A* algorithm. The darker coloured robot at the top of the figure plans a path through moving robots. Blobs indicate the points along the *current* plan for the robot's path.



Fig. 2. An animated character - in this case a to-scale model of a Soviet T-28 circa 1932 with simulated real physical operating characteristics - travels through a 3D-environment in our simulation engine complete with hills and a range of obstacles (trees and houses)

range of much more demanding applications beyond the mathematically simple 2d realm of soccer robots.

We have constructed a real-time simulation engine using the OGRE3D graphics library [5], which has been used successfully for military visualisation and simulation projects [6] with success, to mimic a typical computer game-type environment as a proving ground for the application of hybrid robot soccer control algorithms to much more complex 3D environments. This type of environment also typically has limited CPU available for artificial intelligence computation as 3D graphics processing tends to occupy the bulk of resources [7] so the need for efficient algorithms is paramount, and can process sensory information with large amounts of environment noise or complexity [8] - the kind of environments where Fuzzy Logic is at a premium.

The Hybrid Fuzzy A* algorithm has been adapted so that it is scalable and can be applied to a range of animated characters with their own specific motion characteristics. Our initial implementation has been successful with vehicle-type animated characters - see Figure 2. As depicted in the figure, the vehicle has planned a prospective path around obstacles using the A* component of the Hybrid Fuzzy A* algorithm (as indicated by the segmented line visible ahead of the vehicle in the figure) and has detected, and is driving around the *nearest obstacle* on its right hand side (as indicated by the second line visible in the Figure) - in this case a house.

Due to the complexity of manually tuning the many algorithm parameters for optimal results we have also designed an architecture for automatic parameter calibration so that the system can self-train for application to a range of new characters and environments by operating on a series of *obstacle courses* - typical operating environments created by a designer for training - and self-evaluating.

We also detail here some new visualisation techniques, which we have found necessary to develop, for visibility of complex multi-level artificial intelligence systems in real-time applications; as post-run analysis of animated character motion is an almost impossible task.

2 Hybrid Algorithm

The Hybrid Fuzzy A* algorithm is arranged in a cascade of systems. At the top level, we give the character some information about the obstacles environment and it builds an *environment obstacle map*. The environments that we are considering so far can be simplified and expressed as a 2-dimensional map and are stored as a 2D array of values. More complex path-planning might require a 3D array of values. We are using a character array to store our obstacle map, as this allows us to express a range of obstacle types with single letters ('s' for shrubs, 'b' for buildings). A requirement of the system is that the character has some ability to divide the environment into a graph or searchable area. The character then translates known obstacle positions into graph indexes and marks the locations. In an environment with many dynamic (moving) obstacles we recompute this map with every frame of calculation. Larger obstacles can occupy multiple graph cells. The character will ultimately make use of several types of *environment map* as weighted search domains for a *depth-limited dynamic A* algorithm* (with depth limited to scale to available CPU window per application) to compute a near-optimal, partially complete path to its target destination. Graph nodes marked as containing obstacles will be either excluded from the search domain (impassable obstacles) or given a weight based on their resistance to the character's movement (for example shrubs may be given a moderate weight modifier for heavy vehicles in a military simulation or game, as they can be driven over, but buildings may be impassable and excluded from the search domain). This path is broken into *waypoints* and the first or an early-stage waypoint (if the graph resolution is very high) is used as the *current waypoint* and given to the Fuzzy navigation system as *target location crisp input*. Our experience with robot soccer path-planning [4] has shown us that the A* algorithm's heuristic can be modified to incorporate many different weighted inputs into the path-planning decision process.

$$f^* = h^* + u^* + g^* \quad \text{scoring function with undesirability feature } (u^*) \quad (1)$$

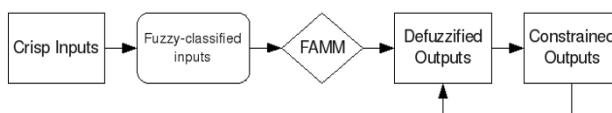


Fig. 3. Scalable Fuzzy Navigation cascade with feedback loop to adjust instructions for a character's known physical limitations

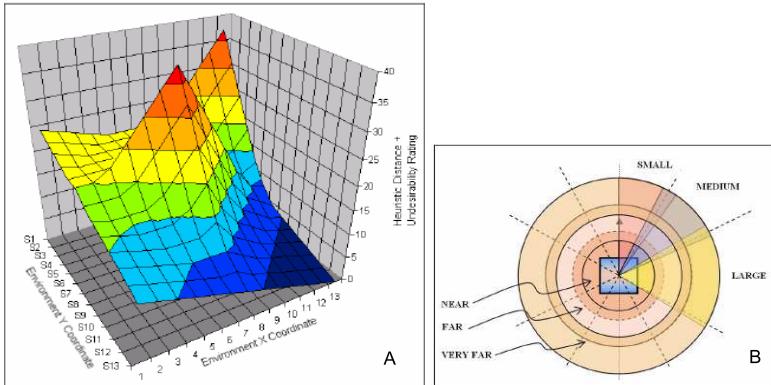


Fig. 4. (A) Representing combined heuristic weights of cells in the environment graph. The A* algorithm will choose the most overall down-sloping path (minimising overall cost + balanced heuristic). (B) Classifying environment elements into overlapping fuzzy sets representing angles and distances.

Eq. (II) depicts the modified scoring function. The ranges of u^* (undesirability) must be carefully balanced with h^* values (heuristic distance from starting node to goal node). On the other hand, the g^* component (actual cumulative distance) ensures an overall steepest downhill (shortest) path. This generates the *heat environment map* (Figure 4, A). It influences the selection of paths that are slightly longer but safer; inculcating an evasive path planning property. Factors such as slope of terrain, and known condition of ground surface can be weighted as proportions of the heuristic. Prudent designer decisions are required here to choose heuristics that are meaningful in a particular 3D environment, although the best ratio of weights of each input could be determined by self-training simulation runs. These heuristics may be tailored to a particular vehicle or re-evaluated periodically to adapt to changing conditions; reflecting the value of a true dynamic A*, as opposed to a less flexible variant such as D* [9]. The balancing of heuristic weights is deemed delicate.

The Fuzzy navigation layer is a rapid calculation system that takes the *current waypoint* on the path created by the A* layer as a rough guide (Figure 6). The systems can disagree - for example if the A*-generated path dictates to move left around a tree, but the vehicle is actually already moving around it to the right - then the path will be recalculated in the next frame of calculation. The fuzzy system consider two groups of crisp input - the *distance and angle to the nearest obstacle* and the *distance and angle to the current waypoint* see Figure 4 (B) for an illustration of this classification. These two groups of inputs are fed into distinct fuzzy set membership functions (only 3 sets per input) and fuzzy associative memory matrices. We are taking advantage of angle symmetry, and therefore, the rules are generalised - the antecedents do not explicitly state whether the obstacle/target is on the left or right [4]. In effect, using 3 fuzzy sets (9 rules) for half of the angle space (right-half) (Fig. 4, B) gives the same accuracy as using 6 fuzzy sets. The fuzzified outputs for obstacle avoidance and

target seeking behaviour are then be blended together using a centre of gravity function to produce smooth transitions between the two behaviours. Take note that this reactionary layers are controlling the steering angle and speed for both target pursuit and obstacle avoidance(Figure 6). On the other hand, the evasive behaviour is inculcated by the modified A* scoring function Eq.(11).

Navigation outputs are, in our adapted hybrid algorithm, expressed as *ratios* of character properties. Steering is also expressed as ratio of the speed control outputs. In this way the algorithm should produce motion that is both scalable to a character's motion limitations and applicable to a range of characters. We have also extended the Fuzzy A* algorithm with a feedback loop (see Figure 3). This allows us quickly re-scale our defuzzified output ratios should the intended output ratios (for example a vehicle can not steer as sharply as desired at its current speed) exceed a character's physical limitations.

3 Visualisation

Figure 5 is a screenshot from the 3D graphical simulation that we have created as a proving ground for our algorithms. Obstacles of various sizes are strewn about the 3D landscape (houses and trees) and several monitoring panels overlay the display. Because of the level of complexity and number of overlapping systems involved in real-time hybrid algorithms, it is a great aid to have a system in place for visualising these different processes in real-time so that the designer can monitor the system as it happens with a full range of information. When an interesting case does seem to occur in the behaviour of the character - for instance, we observed that our character was speeding up and slowing down even when travelling in a straight line - as illustrated by the sawtooth pattern in the 3rd graph down on the left in Figure 5. When this behaviour was observable we were able to pause the simulation time so that the character stopped moving mid-sequence. We observed the saw-tooth velocity graph and have a system in place whereby we were able to export the dataplots for all of these graphs to a range of formats for closer inspection in external programmes, where we could see that the *current waypoint* being assigned to the Fuzzy control system was always too close to the vehicle due to the high resolution of the path-finding system.

4 Proposed Self-training Architecture

Given the large number of parameters that need to be calibrated for animated character control, and in particular for hybrid algorithms, it has been proposed that various automatic training systems be employed for this task; genetic algorithms have been explored for robots soccer agent training [10], and evolving neural networks have been used to improve and even generate entire behaviours for animated characters [11] [12] with some success. We intend to adopt a subset of these ideas to self-train our characters. We propose an obstacle-course based *survival of the fittest* paradigm.

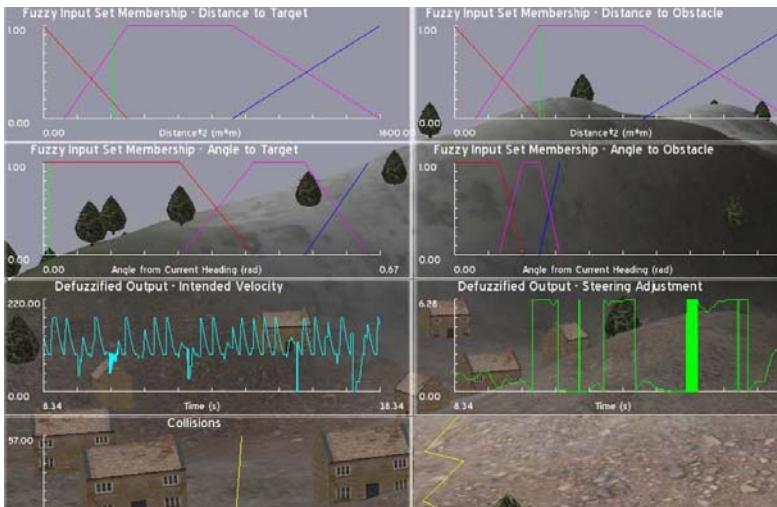


Fig. 5. Graphing navigation data in real time

A range of obstacle courses can be quickly created with scenario designers, such as the real-time designer that we have built into our simulation engine. These courses can be quickly designed as test cases for the sort of 3D environments that the characters would be expected to operate when finally deployed; for example we might design a course or scenario consisting of flat landscape and large streets of house-like obstacles, and another course consisting of hills and valleys with sporadic trees to avoid for training an animated soldier character for a war scene in a film production.

The advantages of this kind of automatic training system would be:

- Initial characters start from some adequate basis, rather than from a zero-skill base
- Can train a large range of simulated characters or even simulations for real vehicles.
- Can adapt itself to new environments - e.g. we can hand-craft a new scenario that the character should have to be able to cope with and it can improve its system to deal with this new environment as well.
- Could be set to self-improve in real-time during real execution over a time-slice basis, rather than a per-individual character basis.

If we arrange a batch of simulation runs where we randomly adjust the parameters of a character by some threshold, then start the character at a range of pseudo-random start locations and orientations, and ask for them to move to some fixed destination over a number of runs, then we can also ask the character to evaluate its performance based on a fitness function that generates a fitness score based on *completion time* and a metric determining number and severity of *collisions* (based on seconds of simulation time * distance inside obstacle bounds). You can see in Figure 5 that we are already extracting these parameters

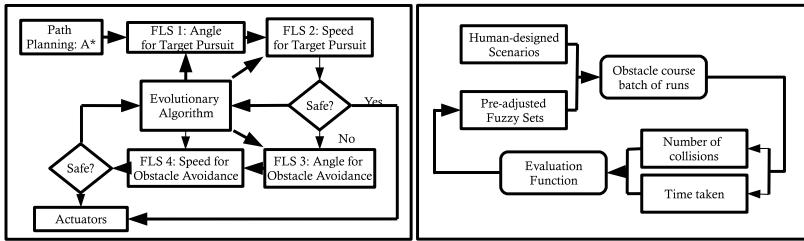


Fig. 6. The Hybrid Fuzzy A* - Evolutionary Algorithm (left) and the proposed algorithm for self-improving animated character navigation (right)

per character evaluation. Using a *generation* of our character, each using slightly different modifications of parameters, we can *select* the best two members of the generation (with the most minimal fitness scores) to create the base parameters for the next generation. This process is illustrated in Fig 6. We could also distribute this system over a network of commodity-level machines for very rapid mass training. We have infrastructure in our simulation engine in place for this sort of distributed computing and intend to carry out a full investigation of this training architecture in works in the near future.

5 Conclusions

We have presented a snapshot of our progress adapting hybrid robot navigation algorithms for animated characters in 3D environments, introducing adaptations that we have had to make to the system for application to more complex 3D environments, our work towards completing a generalised version of this system that can be applied to a range of autonomous animated characters, and illustrated some visualisation techniques that we have found particularly relevant to developing these sorts of real-time systems. We seek to expand on this work by automating the calibration procedures for the hybrid algorithms dictating path-planning and motion control for these characters, and have in this paper introduced our proposed architecture for this system.

References

1. Gerdelen, A.P., Reyes, N.H.: A novel hybrid fuzzy a* robot navigation system for target pursuit and obstacle avoidance. In: Proceedings of the First Korean-New Zealand Joint Workshop on Advance of Computational Intelligence Methods and Applications, Auckland, New Zealand, vol. 1, pp. 75–79 (2006)
2. Gerdelen, A.P.: Artificial Intelligence in Robot Soccer. Bachelor of engineering honours year project report, Institute of Information and Mathematical Sciences, Massey University, Albany, New Zealand (October 2006)
3. Gerdelen, A.P., Iskandar, D., Djohar, A.F., Reyes, N.H.: Utilising the hybrid fuzzy a* algorithm in a cooperative multi-agent system. In: Conference Program and Abstracts of the 4th Conference on Neuro-Computing and Evolving Intelligence (NCEI 2006) and 6th International Conference on Hybrid Intelligent Systems, HIS 2006 (2006)

4. Gerdelen, A.P., Reyes, N.H.: Synthesizing Adaptive Navigational Robot Behaviours using a Hybrid Fuzzy A* Approach. In: Advances in Soft Computing: Computational Intelligence: Theory and Applications, pp. 699–710. Springer, Heidelberg (2006)
5. Junker, G.: Pro OGRE 3D Programming. APress (2006); ISBN 1590597109
6. Lawes, G., Barlow, M.: Visual realism and decision making: A novel approach to real-time maritime battlespace visualisation. In: SimTecT 2007 Conference Proceedings. Virtual Environment and Simulation Laboratory (VESL), University of New South Wales at the Australian Defence Force Academy, Simulation Industry Association of Australia (2007); ISBN:0 9775257 2 4
7. Barron, T.: Strategy Game Programming With DirectX 9.0. Wordware (2003); ISBN 1-55622-922-4
8. Funge, J.D.: Artificial Intelligence for Computer Games. A K Peters, Ltd., Wellesley (2004)
9. Choset, H.M., Hutchinson, S., Lynch, K.M., Kantor, G., Burgard, W., Kavraki, L.E., Thrun, S.: Principles of Robot Motion: Theory, Algorithms, and Implementation. MIT Press, Cambridge (2005)
10. Messom, C.H.: Genetic algorithms for autotuning mobile robot motion control. Res. Lett. Inf. Math. Sci. (3), 129–134 (2002); ISSN 1175-2777
11. Stanley, K.O., Bryant, B.D., Miikkulainen, R.: Real-time neuroevolution in the nero video game. IEEE Transactions on Evolutionary Computation 9(6), 653–668 (2005)
12. Miikkulainen, R.: Creating intelligent agents in games. The Bridge 36(4), 5–13 (2006)

Cultivated Microorganisms Control a Real Robot: A Model of Dynamical Coupling between Internal Growth and Robot Movement

Hiroaki Wagatsuma

Laboratory for Dynamics of Emergent Intelligence, RIKEN BSI,
2-1 Hirosawa, Wako-shi, Saitama
waga@brain.riken.jp

Abstract. In biological systems, internal microorganisms adaptively survive but often serve necessary functions to the host, such as energy production by mitochondria as symbiosis. On the other hand, malignant germs have a parasitic relationship with the host and provide no benefit. A significant property that distinguishes healthy symbioses and malignant parasites is reproduction speed, or pace. For example, the rapid reproduction of influenza viruses destructs the host system, resulting in death. This study explored the necessary temporal property to establish a healthy relationship with the host under conditions where internal organisms have individual life spans. We propose a simple model of microorganisms, which are distributed spatially as colonial organizations undergoing temporal evolution and hypothesize that a self-consistent rhythm generated in collective behavior that is functionally coupled with the temporal global property of the host system is critical. To investigate the real-time coordination capability, an experimental framework with a mobile robot moving in the real world was used. As the on-line system, the microorganism model controls this robot. In this model, microorganisms expanded spatially and had colonial and power law distributions through time evolution. The neighboring distances, which are crucial for reproduction speed and are globally modulated by the size of the whole living area, are plastically changed to exhibit a rhythmic modulation. In the real-environmental experiment, the robot's navigation was successfully demonstrated by producing a temporal adaptability of microorganisms with the living area reshaped according to the current sensory information of the mobile robot. This is a first step of the microorganism-based framework to investigate the real-time coordination mechanism between internal and external timescales. The result may further groundbreaking research of bio-morphological robots.

1 Introduction

Biological organisms spend time and energy in their own spatial scales. They live cooperatively in a nested structure. An organism folds smaller organisms inside. For instance, animals have brains consisting of nerve cells. These nerve cells encompass symbiotic microorganisms in the hierarchical ecological system. Such internal organisms provide a collective activity displaying flexibility and adaptability for their own

survival and that of the host system. We know that, for instance, synaptic growth of the neuronal attachments that frequently occur in our brain in several tens of milliseconds, retains our memory of daily experiences. Sequential processes in gene expression and protein synthesis on a micro scale are necessary for the development and sustenance of our physical bodies. These facts suggest that activities of the upper (host) and lower (internal) levels, even in different scales, are cooperatively coupled by means of temporal coordination between individual progression speeds. Unlike digital computers with an invariant system clock, biological systems flexibly vary their internal paces to make spatial and temporal structures, with respect to that of other systems even in different levels. More simply a cooperative relationship can be observed in the life of some bacteria, such as mitochondria [1]. These bacteria live in the cell and supply energy materials to the host cell. However, the parasitic relationship is not always beneficial to the host, *i.e.* malignant pathogens. A significant distinction of healthy symbioses and malignant parasites is reproduction pace. For example, the rapid reproduction of influenza viruses destructs the host system, resulting in death. In the case of mitochondria, the cell reproduction cycle and growth are well regulated, possibly in relation to the host cell activity.

This study explored the temporal and collective dynamics necessary to establish an organic and healthy relationship with the host even if the organisms have life spans and generational shift. To investigate temporal necessity, we propose a simple behavioral model of microorganisms distributed spatially as colonial organizations through the temporal evolution. We hypothesize that the generation of a self-consistent rhythm in the collective behavior, so that it is functionally coupled with the temporal global property of the host system, is critical. To construct an experimental framework to examine the real-time coordination capability, we used a mobile robot moving in the real world as the host system. As a dynamical coupling system, the robot is interactively controlled by the collective spatial property of microorganisms and the sensory information of the robot feeds back to the internal system immediately and it is transformed to the shape of the living area of the microorganisms. Since the growth of microorganisms is sensitive to the living size, we observed a temporal adaptation of microorganisms and a change of the collective behavior, which are expected to contribute to the robot's real-time control in the behavioral time scale.

Section 2 describes the dynamics of the microorganism model. Section 3 introduces the fundamental collective behavior of the model displaying a self-organized distribution expanded to the whole living space that examined the internal rhythm and sensitivity to external modulation. Section 4 exhibits a preliminary result of the real-time control of the mobile robot using the proposed model. Finally section 5 presents the conclusions of this work.

2 Model of Microorganisms

In this section, we define the dynamics of the model of microorganisms that we use in the present paper. For the current purpose, we propose a simple model of microorganisms with the following behavioral properties: 1) separation dynamics: to attempt to avoid dense crowding, 2) cohesion dynamics: to attempt to move to a central position of local neighbors and establish life spans: 3) reproduction dynamics: to develop a

new generation in a sustained crowded population and assign them initial life spans randomly, 4) extinction dynamics: to determine time of death, and 5) life extension dynamics: to extend the life span in a crowded population in a certain period. The first and second behavioral properties were established in the model of animal collective behavior proposed by Craig Reynolds [2]. The properties of the life span setting are inspired by the bacterial cell-density-dependent gene expression, known as the quorum sensing [3,4]. Fig. 1 schematically illustrates individual properties of microorganism's behavior and their survival.

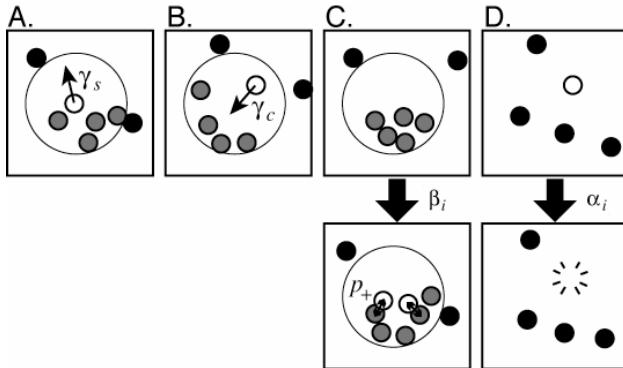


Fig. 1. The schematic illustration of the dynamics of the microorganism model. **A.** Separation dynamics, to set the i -th microorganism (white ball) apart from the central position of the neighbors (gray balls). The movement distance in a time step is determined by the parameter γ_s . **B.** Cohesion dynamics, to approach the i -th microorganism to the central position of the neighbors. The movement distance in a time step is determined by the parameter γ_c . **C.** Reproduction dynamics, to duplicate the i -th microorganisms if it spent time of β_i with neighbors that exceed a certain numbers, N_a . The reproduction ratio from the one is determined by the parameter p_+ . **D.** Extinction dynamics, to eliminate the i -th microorganism that exceeds its life span, α_i , which is defined in birth. The fifth rule is added to extend the life span e to $\alpha_i + e$ if it spent a certain time, t_E , with a certain neighbors, N_b , similarly to the third rule in C. In the simulation of the model, all five rules are calculated individually in each time step. Behavioral rules, A and B, give two vectors for the movement in the same time step, and then the resultant movement is determined by the summation of the two vectors.

3 Fundamental Properties of the Model

To observe the growth and the sustainable property of the model, we defined the initiation phase in the time period $[0, 250]$ and self-sustained phases after the initiation. In the former phase, a microorganism is automatically added in the origin in each time, so that the number of microorganisms becomes 250 at the time 250. After the phase, the self-sustained phase starts to examine the capability of their survival. In the following experimental data, the population of microorganisms expands spatially and forms a star shape. Microorganisms easily survive around the origin because of the high density, giving the life span extensions. On the other hand, the behavioral

rule of the separation dynamics provides them to push out. Interestingly, the global distribution shape does not change dramatically according to time, while individual organisms tend to move across the whole distributions, which cause to reorganize neighboring relationships. The histogram of distances between all pair of organisms is observed as a logarithmic function (Fig. 2F). Parameters of this experiment are given as: $\gamma_s = 1.5 \times 10^{-5}$, $\gamma_c = 2 \times 10^{-5}$, $\beta_i = 1$, $e = 1$, $t_E = 10$, $N_a = 10$, $N_b = 4$, $p_+ = 2$, and α_i is given by the uniform random variable in the range [1, 200].

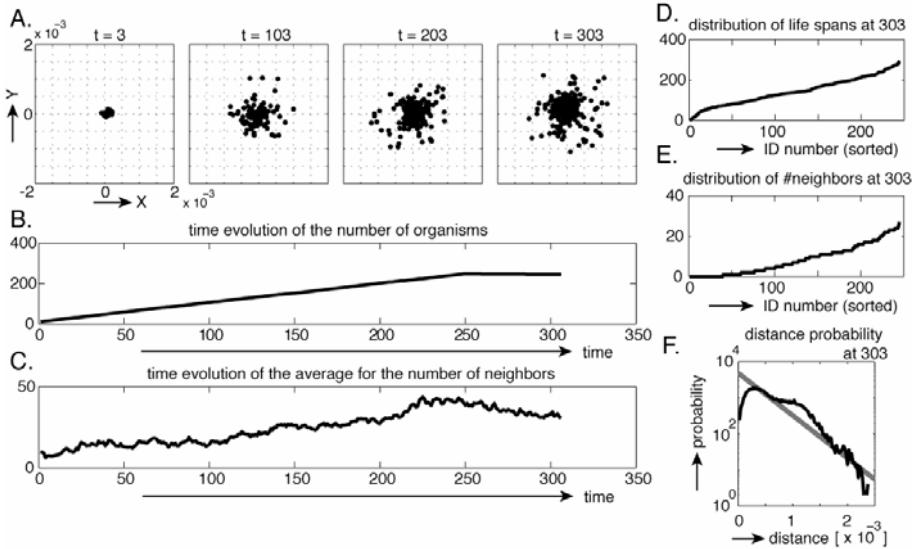


Fig. 2. A. An example of the population growth. The origin is defined as the source of the new generation of organisms in the initiation phase [0 250]. **B.** The time evolution of the number of organisms in the period [0, 300]. **C.** The time evolution of the average for the number of effective neighbors in the range the interaction (Fig. 1). **D.** The distribution of life spans α_i (i = ID number of the organisms that sorted according to the length of the span) at the last moment, $t = 303$. **E.** The distribution of the number of effective neighbors with respect to ID number that sorted according to the number of neighbors at the last moment, $t = 303$. **F.** The logarithmic plot of the histogram of distances between all pair of organisms (black line). The gray line is the linear regression of the data. This clearly has a power law distribution.

The reorganization of the neighbors observed in the long time experiment, as shown in Fig. 3. This property is simply observed in the number of effective neighbors, which mean neighbors in the range of the interaction of the behavioral rules, exhibiting a fluctuation that seems to be an oscillation. One possible hypothesis is that the self-sustained rhythm helps making a shape of the distribution to be suitable for their survival. In ten parameters we set in this experiment, the most important parameter is the living area (LV) because other parameters, such as the behavioral and reproduction speeds can be defined by the relative value to the size of LV. Fig. 3E showed the survival ability of the microorganism's population, which is depending on the LV size.

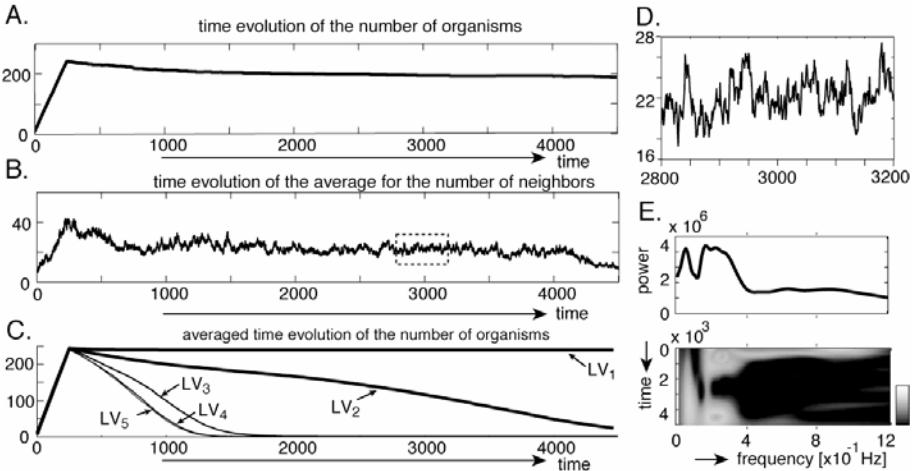


Fig. 3. Fundamental properties of the dynamics of the microorganism model. **A.** The time evolution of the number of organisms in the long period [0, 5000]. In the self-sustained phase, the population stays constantly in the suitable condition. **B.** The time evolution of the average for the number of effective neighbors in the range the interaction (Fig. 1) The living area is restricted in $LV_1 = [x, y] = [1, 1] \times 10^{-3}$. The plot is fluctuated and accompanied with an oscillatory change. The plot in the dotted box is enlarged in **D**. **C.** Survival abilities depending on the living area size, $LV_{2,3,4,5} = 1.25, 1.5, 1.75, 2 \times LV_1$. **E.** The wavelet analysis with the Morlet function in the case of LV_1 . Slow components of frequencies is lasting in the whole period.

4 Real-Environmental Robotic Experiment

As seen in Fig. 3E, when the living size is getting bigger, microorganisms tend to escape to free areas and the population forms a sparse distribution over the whole area, which is difficult for them to find their neighbors and finally going to the death quickly. Therefore, the modulation of the LV size is effective in controlling them.

To examine the real-time adaptability of the model, we propose a framework to connect between the model and the mobile robot moving in the real-environment. In this framework, the distribution shape of the organisms determines the direction and the speed of the robot at each moment. The robot goes forward if they expand upward and backward if they expand downward in the living area. The deviation of the center of the mass in the cluster that expands upward reflects on the direction of the robot movement, as shown in Fig. 4. Microorganisms are living, which attempt to shift to the vacant area avoiding the wall, changing the direction of the robot, which defined by $\lambda \sum_i P_{xi} \cdot P_{yi} / N$ where P_x and P_y are x and y coordinates of the position in the living area, λ is the speed constant, N is the number of organisms with the condition $P_y > 0.0005$. The simulation of the model is done in the computer having the real-time communication with the robot (Fig. 4D). Parameters of this experiment are given as: $\gamma_s = 1.5 \times 10^{-5}$, $\gamma_c = 2 \times 10^{-5}$, $\beta_i = 1$, $e = 1$, $t_E = 5$, $N_a = 10$, $N_b = 4$, $p_+ = 3$,

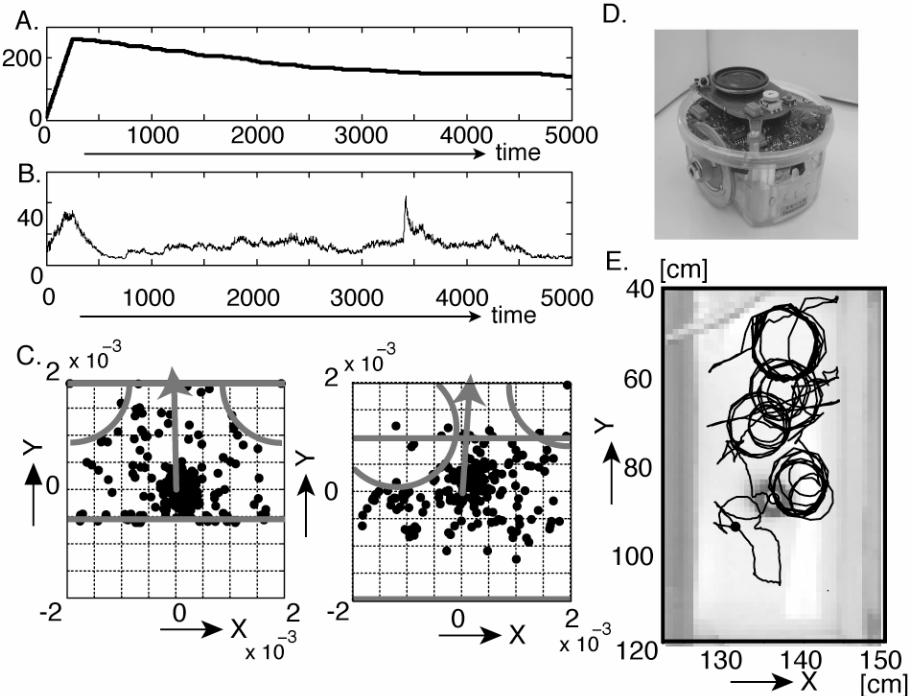


Fig. 4. **A.** The time evolution of the number of organisms in the long period [0, 5000]. In the self-sustained phase, the population gradually decays in this condition. **B.** The time evolution of the average for the number of effective neighbors in the range the interaction, same as Fig.3B. The fluctuation is similarly observed when the model actually connects to the robot. **C.** Examples of situations when the living area is reshaped by the sensor values of the robot. Gray lines in the top and bottom of the area correspond to distance sensors in the front and backward of the robot's body, and gray circles in the right and left correspond to the sensors in the right-forward and left-forward of the body. Those lines and circles move smoothly and minimize the living area when the sensor value increases. **D.** The e-puck mobile robot, with two wheels. **E.** The trajectory of the robot during the experiment, which is superimposed on the top view of the environment. The robot's behavior is monitored by the ceiling camera.

$\lambda = 0.125 \times 10^6$, and α_i is given by the uniform random variable in the range [1, 1500]. The trajectory is obtained four successive experiments. To set the interactive property with the environmental change, the living area is assumed to be reshaped depending on values of distance sensors, which are attached with the robot's body. Although we omit the experimental data in other parameter conditions because of space limitations, the environmental change in the uncoupling condition disrupts the neighboring relationship in the distribution of the microorganism, scatters the colonial structure, and the whole population is going to death rapidly. In the suitable condition, the whole structure is also modulated by such global change; however the speed of the external change and the pace of the internal growth can be coupled in the time constant of

suitable conditions. The internal rhythm, as shown in Fig. 4B, seems to be an index of the healthy growth of the organisms even with the external modulation. This coupling works for the real-time control of the robot, as symbiotic relationship with the host system. In the longtime observation shown in Fig. 4E, behavioral trajectory is relatively smooth including circular patterns. The robot's movement is controlled only by the turning direction, so that this robot system tends to get into an equilibrium state when the robot is located in the free area and having the same movement direction. The living area of microorganisms is hardly reshaped when the robot is in the free area, and microorganisms can form sustained distribution patterns.

5 Concluding Remarks

In the model, microorganisms expanded spatially and exhibited colonial and power law distributions through the time evolution. The neighboring distances, which are crucial for reproduction speeds and are globally modulated by the size of the whole living area, exhibited a rhythmic modulation. In the real-environmental experiment, the robot's navigation was simply demonstrated by producing a temporal adaptability of microorganisms with the size of the living area reshaped according to the current sensory information of the mobile robot. Such a coordination mechanism, known as a synchronization process, has been long studied theoretically using coupled non-linear oscillator models of biological systems; however its functional roles are still argued [5]. As an approach to consider the coordination mechanism, we here proposed a theoretical model of microorganisms interacting with the external environment as an on-line system, by using a real mobile robot. Recently the investigation of biological mechanisms by using the real robot is getting popular [6,7], bridging between the biological plausibility and the functional role in the real-time scale [8]. Focusing on the temporal aspect, we investigate the biological mechanism in the collective behavior with life spans, or alternation of generations. The contribution of the generation of the internal rhythm in microsystems [9], which may bridge between different time scales, is still questionable. This is a first step to investigate the microorganism-based framework to mimic a real-time coordination mechanism between internal and external timescales, by focusing on the oscillatory internal activities. The result may further groundbreaking research of bio-morphological robots [10,11].

Acknowledgments. The author would like to thank Takashi Ikegami, Jun Tani, Ezequiel Di Paolo, and all participants in the Kyoto Agency Workshop 2008 for inspiring the author to view interactive aspects of self-construction processes in a living system, giving an impulse to the concept of the present framework. The author would also like to especially thank Yoko Yamaguchi, who encouraged in proceeding of robotic research projects in the RIKEN BSI and gave advices as need arises.

References

1. Henze, K., Martin, W.: Evolutionary biology: essence of mitochondria. *Nature* 426(6963), 127–128 (2003)
2. Reynolds, C.: Flocks, herds and schools: A distributed behavioral model. In: SIGGRAPH 1987: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pp. 25–34 (1987)

3. West, S.A., Griffin, A.S., Gardner, A., Diggle, S.P.: Social evolution theory for microorganisms. *Nature Reviews Microbiology* 4(8), 597–607 (2006)
4. Francisco, J., Romero-Campero, R.J., Pérez-Jiménez, M.J.: A model of the quorum sensing system in *vibrio fischeri* using P systems. *Artificial Life* 14(1), 95–109 (2008)
5. Yamaguchi, Y., Sato, N., Wagatsuma, H., Wu, Z., Molter, C., Aota, Y.: A unified view of theta-phase coding in the entorhinal-hippocampal system. *Current Opinion in Neurobiology* 17, 1–8 (2007)
6. Jens Ziegler, J., Banzhaf, W.: Evolving control metabolism for a robot. *Artificial Life* 7(2), 171–190 (2001)
7. Verschure, P.F.M.J., Voegtlin, T., Douglas, R.J.: Environmentally mediated synergy between perception and behaviour in mobile robots. *Nature* 425, 620–624 (2003)
8. Wagatsuma, H., Yamaguchi, Y.: Context-dependent adaptive behavior generated in the theta phase coding network. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) *ICONIP 2007, Part II. LNCS*, vol. 4985, pp. 177–184. Springer, Heidelberg (2008)
9. Pirollet, F., Job, D., Margolis, R.L., Garel, J.R.: An oscillatory mode for microtubule assembly. *The EMBO Journal* 6(11), 3247–3252 (1987)
10. Weng, J.: Developmental Robotics: Theory and Experiments. *International Journal of Humanoid Robotics* 1(2), 199–236 (2004)
11. Krichmar, J.L., Edelman, G.M.: Brain-based devices for the study of nervous systems and the development of intelligent machines. *Artificial Life* 11, 63–77 (2005)

Stream Processing of Geometric and Central Moments Using High Precision Summed Area Tables

Chris Messom and Andre Barczak

Institute of Information and Mathematical Sciences, Massey University, New Zealand

{C.H.Messom, A.L.Barczak}@massey.ac.nz

Abstract. This paper introduces a stream programming based design of the zero and higher order central moments that use an integral image or summed area data structure of geometric moments. The stream programming algorithm runs on a general purpose graphics processing unit (GPGPU) that are becoming commodity hardware, giving real-time performance even for large image sizes and a large number of scan window sizes.

1 Introduction

Geometric and central moments are used in the calculation of moment invariants [1,2] that have been used in many image processing systems. Moment invariant classifiers have been studied [3-5] for many different applications. Of particular success have been classifiers for printed and hand written character recognition [3,6] (see figure 1). Moment invariant face detectors (see figure 2) and hand trackers have been developed for human-robot interaction [6]. The main advantages of the moment invariant classifiers over other types of classifiers are that they are scale, translation and rotation invariant. This means that a single classifier can identify the object of interest at any size, position or orientation in the image, reducing the need to implement multiple parallel classifiers [7].

Training of moment invariant classifiers are in general faster than using other feature sets, such as Haar-like features[8]. This is the case as there are a smaller set of moment invariant features, limiting the size and complexity of the classifiers that can be developed, so reducing the training time. This often comes at the expense of discriminatory power, but adds the possibility of dynamically updating classifiers, which is not possible with Haar-like feature based classifiers, where training a boosted classifier often takes weeks rather than days when a reasonably large training set and feature set are used. Although the training time of moment invariant based classifiers are fast, they are in general slow during the execution phase. This is due to the fact that the operations required for the invariant moments are quite complex and only recently have summed area table approaches been adopted to improve performance. A real-time execution environment for moment invariant based classifiers will open the possibility of novel classifier architectures that would otherwise be impractical.

Moment invariants are calculated using the central moments that depend on the geometric moments. The geometric moments can be calculated with a summed area table[9], improving performance significantly. This paper introduces the summed area

table approach to moment invariant classifiers implemented on a GPGPU system. Modern GPUs have adopted the general purpose computing model, allowing traditional graphical hardware operations to be programmed on the fly. Modern GPUs have a significantly larger peak floating point performance than CPUs, 1Tflop and above of single precision peak performance for the new AMD ATI graphics processors HD4850, HD4870 and HD4870X2 versus about 40Gflops of double precision performance for a modern quad core CPUs.



Fig. 1. A CPU based written digit detector using the standard 11 moment invariants

Accessing the peak performance of graphics processors is difficult as the peak performance is obtained when a large number of operations are processed simultaneously, that is GPGPUs are massively parallel processors. This requires the problems that target GPGPUs to be data parallel problems. Many vision processing operations are inherently parallel as each pixel or group of pixels is processed independently. This is also the case for the geometric moments and central moments that are required for the calculation of the invariant moments.



Fig. 2. A CPU based face detector trained using extended set of 55 moment invariant features

Section 2 introduces the geometric moments and the use of the summed area table to improve performance. The stream programming implementation of the summed area table is also introduced in section 2. Section 3 discusses the calculation of the central moments based on the geometric moment summed area tables. Section 4 presents the performance results of the GPGPU implementations of the summed area tables, geometric moments and central moments and compares these to the best available CPU implementations.

2 Geometric Moments

The geometric moments are calculated using the image pixel value and the x and y image coordinate values. This is shown in equation 1.

$$GM_{mn}(a,c,b,d) = \sum_{i=a}^b \sum_{j=c}^d x^m y^n I(i, j) \quad (1)$$

Where $GM_{mn}(a,c,b,d)$ is the $x^m y^n$ geometric moment of the rectangular region of pixels between (a,c) and (b,d) . $I(i,j)$ is the pixel value at position (i,j) and m and n are integers where $m+n$ is between 0 and 4 corresponding to all the geometric moments up to order 4. Geometric moments can be calculated via a summed area table. A summed area table is an array data structure that contains at position (i,j) the sum of all the items in the original array above and to the left of (i,j) . This can be calculated via a serial algorithm shown in equation 2.

$$IGM_{mn}(i,j) = IGM_{mn}(i,j-1) + IGM_{mn}(i-1,j) - IGM_{mn}(i-1,j-1) - I(i,j) \quad (2)$$

Where $IGM_{mn}(i,j)$ is the summed area table of the geometric moments at position (i,j) . The most efficient serial implementation of the summed area table algorithm will have a time complexity of $2MN$ where the array has size MxN . The summed area table or integral image can be efficiently implemented on a GPGPU via a stream programming algorithm based on the standard massively parallel prefix sum algorithm. The summed area table has the advantage that once computed a sum of the values in any rectangular area of the original array can be calculated by a lookup of 4 data items and 3 operators as shown in equation 3.

$$GM_{mn}(a,c,b,d) = IGM_{mn}(a,c) + IGM_{mn}(b,d) - IGM_{mn}(a,d) - IGM_{mn}(b,c) \quad (3)$$

This gives a significant performance gain as large areas of the geometric moment array can be summed efficiently rather than the HxW operations required for summation of a region of height H and width W .

2.1 Stream Programming Algorithm

The summed area tables are implemented in a stream programming paradigm with a modified prefix sum algorithm. The prefix sum algorithm calculates the sum of all data values up to the current position. The prefix sum algorithm is applied to a 1 dimensional array and so has to be extended to a two dimensional system. A high precision (double) implementation of the summed area table is required since for large image sizes the higher order geometric moments are very large values. The higher precision implementation is required to represent these values accurately. The Brook[10] stream programming paradigm implements the parallel component of the algorithm via a kernel that is applied to all the data elements. The prefix sum kernel is shown in figure 3. This calculates the sum of the current data item with the value from a position that is offset from the current position. For the integral image algorithm this offset is either a horizontal or a vertical offset.

```

kernel void doubleprefixsum_scan(double input[][],
                                 out double output<>, float2 twoDoffset) {
    float2 i = indexof(output);
    float2 index=i+twoDoffset;
    if (index.x < 0.0f || index.y < 0.0f){
        output = input[i];
    }
    else{
        output = input[i] + input[index];
    }
}

```

Fig. 3. Prefix sum kernel for a stream processing high precision summed area table algorithm

The stream programming summed area table algorithm is implemented via a horizontal prefix sum followed by a vertical prefix sum. The stream processing horizontal prefix sum algorithm is shown in figure 4. The vertical stream processing algorithm is very similar. The prefix sum algorithms repeatedly call the prefix sum kernel with an exponentially growing offset (either horizontal or vertical offset). An image of size $M \times N$ requires just $\log_2 M + \log_2 N$ cycles through the prefix sum kernel to calculate the summed area table for a given input array.

```

void Horizontal_prefix_sum(double istream<>, out double ostream<>,
                           int Height, int Width){
    double si2<Height, Width>, so2<Height, Width>;
    float2 signWidth, float2 sign;
    unsigned int logNWidth = 0;
    int offset=1, j=0;

    signWidth.x=-1;signWidth.y=0;
    logNWidth = (unsigned int)ceil(log((float)Width)/log(2.0f));

    double_prefix_sum_scan(istream, so2, signWidth);

    for (j = 1; j < logNWidth; ++j){
        offset=(1 << j);
        sign=signWidth*offset;

        if (j & 0x1){
            double_prefix_sum_scan(so2, si2, sign, maxvalue);
        }
        else{
            double_prefix_sum_scan(si2, so2, sign, maxvalue);
        }
        if (!(logNWidth % 2) && (Width !=1)){
            streamSwap(si2, so2);
        }
        streamSwap(so2, ostream);
    }
}

```

Fig. 4. Horizontal prefix sum stream function that calls the prefix sum kernel with an exponentially increasing horizontal offset. The vertical prefix sum function is similar with an exponential increasing vertical offset.

3 Central Moments

Central moments are moments that are calculated based on the centroid of the area of interest. The centroid for the rectangular area is given by (\bar{x}, \bar{y}) . The central moments are calculated via the equation 4.

$$CM_{mn}(a,c,b,d) = \sum_{i=a}^b \sum_{j=c}^d (\bar{x}-x)^m (\bar{y}-y)^n I(i, j) \quad (4)$$

Where $CM_{mn}(a,c,b,d)$ is the central moment of the rectangular region from (a,b) to (c,d) and \bar{x} and \bar{y} are the average x and y values. Both \bar{x} and \bar{y} are constant for a given subwindow. Central moments can be calculated from the summed area tables. This is illustrated by $CM_{11}(a,c,b,d)$ in equation 5:

$$\begin{aligned} CM_{11}(a,c,b,d) &= \sum_{i=a}^b \sum_{j=c}^d (\bar{x}-x) (\bar{y}-y) I(i, j) \\ &= \sum_{i=a}^b \sum_{j=c}^d (xy - \bar{x}\bar{y} + \bar{x}\bar{y}) I(i, j) \\ &= \sum_{i=a}^b \sum_{j=c}^d xy I(i, j) - \sum_{i=a}^b \sum_{j=c}^d \bar{x}\bar{y} I(i, j) - \sum_{i=a}^b \sum_{j=c}^d \bar{x} I(i, j) + \sum_{i=a}^b \sum_{j=c}^d \bar{y} I(i, j) \\ &= GM_{11}(a,c,b,d) - \bar{y}GM_{10}(a,c,b,d) - \bar{x}GM_{01}(a,c,b,d) + \bar{x}\bar{y}GM_{00}(a,c,b,d) \\ &= GM_{11}(a,c,b,d) - GM_{01}(a,c,b,d) GM_{10}(a,c,b,d)/GM_{00}(a,c,b,d) \\ &\text{since } \bar{x} = GM_{10}(a,c,b,d)/GM_{00}(a,c,b,d) \text{ and } \bar{y} = GM_{01}(a,c,b,d)/GM_{00}(a,c,b,d) \end{aligned} \quad (5)$$

In a similar manner all the central moments can be calculated based on the geometric moment summed area tables. These are shown in equations 6 with the (a,c,b,d) suppressed.

$$\begin{aligned} CM_{20} &= GM_{20} - GM_{10}^2/GM_{00} \\ CM_{30} &= GM_{30} - 3GM_{10}GM_{20}/GM_{00} + GM_{10}^3/GM_{00}^2 \\ CM_{21} &= GM_{21} - 2GM_{10}GM_{11}/GM_{00} - GM_{01}GM_{20}/GM_{00} + 2GM_{10}^2GM_{01}/GM_{00}^2 \\ CM_{22} &= GM_{22} - 2GM_{10}GM_{21}/GM_{00} + GM_{01}^2GM_{20}/GM_{00}^2 - 2GM_{10}GM_{12}/GM_{00} \\ &\quad + 4GM_{10}GM_{01}GM_{11}/GM_{00}^2 - 3GM_{10}^2GM_{01}^2/GM_{00}^3 + GM_{10}^2GM_{02}/GM_{00}^2 \\ CM_{31} &= GM_{31} - GM_{01}GM_{30}/GM_{00} + 3GM_{01}(GM_{01}GM_{20}/GM_{00} - GM_{21})/GM_{00} \\ + 3GM_{10}^2(GM_{11} - GM_{10})/GM_{00}^2 + GM_{10}^3(GM_{01} - GM_{10})/GM_{00}^3 \\ CM_{40} &= GM_{40} - 4GM_{10}GM_{30}/GM_{00} + 6GM_{10}^2GM_{20}/GM_{00}^2 \\ &\quad - 4GM_{10}^3GM_{01}/GM_{00}^3 + GM_{10}^4/GM_{00}^3 \end{aligned} \quad (6)$$

The equations for the central moments CM_{02} , CM_{03} , CM_{12} and CM_{13} can be obtained via symmetry from those shown in equations 6.

4 Performance Results

The double precision implementation of the summed area table comes at the expense of some performance as shown in 5. GPUs are optimized for single precision floating point operations and so using double precision results in the performance penalty. However for the higher order moments the double precision is required so that the values are represented to sufficient accuracy for the classifier to be able to distinguish between distinct images but with identical single precision moments.

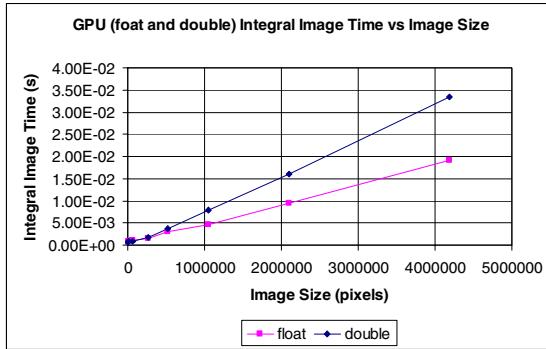


Fig. 5. Performance of summed area table subsystem for single and double precision

4.1 Multiple Kernels

The calculation of each of the central moments can be implemented in a corresponding kernel. However this unnecessarily duplicates some computation and also significantly increases the number of kernels. An alternative approach is to combine the calculation of the central moments in a single kernel. However the restriction that the AMD Brook+ system (and the associated hardware) can only support a maximum of 8 stream outputs means that the 13 central moments must be processed in two separate kernels, one processing the lower order kernels and the other the higher order kernels.

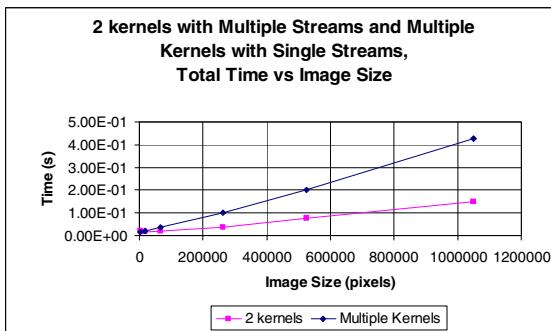


Fig. 6. Performance of system to calculate the central moment over a range of image sizes using a moment window of size 64x64

Figure 6 shows the performance comparison of the two approaches, dedicating a kernel for each central moment versus the two kernels that calculate all the central moments. As can be seen from the graph the two kernels are significantly more efficient. This is due to the computational intensity of the kernels that calculate multiple central moment values being higher than the individual kernels. The multiple output stream approach is more than twice as fast as the multiple kernel single output stream approach.

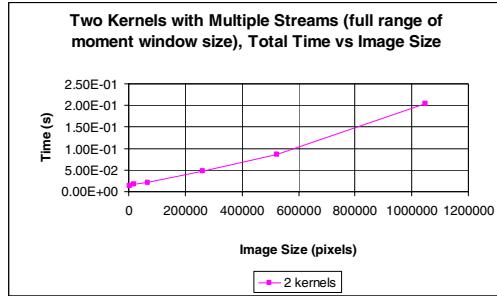


Fig. 7. Performance of system to calculate the multiple scans of the central moment over a range of image sizes using initial moment window of size 20x20 and increasing scale by 1.1. This uses the two kernels and multiple streams to calculate all the central moments.

The central moments are normally calculated over a range of window sizes. This allows objects that are of different sizes to be identified by the classifiers. Figure 7 shows the performance of the system where a range of window sizes from 20x20 up to the size of the image are calculated with an exponentially increasing scan window size (increasing by 1.1 at each stage). This is a more realistic scenario rather than a system that calculates the central moment at just one window size. The large number of scans do not significantly increase the run time as when only a single scan window size is used the run time is dominated by the geometric summed area table calculations. This means that real-time performance can be maintained even if a large number of scan window sizes are used.

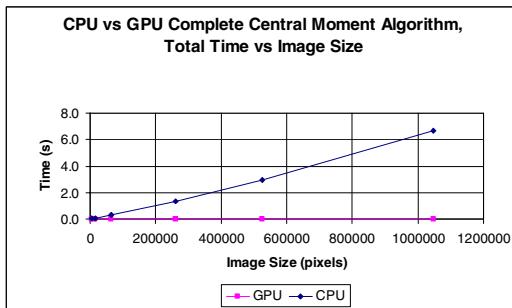


Fig. 8. Performance comparison of central moment algorithm implementation. GPU versus best available CPU implementation.

Figure 8 shows the performance comparison of the GPU implementation versus the best available CPU implementation of the central moments calculation using geometric moment summed area tables. There is a significant performance difference between the two with the CPU implementation not able to reach real-time performance except for very small image sizes.

5 Conclusions

This paper introduced a stream programming implementation of the central moment calculations based on a summed area table implementation of the geometric moments. This was shown to be capable of real-time performance (more than 10 frames per second) when using large images (512x512) and a large number of scan window sizes. The real time performance allows more complex adaptive moment invariant architectures to be developed and investigated in the future.

Acknowledgments. This research was supported by the Tertiary Education Commission of New Zealand's Broadband Enabled Science and Technology GRID project.

References

1. Hu, M.K.: Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory* 8, 179–187 (1962)
2. Flusser, J.: On the Independence of Rotation Moment Invariants. *Pattern Recognition* 33, 1405–1410 (2000)
3. Barczak, A.L., Johnson, M.J., Messom, C.H.: Revisiting Moment Invariants: Rapid Feature Extraction and Classification for Handwritten Digits. In: *Proceedings of IVCNZ* (2007)
4. Barczak, A.L.C., Dadgostar, F., Messom, C.H.: Real-Time Hand tracking based on non-invariant features. In: *IEEE Instrumentation and Measurement Technology Conference*, Ottawa, Canada, pp. 2192–2197 (2005); ISBN 0-7803-8879-8
5. Barreto, J., Menezes, P., Dias, J.: Human-robot interaction based on haar-like features and eigenfaces. In: *International Conference on Robotics and Automation*, New Orleans (2004)
6. Barczak, A.L.C.: Feature Based Rapid Object Detection: From Feature Extraction to Parallelisation, Phd Thesis, Massey University (2007)
7. Rowley, H., Baluja, S., Kanade, T.: Neural Network Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Vision* 20, 23–38 (1998)
8. Jones, M., Viola, P.: Fast Multi-view Face Detection, Mitsubishi Electric Research Laboratories, TR2003-96 (2003)
9. Crow, F.C.: Summed-area tables for texture mapping. *Computer Graphics* 18, 207–212 (1984)
10. Buck, I., Foley, T., Horn, D., Sugerman, J., Fatahalian, K., Houston, M., Hanrahan, P.: Brook for GPUs: Stream computing on graphics hardware. *ACM Trans. Graph.* 23(3), 777–786 (2004)

Bayesian Fusion of Auditory and Visual Spatial Cues during Fixation and Saccade in Humanoid Robot

Wei Kin Wong¹, Tze Ming Neoh¹, Chu Kiong Loo¹, and Chuan Poh Ong²

¹ Center of Robotics and Automation, Multimedia University,

Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia

² Intel Malaysia Sdn. Bhd., Bayan Lepas Free Industrial Zone Phase 3,
11900 Penang, Malaysia

Abstract. In this paper, the Bayesian fusion of auditory and visual spatial cues has been implemented in a humanoid robot aiming to increase the accuracy of localization, given a situation that an audiovisual stimulus was presented. The performance of auditory and visual localization was tested under two conditions: fixation and saccade. In this experiment, we proved that saccade did greatly reduce the accuracy of auditory localization in the humanoid robot. The Bayesian model became not reliable when the results of auditory and visual localization were not reliable, particularly during saccade. During the tests, localization in two conditions (saccade onset and changing of direction of motion) has been ignored and only azimuth position has been considered.

1 Introduction

Studies on behavior of biological subjects suggest that the brain is able to integrate every available sensory cue from sensors primarily in the midbrain super colliculus (SC). The fusion of multisensory cue could improve quality of perception including localization. As a result, the way the brain fuses the sensory cues has became an interesting topic to neuroscientists and many of them focused their study on human's response on the audiovisual stimulus recently [1-6], aiming to model the fusion of multimodal cues.

Previously, robot scientists have conducted research on how to locate objects more accurately in their robot, if audiovisual target is presented. They developed methods to describe the fusion of auditory and visual cues, mainly by applying neural network. For example, Lorenzo Natale et al [8] used a self-supervised neural network base on visual feedback to gaze towards audiovisual source. Carlos Beltrán-González's et al [9] model measured the level of synchrony of auditory and visual data streams using the method suggested by Hershey and Movellan based on mutual information. After that, they represented the output of mutual information as mixel. Hans-H. Bothe et al [10] utilized neural network to compare visual spatial map with auditory spatial map by calculating Gaussian function of each pixel and average the result. However, neural network requires considerable knowledge and programming techniques. Thus, another biological plausible and simpler solution is desired.

Reviewing works done in neuroscience, we found that the Bayes' Theorem (1764) had been applied to model the fusion of auditory and visual spatial cues [1,3] recently.

Particularly in Paola Binda's et al research, they found that during fixation, visual localization was much more precise than auditory localization. Bimodal conflicting stimuli were localized at the place where their visual component was actually displayed. For stimuli presented saccadically, the auditory performance remained unchanged, but visual localization became less precise and grossly biased towards the direction of saccadic motion. The perceived position of bimodal stimuli was intermediate between perceived position of their visual and auditory components. According to their model, standard deviations of visual and auditory localization were used to estimate the corresponding weight, which was used to estimate the position of bimodal stimulus. Their Bayesian model predicted the result of bimodal localization during saccade accurately as been proven by their data and was simpler to apply.

In this paper, we modeled the fusion of bimodal spatial cues with the Bayes' Theorem, similar to Paola Binda's et al work, and conducted tests to evaluate the performance of this model. We intended to investigate whether Baye's Theorem helps to generate better localization result during fixation and saccade in robotics system. In the next section, the Bayesian model is presented. A detail description of the hardware, localization methods, experimental procedure and experimental setup are provided in section 3. Section 4 contains the test results and discussion. The last section, section 5, covers the conclusion of this experiment.

2 The Bayesian Model

By assuming the Bayesian integration of auditory and visual spatial cues, the relationship between audiovisual, auditory and visual spatial cues is given by [1]:

$$S_{AV} = w_A S_A + w_V S_V \quad . \quad (1)$$

S_{AV} is the estimated audiovisual spatial location, S_A is the auditory spatial location and S_V is the visual spatial location. w_A and w_V are weights by which the unimodal spatial location are scaled, related to standard deviation (SD) of auditory localization σ_A and SD of visual localization σ_V by the following:

$$w_A = k/\sigma_A^2, w_V = k/\sigma_V^2 \text{ where } k = 1/(1/\sigma_A^2 + 1/\sigma_V^2) \quad . \quad (2)$$

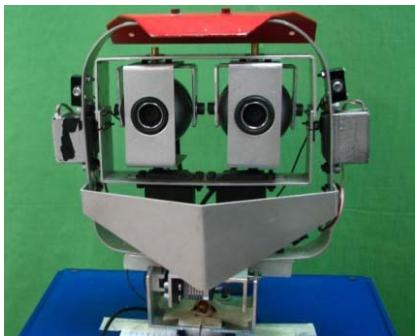
The predicted SD of bimodal σ_{AV} is given by the following:

$$\sigma_{AV} = \sqrt{\sigma_A^2 \sigma_V^2 / (\sigma_A^2 + \sigma_V^2)} \quad . \quad (3)$$

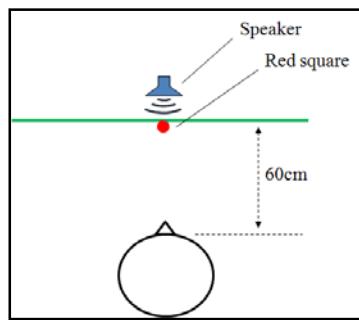
3 Experimental Setup

The most important apparatus in this experiment was a humanoid robot with a 5-DOF head, shown in Figure 1(a). The head was purposely set to turn left and right at its neck, whereas other joints were set to face forward and were fixed during the experiment. The neck was driven by a HITEC HSR-5995TG servomotor, connected to Lynxmotion SSC-32 servo controller. Logitech Quickcam Pro 5000 was used as

the image capturing device (resolution was 176x144 pixels, 256 RGB color and frame rate was 30 frames/sec). The image captured was then been transformed into Log-Polar image (80x32 pixels) [12, 13] to reduce the amount of information. Color segmentation was performed based on the Log-Polar image in HSV color space with the assistance of H-S histogram (15x15 pixels) [12]. On the other hand, a pair of omni-directional ECM PC-50 mini electret microphones were used to record the sound. Left and right signal from microphones were compared using generalized cross-correlation (GCC) algorithm to obtain their interaural time difference (ITD) in frequency domain [8] with PHAT as the pre-filter [11]. Eye-center, ear-center and head center were assumed to have a common tilt plane.



(a)



(b)

Fig. 1. (a) is the 5-DOF robot head used in the experiment and (b) is the bird eye view of the experiment setup. The positions of visual and auditory stimulus were fixed and the head turned relative to the stimulus.

The background of image was a static color background in green color, located in front of the humanoid robot, with a distance of 60cm from the eye-center (please refer to Figure 1(b)). The visual stimulus was presented as a 2cmx2cm red square on the green background. Artificial lighting (fluorescent light) was used with no additional light source such as sun light. The auditory signal was presented as continuous white noise, played through a normal speaker. The audiovisual stimulus was virtually ‘move’ within a range of 20° to the left and right of the robot. However, due to hardware limitation, the neck was turned relatively to the stimulus, instead of driving the audiovisual stimulus to move. The sampling rate of the program was 6data/sec. The transformation from eye-center coordinate to head-center (ear-center) coordinate was calibrated manually before the test. The robot was controlled by Matlab, operated under Window XP. The personal computer (PC) used was DELL Precision 490, which contained Intel Xeon E5335 Quad-core processor, 2GB RAM, 100Mbps Ethernet, STAC9200 soundcard and FireGL 128MB graphic card.

Localization performance under two conditions: fixation and saccade, was tested. During fixation test, the robot head and eyes were initially fixated towards the background at position (0°, -20°) (elevation angle of 0° and azimuth angle of -20°; positive value indicates right and up of the humanoid robot while negative value

indicates left and down). The head was then commanded to turn (10 stepping and fixate for 1sec after each step) until it reach (0° , $+20^\circ$) and back to (0° , -20°). The test was repeated 3 times. Positions of auditory and visual stimulus were calculated each time after the robot stepped and fixated. During saccade test, the neck was turned by three different angular speeds (20deg/sec, 30deg/sec and 60deg/sec). The robot head was firstly fixate at (0° , -30°). Once the tests start, the head was commanded to move to (0° , $+30^\circ$) and back to (0° , -30°). The program kept track of the position of the audiovisual stimulus continuously, by applying the visual and auditory localization algorithm. The test was repeated 50 times in order to minimize the influence of random hardware error. The motion range was extended to 30° at both sides during saccade test to avoid obtaining data under two situations: saccade onset and changes of direction of motion. During these three conditions, the image capture was noisy and was not reliable. However, only those data fell within the range (20° to the left and right) were recorded. The robot head was assumed to turn with a constant speed until the end of motion, after a short time from saccade onset and changes of direction of motion. Thus, the actual azimuth position of robot head can be estimated by a simple mathematic equation.

4 Results and Discussion

Total of five tests have been conducted in this experiment. They were fixation test, 20deg/sec saccade test, 30deg/sec saccade test, 60deg/sec saccade test and condition specific calibration saccade test. For each of these tests, SD of unimodal localization has been calculated and been substituted into Eq.1 and Eq.2 to estimate the result of Bayesian bimodal localization.

Figure 1 demonstrates the result of fixation test. Calibration has been done based on the raw result of the test. The final result indicates that during fixation, auditory and visual localization was accurate with small mean mislocalization of 0.7159° and 0.7683° respectively. During the following test (20deg/sec saccade test, 30deg/sec saccade test and 60deg/sec saccade test), same set of calibration as in fixation test had been used. The head turned in three different angular speeds (20deg/sec, 30deg/sec and 60deg/sec) during each of the tests. According to the results, the mean mislocalizaion of unimodal localization increased as the turning speed of robot head increased. During these three tests, the images captured were blur and were not reliable. This phenomenon affected the result more and more significantly as the angular speed of the robot head increased, especially during the time saccadic motion starts and during changes of direction of motion. Table 1 summarized the mean and SD of mislocalization of unimodal and Bayesian bimodal localization, as well as the calculated weights of unimodal localization of each test. The σ_{AV} in Table 1 was calculated by comparing the result of Bayesian bimodal stimulus with the ideal position. It is clear that Bayesian fusion of auditory and visual spatial cues did improve the accuracy of localization as the mean mislocalization of Bayesian bimodal localization were smaller then mean mislocalization of visual localization all of the time. During all saccade tests, auditory localization became more reliable than visual localization. However, the results of auditory localization during saccade test were not reliable since they deviated greatly (3° to 8°) from ideal position throughout the tests.

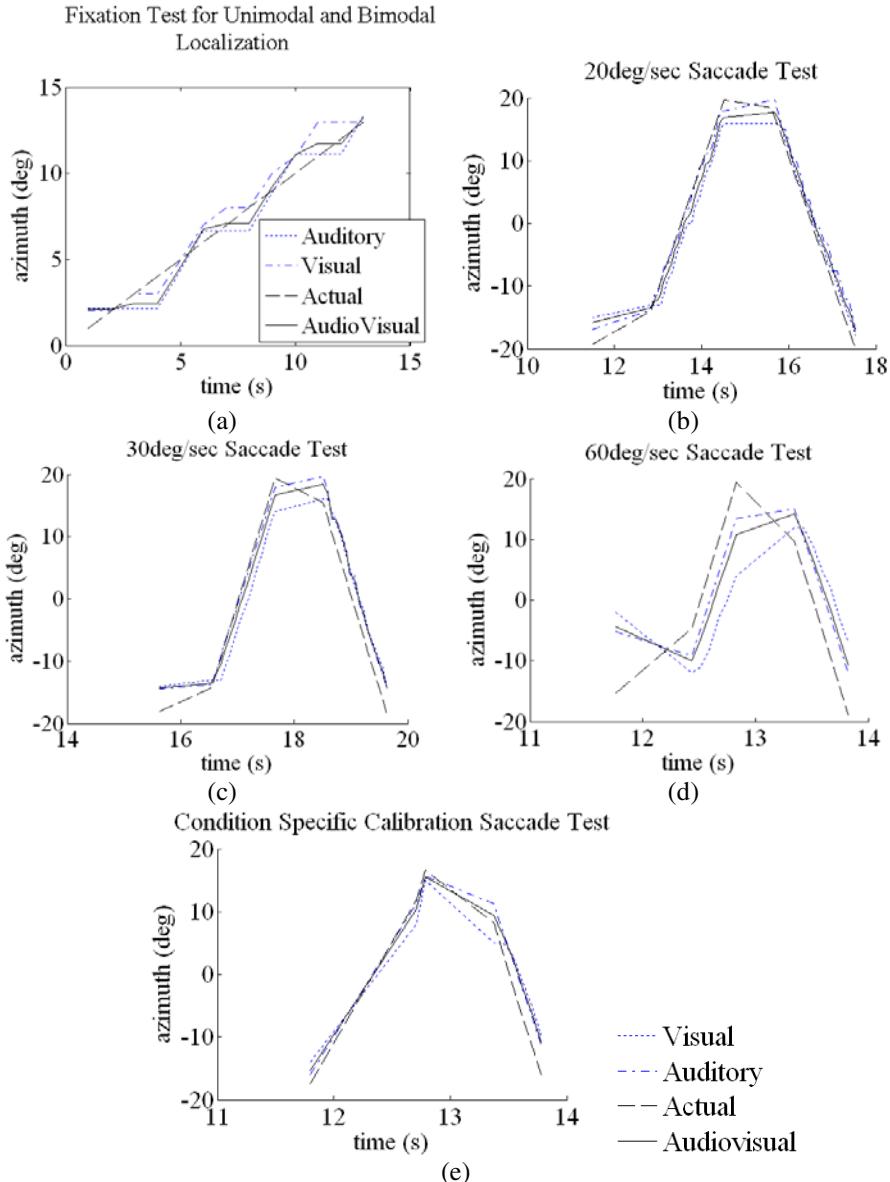


Fig. 2. (a) demonstrates a portion of the results of unimodal and bimodal localization during fixation test. Both unimodal (auditory and visual) localizations were accurate and Bayesian fusion of bimodal spatial cues did improve the localization performance as the SD was smaller. (b) (c) (d) are the result of 20deg/sec, 30deg/sec and 60deg/sec saccade test respectively. The unimodal localization algorithm mislocated the audiovisual stimulus more and more seriously when the turning speed of the robot head increased. Bayesian model generated distorted spatial position of the audiovisual stimulus when both of the unimodal localization algorithms were not reliable. (e) is the result of condition specific calibration saccade test. Both unimodal localizations have been recalibrated under 60deg/sec saccade condition and the result shows encouraging improvement.

Table 1. Comparison of mean (\bar{x}) and SD (σ) of mislocalization of auditory, visual and Bayesian audiovisual localization, as well as the weight (w) of unimodal localization

Test	Auditory localization			Visual localization			Bayesian audiovisual localization	
	\bar{x}_A	σ_A	w_A	\bar{x}_V	σ_V	w_V	\bar{x}_{AV}	σ_{AV}
Fixation	0.7159	0.8732	0.6897	0.7683	1.3020	0.3103	0.6341	0.7905
20deg/sec saccade	1.9910	2.5389	0.4636	2.0713	2.3605	0.5364	1.7941	2.1015
30deg/sec saccade	2.7428	3.3786	0.6584	4.2705	4.6905	0.3416	3.1689	3.6338
60deg/sec saccade	6.6249	6.9261	0.7239	10.7557	11.2159	0.2761	7.7643	7.9769
Condition specific calibration saccade	1.9197	2.3324	0.7132	2.9261	3.6779	0.2868	1.9492	2.3861

The result of Bayesian bimodal localization was also distorted greatly when both unimodal localizations were not accurate. This has clearly demonstrated the weakness of Bayesian model that, during the absence of at least one accurate unimodal localization system, such as auditory localization system of human [1], the result of Bayesian model could became not reliable. In order to overcome the weakness of Bayesian model, a possible solution has been proposed, aiming to increase the accuracy of unimodal localization during saccade. The robot system has been recalibrated based on the result of 60deg/sec saccade test, in order to obtain the condition specific calibration value specifically for 60deg/sec saccade condition. Figure 5 shows the result of this test. Compare to the 60deg/sec saccade test, the mean mislocalization of unimodal localization was much smaller and the result of Bayesian bimodal localization was closer to ideal position. However, the weakness of this solution was that, the humanoid robot must always aware of the angular speed of its head and condition specific calibration must be available to the robot.

5 Conclusion

The robot locate spatial position of an audiovisual stimulus differently compare to biological subject. As Bayesian fusion of auditory and visual spatial cues can sufficiently model the result of bimodal localization of biological subject in either fixation or saccade, we have shown in this experiment that it might not be able to generate correct spatial position of audiovisual stimulus when both of the unimodal localization systems were not reliable, especially during saccade. However, we also shown that by using condition specific calibration, the result can be improve and was reasonably accurate. The future works include advance tuning of Bayesian model, real-time implementation of the Bayesian model and using asymmetric multiprocessing (AMP) operating system to increase the processing speed of localization algorithm.

Acknowledgement

We would like to thank Noel Tay of Center of Robotics and Automation of Multimedia University who helped to conduct the experiment. Without his help this experiment could not been done smoothly. This research was funded by Intel Malaysia Sdn. Bhd. during year 2007/08.

References

1. Bindu, P., Bruno, A., Burr, D.C., Morone, M.C.: Fusion of Visual and Audio Stimuli during Saccades: A Bayesian Explanation for Perisaccadic Distortions. *The Journal of Neuroscience* 27(32), 8525–8532 (2007)
2. Burr, D., Alais, D.: Combining Visual and Auditory Information. *Progress in Brain Research* 155, 243–255 (2006)
3. Kording, K.P., Beierholm, U., Ma, W.J., Quartz, S., Tenenbaum, J.B., Shams, L.: Causal Inference in Multisensory Perception. *PLoS ONE* 2(9), e943 (2007)
4. Wallace, M.T., Roberson, G.E., Hairston, W.D., Stein, B.E., Vaughan, J.W., Schirillo, J.A.: Unifying Multisensory Signals across Time and Space. *Exp. Brain Res.* 158, 252–258 (2004)
5. Steenken, R., Colonius, H., Diederich, A., Rach, S.: Visual-Auditory Interaction in Saccadic Reaction Time: Effects of Auditory Masker Level. *BRES-37260 4C*, 3–4 (2007)
6. Bolognini, N., Rasi, F., Làdavas, E.: Visual Localization of Sound. *Neuropsychologia* 43, 1655–1661 (2005)
7. Mishra, J., Martinez, A., Sejnowski, T.J., Hillyard, S.A.: Early Cross-Model Interactions in Auditory and Visual Cortex underlie a Sound-Induced Visual Illusion. *The Journal of Neuroscience* 27(15), 4120–4131 (2007)
8. Natale, L., Metta, G., Sandini, G.: Development of Auditory-Evoked Reflexes: Visuo-Acoustic Cues Integration in a Binocular Head. *Robotics and Autonomous System* 39, 87–106 (2002)
9. Beltrán-González, C., Sandini, G.: Visual Attention Priming Based on Crossmodal Expectation. Laboratory for Integrated Advance Robotics, University of Gonova
10. Bothe, H.-H., Persson, M., Biel, L., Rosenholm, M.: Multivariate Sensor Fusion by a Neural Network Model. Applied Autonomous Sensor Systems Laboratory (AASS), Örebro University
11. Kee, K.C., Loo, C.K., Khor, S.E.: Sound Localization using Generalized Cross Correlation: Performance Comparison of Pre-Filter. Center of Robotics and Automation, Multimedia University (2008)
12. Metta, G., Gasteratos, A., Sandini, G.: Learning to Track Colored Objects with Log-Polar Vision. *Mechatronics* 14, 989–1006 (2004)
13. Berton, F.: A Brief Introduction to Log-Polar Mapping (2006)

Solving the Online SLAM Problem with an Omnidirectional Vision System

Vitor Campanholo Guizilini and Jun Okamoto Jr.

Advanced Perception Laboratory

Department of Mechatronics and Mechanical Systems Engineering

Escola Politécnica da Universidade de São Paulo

Av. Prof. Mello Moraes, 2231, São Paulo, SP 05508-030 Brazil

vitor.guizilini@gmail.com, jokamoto@usp.br

Abstract. A solution to the problem of simultaneous localization and mapping, known as the problem of SLAM, would be of inestimable value to the field of autonomous robots. One possible approach to this problem depends on the establishment of landmarks in the environment, using artificial structures or predetermined objects that limit their applicability in general tasks. This paper presents a solution to the problem of SLAM that relies on an omnidirectional vision system to create a sparse landmark map composed of natural structures recognized from the environment, used during navigation to correct odometric errors accumulated over time. Visual sensors are a natural and compact way of achieving the rich and wide characterization of the environment necessary to extract natural landmarks, and the omnidirectional vision increases the amount of information received at each instant. This solution has been tested in real navigational situations and the results show that omnidirectional vision sensors are a valid and desirable way of obtaining the information needed to solve the problem of SLAM.

1 Introduction

A truly autonomous robot should be able to build a model of the environment around it, mapping the regions it has already visited, as well as dynamically localizing itself inside this model as its position changes. Ideally, these two problems should be solved during navigation using nothing but the robot's own embedded sensors and without any prior knowledge of the environment, thus maximizing the field of applications where these abilities can be used. Any sensor used to collect the necessary information will inevitably be imprecise, due to events that cannot be modeled because they are too complex or unpredictable. Probabilistic approaches present straightforward solutions to each problem individually, such as localization given the map [1] or mapping given the localization [2]. This is possible because localization and mapping are mutually dependent, so the knowledge of one is necessary to ensure precision in the other.

However, there are situations where neither localization nor mapping is known in advance, so the robot must estimate both during navigation, creating a

problem known a SLAM. In this case it is not possible to address each case individually, as errors from each estimative will propagate to the other and rapidly accumulate, invalidating both results. The first formalized solution to the problem was the EKF-SLAM, proposed in [3], where the robot uses landmarks from the environment to localize itself. Since then, an extensive research [4][5] has been made as an attempt to overcome some limitations of EKF-SLAM, such as inefficiency in dealing with large amounts of landmarks [6][7], data association problems [8] and negative information [9].

Efforts have also been made to develop more robust ways of landmark detection, increasing the amount of reliable information available at each moment. While range sensors provide ready-to-use distance measurements and are capable of predetermined landmark detection [10], visual sensors provide a much richer characterization of structures, and recent increases in computational power and algorithm efficiency have made possible to extract landmarks directly from natural structures of any kind [11], along with robust matching and object recognition under different points of view [12]. Among visual sensors, omnidirectional systems allows the robot to observe the entire environment around it simultaneously [13], increasing the amount of landmarks detected and the period each one remains visible, characteristics that have been explored in numerous applications [14][15].

This article presents a solution to the problem of SLAM that uses an omnidirectional vision sensor to collect the information necessary to build a sparse map of environment, composed of natural landmarks passively detected, and use this map to correct odometric errors accumulated over time. A precise localization allows for a precise mapping and vice-versa, maintaining the error levels stable for long periods of time, as shown in the results obtained.

2 Proposed Solution

We assume here the online use of the results obtained in an autonomous navigation and no prior knowledge of the environment. At $t = 0$ the robot is at an arbitrarily defined position and its landmark map is empty. As it moves, its localization is updated according to odometry measurements and the uncertainty regarding this estimative increases. At the same time, an omnidirectional image is collected by the vision system and parallel processed using the SIFT algorithm [12] to extract a set of tridimensional landmarks representing the structures around the robot. When this information is ready it is incorporated into the robot's localization and mapping estimatives according to Fast-SLAM [6], increasing its knowledge of the environment and eliminating errors accumulated during the blind navigation. A diagram of this solution is presented in Fig. 1, showing the internal steps of each individual algorithm and how they communicate to create a closed solution to the problem of SLAM as proposed.

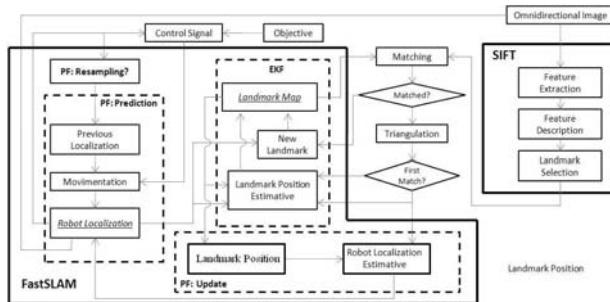


Fig. 1. Diagram of the SLAM solution proposed

2.1 Landmark Selection

The SIFT algorithm used in this paper for feature detection was proposed by Lowe [12] as a method of extracting and describing local features which are robustly invariant to common image transforms. Through searching in different scale spaces and main orientation it is possible to achieve scale and rotation invariance, both desirable properties for feature detection directly in omnidirectional images. Rotational invariance is important because objects detected can appear in any orientation depending on the angle between them and the robot, and so is scale invariance since resolution rapidly decreases in the outer ring of the image, changing the apparent size of observed objects.

The SIFT features obtained must be transformed into the landmarks that will be used by the SLAM algorithm. Possible approaches include direct use of features as landmarks [15], a database of predetermined objects [16] or geometric forms common in certain environments (such as vertical lines in [17]). However, while the former one is prone to false matches in cases of feature similarity and has to deal with a high computational cost created by the large amounts of landmarks generated, the later ones limit the applicability of the solution by establishing a prior knowledge of the environment.

Because of these limitations we developed a method of feature grouping that is based solely on image properties, and therefore can be used equally in any kind of environment. Each feature is compared to the others in terms of pixel distance and intensity difference, and if one is found within predetermined thresholds they are considered part of the same "object" and the process is recursively repeated. After all features have been compared the ones that don't have a minimum of peers are discarded, while the others are considered as landmarks. Each one has its own position estimative, but shares the same object index, which is used during the matching step. If an object is not matched within a certain feature percentage the matches are discarded as false, and if a feature is consistently not matched while its object is it can be discarded as not relevant to the mapping anymore. The result is a smaller amount of landmarks per image, but the ones selected will be more stable, since the probability of a group of features being mismatched is smaller than in individual matches.

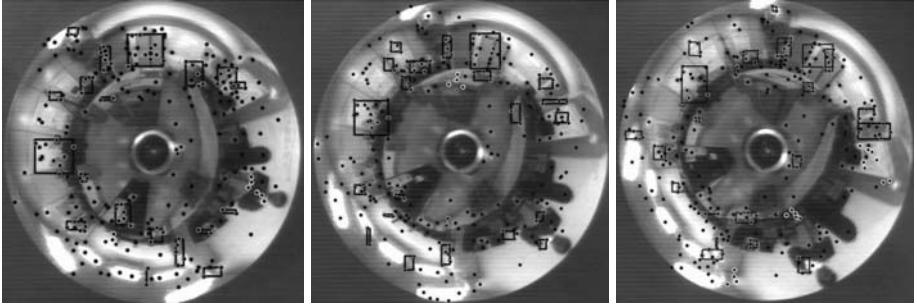


Fig. 2. SIFT results on omnidirectional frames (taken in intervals of 2 seconds). Small dots indicate features, circled dots indicate landmarks and rectangles indicate single objects recognized from previous images.

2.2 Incorporating Information

Once the landmark set from one image is obtained, it is used to refine the robot's mapping and, if possible, localization estimatives. The FastSLAM algorithm uses a particle filter [18] to model the robot's localization uncertainty and updates each landmark observed independently with Extended Kalman Filters [19] according to the independency notion stated in [20] and held true if the robot's position is assumed known, which is possible within each particle's hypothesis. This factorization of the SLAM problem increases drastically its efficiency in dealing with large amounts of landmarks, something especially valuable when visual sensors are used. Each particle also keeps an independent mapping hypothesis and thus can deal naturally with data association problems, following different scenarios and discarding the ones that are eventually seen as inaccurate without compromising the final result.

Mapping. When an observation z_k^t generates a landmark, this landmark is compared to the ones already stored and if no match is found it is added as a new variable to each particle's mapping hypothesis. A newly added landmark $m_{n,p}^t$ has three components: the position x_p^t of the robot when the image was taken, the coordinates c_n^t of the feature that originated the landmark in the image and the vector v_n that describes the properties of this feature. This vector is used in the matching step, and if this landmark is matched with a landmark observed at a later stage the other two components are used in the triangulation process [13] to obtain the full observation vector $z_{k,p}^t$ (different for each particle) and consequently an position estimative $\theta_{n,p}^t$, along with a covariance matrix $\Sigma_{n,p}^t$ that quantifies the uncertainty inherent to this estimative.

$$m_{n,p}^t = (x_p^t, c_n^t, v_n, \theta_{n,p}^t, \Sigma_{n,p}^t) \quad (1)$$

From now on, new matches of the landmark will generate new position estimatives that are incorporated to the current one in order to refine it, decreasing its

initial uncertainty. A landmark is assumed to be static, so its position and uncertainty does not change over time. The difference between the position estimative $\theta_{n,p}^t$ stored in a particle's map and the one obtained with the full observation $z_{k,p}^p$ is given by (2), where $h(x_t^p, \theta_{n,p}^t)$ is the function that connects the landmark position with the robot position using the observation vector.

$$y_{k,n}^{p,t} = z_{k,p}^t - h(x_t^p, \theta_{n,p}^t) \quad (2)$$

The estimative update is done using the EKF equations, which require the linearization of $h(x_t^p, \theta_{n,p}^t)$, given here by the jacobian $H_{n,p}^t$. The matrix Q in (3) represents the error model of the sensor, here assumed constant over time but variable along the radius of the image, representing the decrease in resolution.

$$S_{n,p}^t = (H_{n,p}^t) \Sigma_{n,p}^t (H_{n,p}^t)^T + Q \quad (3)$$

$$K_{n,p}^t = \Sigma_{n,p}^t (H_{n,p}^t)^T (S_{n,p}^t)^{-1} \quad (4)$$

$$\theta_{n,p}^{t+1} = \theta_{n,p}^t + K_{n,p}^t (y_{k,n}^{p,t}) \quad (5)$$

$$\Sigma_{n,p}^{t+1} = (I - K_{n,p}^t H_{n,p}^t) \Sigma_{n,p}^t \quad (6)$$

Localization. At the beginning of navigation the robot's localization is considered known, so all the particles have the same position and weight. As the robot moves, the position x_{t+1}^p of each particle is propagated according to a transition function (7) that depends on the particle's previous localization estimative x_t^p and the control vector u_t composed of a linear v and angular ω velocities. This propagation does not change the weight w_t^p of each particle. A random gaussian noise $\eta(0, \sigma)$ is added to each component to account for imprecision in this transition, proportional to the distance δ navigated and the robot's sensitivity to errors, determined empirically. This sensitivity is modeled by coefficients α_{ij} which indicate the influence of a movement type i in the emergence of errors type j . So, $e_v = \eta(0, \alpha_{vv}\delta^v + \alpha_{\omega v}\delta^\omega)$ and $e_\omega = \eta(0, \alpha_{v\omega}\delta^v + \alpha_{\omega\omega}\delta^\omega)$.

$$x_{t+1}^p = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = g(x_t^p, u_t) = \begin{pmatrix} x + (v\Delta t + e_v) \cos(\theta + \omega\Delta t + e_\omega) \\ y + (v\Delta t + e_v) \sin(\theta + \omega\Delta t + e_\omega) \\ \theta + \omega\Delta t + e_\omega \end{pmatrix} \quad (7)$$

When a landmark which already has a position estimative is observed it is possible to determine the probability of each particle representing the correct localization and mapping states by comparing what the robot is observing with what he should observe according to the particle's localization and mapping hypothesis. This comparison is given by $y_{k,n}^{p,t}$ (2), the certainty inherent to it is quantified by the innovation covariance $S_{n,p}^t$ (3). The result is used to adjust the weight distribution in the particle set as shown in (8). If various landmarks are observed this step is conducted sequentially, and if no landmark is observed there is no update in the weights.

$$w_{t+1}^p = w_t^p \left(\frac{1}{2\pi^{N/2} |S_{n,p}^t|^{1/2}} \exp \left(-\frac{1}{2} (y_{k,n}^{p,t}) (S_{n,p}^t)^{-1} (y_{k,n}^{p,t})^T \right) \right) \quad (8)$$

3 Experimental Results

The solution proposed in this article was tested in a Pioneer 3AT (Fig. 3a) equipped with an odometry system for localization estimatives, a laser scanner used solely to build a metric map of the environment, and an omnidirectional vision system composed by a hyperbolic mirror and a camera vertically placed (Fig. 3b) on the rotation axis of the robot. The omnidirectional images collected were 640x480 grayscale and the landmark extraction and mapping were completed in approximately 2 s using a Pentium Core 2 Duo 2.0 GHz. Long-term navigation didn't affect this time significantly, indicating the algorithm's efficiency in dealing with an increasingly larger number of landmarks.

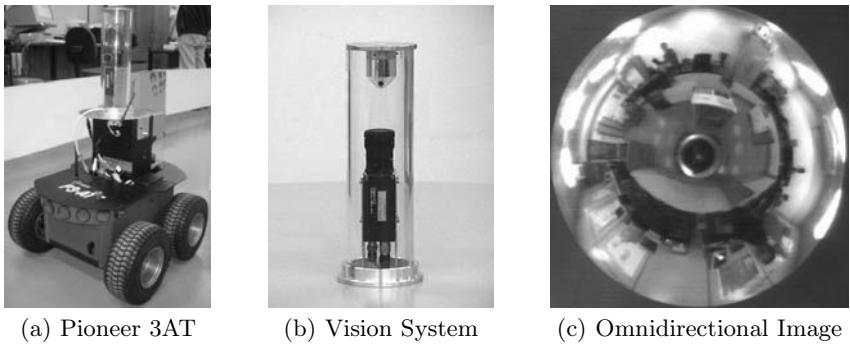


Fig. 3. Equipment used in the experimental tests

An environment of corridors and obstacles was constructed (Fig. 4a) and the robot navigated through it in trajectories of roughly 70m, with a maximum speed of 0.2 m/s. The navigation was done manually when the localization was not being corrected, because of accumulated errors that would prevent the robot from completing the path defined, and otherwise autonomously according to the control theory presented in [21]. Fig. 4b shows the result of localization and metric mapping without the SLAM algorithm, where the errors accumulated can be clearly perceived in both estimatives through repetition and misalignment of regions visited in different periods of time and the inability of the robot to close the final loop and return to its starting position.

Figure 4c shows the results of metric mapping and localization when the same path was done using the SLAM algorithm proposed (the lighter line indicates the previous trajectory without SLAM). The alignment between corridors is much more visible, indicating that the robot was capable of keeping the mapping consistency during all navigation and return precisely to its starting position. The sudden changes in localization are due to the errors accumulated in the intervals between images that spread the particles throughout a larger area. When the visual information is obtained the weight of the particles changes and it is possible that another, distant particle, is elected the most likely localization

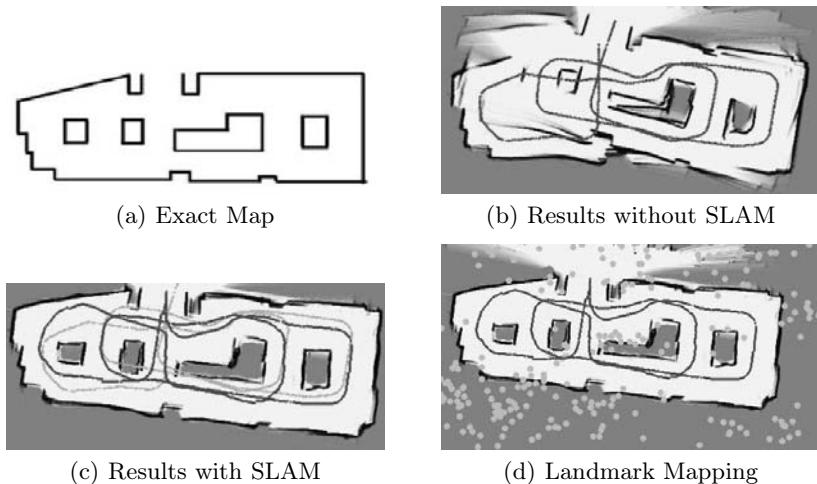


Fig. 4. Results obtained in experimental tests

and mapping hypothesis, creating a sudden change in the estimatives. The results of landmark mapping are shown in Fig. 4d, along with the metric mapping and localization with SLAM. Several groups, both inside the test course and outside (the robot could see above the walls) can be perceived and consistently matched with real objects on these locations, indicating structures it decided to use as landmarks according to their properties in the images.

4 Conclusion

We presented here a solution to the online SLAM problem that uses a single omnidirectional vision sensor to extract landmarks from natural structures that compose the environment around a robot. No premise was made regarding which structures should be used, so this solution can in theory be used equally in any kind of environment. The landmarks are extracted directly from the omnidirectional images and are compared with the ones extracted at earlier moments in search for matches. If a match is found, this information is used to correct odometric errors accumulated over time and decrease the position uncertainty of the landmark observed, solving simultaneously the localization and mapping problems. While the image is being processed the robot navigates blindly, thus increasing its localization uncertainty, and as soon as new information is available it is incorporated to the estimatives and this uncertainty decreases. The tests conducted show that an omnidirectional sensor is capable of maintaining the localization and mapping errors at a stable level during long-term navigation. Information from other sensors can be incorporated in a similar way, increasing the amount of data used by the robot to solve the problem of SLAM.

Acknowledgements. We would like to thank financial supporting agency FAPESP for this work, under grants #07/05293-2 and #07/07104-5.

References

1. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte carlo localization for mobile robots. In: International Conference on Robotics and Automation (May 1999)
2. Thrun, S.: Robotic mapping: A survey. In: Exploring Artificial Intelligence in the New Millennium. Morgan Kaufmann, San Francisco (2002)
3. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. Autonomous Robot Vehicles, 167–193 (1990)
4. Csorba, M.: Simultaneous Localization and Map Builing. PhD thesis, University of Oxford, Robotics Research Group (1997)
5. Bailey, T.: Mobile Robot Localization and Mapping in Extensive Outdoor Environments. PhD thesis, University of Sydney (2002)
6. Montemerlo, M.: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association. PhD thesis, Robotics Institute, Carnegie Mellon University (July 2003)
7. Williams, S.: Efficient Solutions to Autonomous Mapping and Navigation Problems. PhD thesis, University of Sydney (2001)
8. Nieto, J., Guivant, J., Nebot, E., Thrun, S.: Real time data association for fastslam. In: International Conference on Robotics and Automation (2003)
9. Leonard, J., Rickoski, R., Newman, P., Bosse, M.: Mapping partially observable features from multiple uncertain vantage points. International Journal of Robotics Research 21(10-11), 943–975 (2002)
10. Press, P., Austin, D.: Approaches to pole detection using ranged laser data. In: Proceedings of Australasian Conference on Robotics and Automation (2004)
11. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of IEEE, Conference on Computer Vision and Pattern Recognition (June 1994)
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 20, 91–110 (2004)
13. Zhu, Z.: Omnidirectional stereo vision. In: Proceedings of IEEE, 10th International Conference on Advanced Robotics (2001)
14. Gaspar, J.: Omnidirectional Vision for Mobile Robot Navigation. PhD thesis, Universidade Tecnica de Lisboa, Instituto Superior Tecnico (2003)
15. Se, S., Lowe, D., Little, J.: Vision-based mobile robot localization and mapping using scale-invariant features. In: International Conference on Robotics and Automation, May 2001, pp. 2051–2058 (2001)
16. Asmar, D., Zelek, J., Abdallah, S.: Tree trunks as landmarks for outdoor vision slam. In: Conference on Computer Vision and Pattern Recognition (June 2006)
17. Ledwich, L., Williams, S.: Reduced sift features for image retrieval and indoor localization. In: Australian Conference on Robotics and Automation (2004)
18. Rekleitis, I.: A particle filter tutorial for mobile robot localization. In: International Conference on Robotics and Automation (2003)
19. Welch, G., Bishop, G.: An introduction to the kalman filter. Technical report, University of North Carolina, Department of Computer Science (1995)
20. Murphy, K.: Bayesian map learning in dynamic environments. In: Neural Information Processing Systems (1999)
21. Davis, M., Vinter, R.: Stochastic Modelling and Control. Chapman and Hall, Boca Raton (1985)

Part XIV

**WORKSHOP: Neurocomputing and
Evolving Intelligence – NCEI 2008**

A Notable Swarm Approach to Evolve Neural Network for Classification in Data Mining

Satchidananda Dehuri¹, Bijan Bihari Mishra², and Sung-Bae Cho¹

¹ Soft Computing Laboratory, Department of Computer Science,
Yonsei University, 262 Seongsanno, Seodaemun-gu, Seoul 120-749, Korea
satchi.lapa@gmail.com, sbcho@cs.yonsei.ac.kr

² Department of Computer Science and Engineering, College of Engineering Bhubaneswar,
Bhubaneswar-751024, India
misrabijan@gmail.com

Abstract. This paper presents a novel and notable swarm approach to evolve an optimal set of weights and architecture of a neural network for classification in data mining. In a distributed environment the proposed approach generates randomly multiple architectures competing with each other while fine-tuning their architectural loopholes to generate an optimum model with maximum classification accuracy. Aiming at better generalization ability, we analyze the use of particle swarm optimization (PSO) to evolve an optimal architecture with high classification accuracy. Experiments performed on benchmark datasets show that the performance of the proposed approach has good classification accuracy and generalization ability. Further, a comparative performance of the proposed model with other competing models is given to show its effectiveness in terms of classification accuracy.

1 Introduction

Data mining [1] is an interdisciplinary field, whose core is at the intersection of machine learning, statistics, and databases. Data mining includes several tasks such as classification, association rule mining, clustering, regression, summarization etc. Each of these tasks can be regarded as a kind of problem to be solved by a data mining algorithm. Therefore, the first step in designing a data mining algorithm is to define which task the algorithm will address.

In this paper, we consider the problem of classification in data mining to validate our novel evolved neural network by particle swarm optimization. In general, a classifier partitions the feature space X into $C_i, i = 1, 2, \dots, M$ classes such that:

i) $C_i \neq \emptyset, i = 1, 2, \dots, M$, ii) $\bigcup_{i=1}^M C_i = X$ and iii) $C_j \cap C_k = \emptyset, j \neq k, j = 1(M), k = 1(M)$ (except for fuzzy classification domain) by constructing the hyper-planes or hyper-cubes. A hyper-plane can be written as: $d(x) = W \cdot X^T$, where $W = [w_1, w_2, \dots, w_N]$ and $X = [x_1, x_2, \dots, x_{N-1}, 1]$ are called the weight vector and augmented feature vector respectively. Now the problem is to classify an unknown sample based on the hyper-plane.

Various-benchmark studies indicate that, the success of artificial neural networks (ANN's) for classification depends on the architecture and training of the network (i.e. searching an optimal sets of weights). A large number of techniques exist with which ANN's can be trained. Most applications use the back propagation (BP) algorithm or other training algorithms in the feed forward ANN's. But all these training algorithms assume a fixed ANN's architecture. There have been many attempts in designing ANN's architectures automatically, such as various constructive and pruning algorithms [2].

Design of a near optimal ANN's architecture can be formulated as a search problem in the network architecture and weight space where each point represents a type of architecture. Given some performance criteria, e.g., high classification accuracy, lowest architectural complexity, etc., the performance level of all architectures forms a surface in the space. Although genetic algorithm (GA) [3] is a better candidate for searching optimal neural network architecture but it can face the problem like permutation, noisy fitness evaluation, etc. [2].

This paper presents a novel method of evolution of artificial neural network using PSO [4] that not only evolves the set of weights but also evolves network architecture of low complexity. Compared with the GA, the PSO algorithm possesses some attractive properties, such as memory and constructive cooperation between individuals, which can avoid the permutation problem.

2 Background and Related Work

2.1 Basics of PSO

Particle swarm optimization (PSO), introduced by Kennedy and Eberhart [4] is a stochastic optimization technique mimicking the behavior of a flock of birds.

The algorithm presented below uses the global best and local best mechanism of PSO. Let $f : R^n \rightarrow R$ be the fitness function and there are n particles, each with associated positions $x_i \in R^n$ and velocities $v_i \in R^n, i = 1(1)n$. Let \tilde{x}_i be the current best position of each particle and let \tilde{g} be the global best.

```

Algorithm_PSO()
{
    Initialize xi and vi, i = 1(1)n ;
     $\tilde{x}_i \leftarrow x_i$  ;  $\tilde{g} \leftarrow \arg \min_{x_i} f(x_i)$ , i = 1(1)n ;

    Repeat
        For each particle i = 1(1)n
            Create random vectors r1 and r2 ;
             $x_i \leftarrow x_i + v_i$  ;
             $v_i \leftarrow wv_i + c_1(r_1 \otimes (\tilde{x}_i - x_i)) + c_2(r_2 \otimes (\tilde{g} - x_i))$  ;
            If f(xi) < f( $\tilde{x}_i$ ) ;  $\tilde{x}_i \leftarrow x_i$  ;
            If f(xi) < f( $\tilde{g}$ ) ;  $\tilde{g} \leftarrow x_i$  ;
        End
    Until (stopping criterion met) ;
}

```

2.2 Related Work

X. Yao [2] seem to have been the first researcher for evolutionary neural network. He used a GA algorithm to evolve the architecture and weights of the network. Compared to GA, PSO algorithm possesses some attractive properties such as memory and constructive cooperation between individuals, so it has more chance to fly into the better solution areas more quickly and discover reasonable quality solution much faster. In PSO only few parameters are to be tuned. Unlike GA, the representations of the weights are easy and as there is no recombination and mutation operator, so there is a very negligible chance of facing the permutation problem. Further, as there is no selection operator in PSO, each individual in an original population has a corresponding partner in a new population. From the view of the diversity of population, this property is better than GA, so it can avoid the premature convergence and stagnation in GA to some extent.

Zhang et al. [5] has evolved PSO based ANN's and used in modeling product quality estimator for a fractionator of the hydrocracking unit in the oil refining industry. In their methodology, evolving ANN's architecture and weights are alternated and they used partial training for each individual architecture. The computation time of their proposed work is quite high due to the partial training.

Carvalho et al. [6] inspired by Zhang et al. works and then introduces the method PSO-PSO:WD based on the weight decay heuristic in the weight adjustment process in an attempt to obtain more generalization control.

Analyzing both algorithms we found that no one has given much emphasis on how many hidden layers required for the optimal architecture and nor even show about its optimality by experimental study. Further, neither given emphasis on how the particle is represented nor given much importance on numerical simulation to show the potentiality of their method. Further, as they are using PT algorithm for training so their method introduces overhead in computational time. Hence to cope with these problems we have motivated to develop a method, which can solve the said issues in addition to inheriting the basic characteristics.

3 PSO for Evolving Swarm Network

There are two major approaches to evolving the network architectures. One is the evolution of "pure" architecture with the randomly initialized connection weights, and then the connection weights will be trained after a near optimal architecture has been found. Another is the simultaneous evolution of both architectures and weights [2]. Unfortunately, the former brings out the noisy fitness evaluation that can mislead the evolution. Although the latter can effectively alleviate the noisy fitness evaluation problem, the nodes often become involved in a moving target problem because the solution space to be searched is too large. In general, it is the most significant reason why simultaneous optimization of the parameters of the entire networks can fail.

In order to solve preferably the above-mentioned problems, an approach is proposed where the number of hidden layers and number of neurons in the respective layer and set of weights are adaptively adjusted simultaneously.

In the proposed approach, both the architecture and the set of weights are encoded in particles and evolved simultaneously i.e. each particle represents a candidate solution of the architecture and weight space. In abstract view, Figure 1 shows the simultaneous evolution of architecture and weights.

- 1) Evaluate each particles based on the predefined criterion.
- 2) Find out the personal best (p_{best}) of each particle and global best (g_{best}) from the swarm.
- 3) Update particles velocity.
- 4) If the performance is satisfactory then stop, otherwise go to step 1.

Fig. 1. Abstract View of Evolving Swarm Network

In its current implementations, swarm approach to evolving neural network is used to evolve feed forward neural networks with a characteristic function. We defined two characteristics functions-one is for 2-class problem and another is for 3-class problems. In the case of 2-class problem the characteristic functions is defined as: $f(o) = 1$ if $o \leq \theta_1$, otherwise $f(o) = -1$, where θ_1 is a threshold value and o is the output produced by the output neuron. The positive value represent class 1 and the negative value represent class 2. Similarly, for three class problems the user has to choose two threshold values θ_1 and θ_2 based on that the characteristic function is defined as:

$$f(o) = \begin{cases} 1 & \text{if } o < \theta_1 \\ 0 & \text{if } \theta_1 \leq o < \theta_2 \\ -1 & \text{if } o \geq \theta_2 \end{cases}.$$

In the proposed method we restrict ourselves to put one neuron in the output layer instead of putting neurons based on the number of classes. However, this is not an inherent constraint. In fact, the proposed approach has minimal constraint on the type of artificial neural networks, which may be evolved. The feed forward ANNs do not have to be strictly layered or fully connected between adjacent layers. Their may also contains hidden nodes with different transfer functions.

Let us see how the proposed approach is representing the particles as well as evaluating the fitness of each particle.

3.1 Particle Representation

For representing the particles we have to set the protocols such as maximum number of hidden layers denoted as L_{max} , and maximum number of nodes for a particular hidden layer, denoted as N_{max} a priori. Based on these values the particle can be represented as a vector notation, $P_i = \langle P_{i1}, P_{i2}, \dots, P_{ib} \rangle$.

The first attribute P_{i1} of the particle represent the number of hidden layers in the architecture. The value of P_{i1} lies between 0 to L_{max} , i.e. $0 \leq P_{i1} \leq L_{max}$. The feature from P_{i2} to $P_{iL_{max}+1}$ tells about the number of neurons in the respective hidden layer. The next features stores the weights between input layer and 1st hidden layer and so on except the last feature of the particle P_{ib} . The last feature i.e. P_{ib} stores the weight values of bias unit.

3.2 Fitness Evaluation

The fitness of each particle of the proposed approach is solely determined by the classification accuracy based on the confusion matrix. Let the problem be a C class problem. Then the confusion matrix for the C class problem is defined as follows.

		Predicted			
		C_1	C_2	.	C_C
Actual	C_1	a_{11}	a_{12}	.	a_{1c}
	C_2	a_{21}	a_{22}	.	a_{2c}

	C_C	a_{c1}	a_{c2}	.	a_{cc}

The entries in the confusion matrix have the following meaning in the context of our study: a_{11} is the number of correct predictions that an instance is C_1 , a_{21} is the number of incorrect predictions that an instance is C_2 and so on. The classification accuracy is measured by the following formula with a consideration of imbalance-ness of classes.

$$\hat{f}_{CA} = \frac{1}{c} \left\langle \frac{a_{11}}{\sum_{j=1}^c a_{1j}} + \frac{a_{22}}{\sum_{j=1}^c a_{2j}} + \dots + \frac{a_{cc}}{\sum_{j=1}^c a_{cj}} \right\rangle$$

4 Experimental Study

The performance of the proposed model is evaluated using the four-benchmark databases taken from the UCI machine repository [8]. Table 1 presents a summary of the main features of each database that has been used in this study.

Table 1. Description of the features of the databases employed

Datasets	Patterns	Attributes	Classes	Patterns in Class1	Patterns in Class2	Patterns in Class3
IRIS	150	4	3	50	50	50
WINE	178	13	3	71	59	48
PIMA	768	8	2	500	268	-
BUPA	345	6	2	145	200	-

4.1 Parameters Setup

For simulation of the proposed models and to compare with other competing models like PSO-PSO:WD and PSO-PT, the different parameters considered for using the proposed method are presented in Table 2. For all simulations we have used g_{best} sociometry of PSO algorithm. Although this parameter area is quite restricted, and no systematic parameter optimization process has so far been attempted, however we set the parameters according to the suggested and widely used parameters in literatures [7] with a minor tuning such as $c_1 = 2.0476 \times w$ and $c_2 = 2.192 \times w$.

Table 2. Parameters required simulating the proposed method

Parameters of the Proposed Method			
PSO Parameters and Values		Neural Network Parameter and Values	
W	0.729844	L_{\max}	4
c_1	1.49445	N_{\max}	10
c_2	1.6	θ_1	0.5
Population Size	30	θ_2	0.833
Maximum Iterations			5000

4.2 Cross Validation, Results and Discussion

We have adopted the 2-fold cross-validation strategy with the intension of getting good classification accuracy. The dataset is randomly partitioned into two sets of equal size that are in turn used for building and testing the proposed model. While one part is used for building the proposed model the other part is used for testing the model, in a way that each one is used for opposite purpose. The percentage of correct classification for each dataset using the proposed model is presented in Table 3.

Table 3. Classification accuracy of proposed model

Dataset	Hit Percentage in Training Sets			Hit Percentage in Test Sets		
	Set1	Set2	Average	Set1	Set2	Average
IRIS	97.33	97.33	97.33	100.00	97.33	98.67
WINE	96.63	97.75	97.19	100.00	93.26	96.63
PIMA	73.44	75.26	74.35	85.42	82.81	84.16
BUPA	77.91	76.30	77.10	78.61	79.07	78.84

With the same protocol the average comparative performance of the proposed method with its rival is presented in Table 4.

Table 4. Average comparative performance

Dataset/ Methods	Average Hit Percentage in Training Set			Average Hit Percentage in Test Set		
	Proposed Model	PSO- PSO:WD	PSO-PT	Proposed Model	PSO- PSO:WD	PSO-PT
IRIS	97.33	96.00	96.65	98.67	96.66	98.65
WINE	97.19	87.65	92.15	96.63	79.20	96.10
PIMA	77.10	60.25	72.75	78.84	58.55	76.25
BUPA	74.35	74.75	76.30	84.12	72.50	78.35

From the results presented in Table 4 we can note that for all the datasets except BUPA the proposed method obtained better average classification accuracy in both training set and test set compared to the method PSO-PSO: WD and PSO-PT. In the case of BUPA even though the training performance of the proposed method is close or nearly better than other two methods but during testing the performance is quite better than other two methods.

The number of hidden layers and the number of neurons required in each hidden layer for the different dataset to generate the proposed architecture is presented in Figure 2.

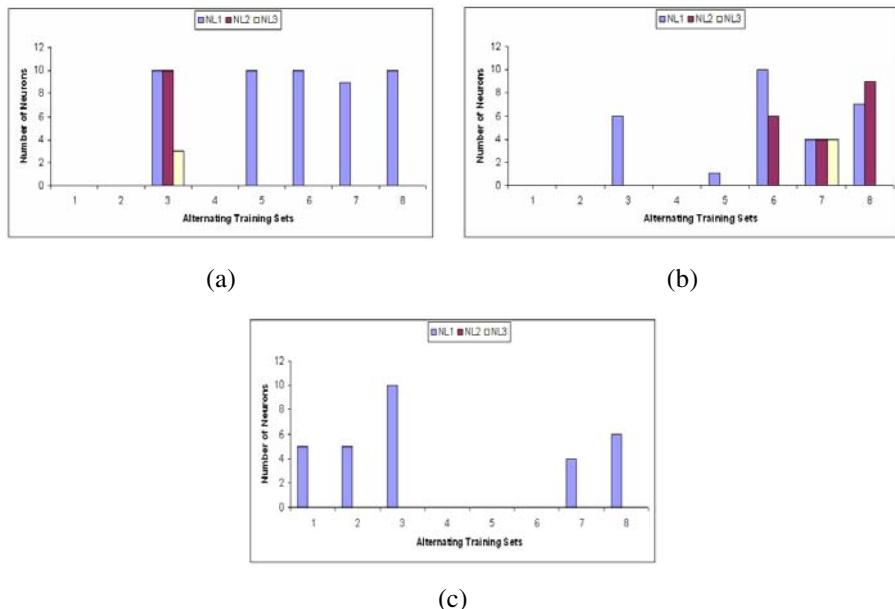


Fig. 2. Architectural Complexity of (a) Proposed Method (b) PSO-PSO:WD and (c)PSO-PT

In view of the architectural complexity, on the whole the proposed approach is complex than other two models, whereas if we compare the classification accuracy the proposed approach is more promising than the other two approaches. Therefore, if one view from a multi-objective perspective no solution is better than others and it is also very difficult to optimize both these objectives simultaneously. These two objectives are very often conflicting to each other.

5 Conclusions and Future Research

In this paper, we have proposed a novel swarm approach to evolve neural network aiming to optimize simultaneously both architecture and set of weights. The proposed model is evaluated using the benchmark datasets considering the task of classification in data mining. Further, we compared our model with two other competing models such as PSO-PSO:WD and PSO-PT. Experimental studies demonstrated that the proposed model is quite superior to other two models in term of classification accuracy whereas in terms of architectural complexity our model introduces little overhead but can be tolerable by the data mining designer and ultimately decision maker. Future work should consist of experiments with other large datasets as well as the fine-tuning

of the parameters used in the proposed method. It would be interesting to optimize the architectural complexity and classification accuracy considering as a multi-objective problem by PSO and generate the Pareto front.

Acknowledgments. The authors would like to thank the financial support of BK21 research program on Next Generation Mobile Software at Yonsei University, SOUTH KOREA.

References

1. Ghosh, A., Dehuri, S., Ghosh, S. (eds.): Multi-objective Evolutionary Algorithms for Knowledge Discovery from Databases. Springer, Heidelberg (2008)
2. Yao, X., Liu, Y.: A New Evolutionary System for Evolving Artificial Neural Networks. *IEEE Transactions on Neural Networks* 8(3), 694–713 (1997)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimisation and Machine Learning. Addison-Wesley Pub. Co., Reading (1989)
4. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimisation. In: Proc. IEEE International Conference on Neural Networks, pp. 39–43. IEEE Service Center, Piscataway (1995)
5. Zhang, C., Shao, H.: An ANN's Evolved by a New Evolutionary System and its Application. In: Proc. of 39th IEEE Conference on Decision and Control, Sydney, pp. 3562–3563 (2000)
6. Carvalho, M., Ludermir, T.B.: Particle Swarm Optimisation of Neural Network Architectures and Weights. In: Proc. of 7th International Conference on Hybrid Intelligent Systems, pp. 336–339. IEEE Computer Society Press, Los Alamitos (2007)
7. Clerc, M., Kennedy, J.: The Particle Swarm-Explosion, Stability, and Convergence in a Multi-dimensional Complex Space. *IEEE Transactions on Evolutionary Computation* 6(1), 58–73 (2002)
8. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases,
<http://www.ics.uci.edu/~mlearn/MLRepository.html>

FPGA Implementation of an Evolving Spiking Neural Network

Alan Zuppich and Snjezana Soltic

School of Electrical Engineering and Trades

Manukau Institute of Technology, Auckland, New Zealand

alan.zuppich@easel.co.nz, snjezana.soltic@manukau.ac.nz

Abstract. This research presents a Field Programmable Gate Array (FPGA) implementation of a taste recognition model. The model is based on simple integrate and fire neurons and facilitates an on-line learning. The whole system, including the hardware required to build (evolve) the network was hosted on one FPGA chip. The implementation used 45% of the logic elements, 76% of the memory, and 23% of the dedicated multiplier slices of the chip. FPGA size was sufficient for 64 neurons with up to 64 synapses each (a total of 4096 synapses). The proposed FPGA implementation was successfully applied to a classification problem of taste recognition and the FPGA implementation was at least 10 times faster when evolving the network and 74 times faster during the classification than the software simulations executed by a stand-alone PC.

1 Introduction

Field Programmable Gate Array (FPGA) integrated circuits offer re-configurable platforms for construction of highly parallel hardware logic circuits. This makes them a logical choice for building the parallel processing systems such as artificial neural networks (ANN). However, FPGA implementations of ANN are linked with finite FPGA resources and designers are required to make trade-offs between resources, calculation speed and approximation accuracy [1]. The main advantage of building the FPGA-based implementations of ANN is the speed of those implementations compared to the speeds achieved in simulation [2]. Furthermore, FPGA implementations are associated with low prototyping cost [3]. Also, FPGA implementations are suitable for embedded design [4]. Many implementations lack on-chip learning, limiting their ability to adapt readily to new data. This is especially so for spiking neural networks (SNNs). Examples of SNN implementations that do exhibit on-chip learning are in [5] and [6], where the networks were evolved using a genetic algorithm run on a microprocessor. Spiking neural networks presently account for only a minority of the connectionist system implementations on FPGA.

This research presents an FPGA implementation of an evolving spiking neural network proposed in [7] for taste recognition. The model is based on simple integrate and fire neurons with rank order coded (ROC) inputs and facilitates

an on-line learning [8]. The whole taste recognition system, including the hardware required to build (evolve) the network was hosted on the same chip, an Altera Cyclone II EP2C35F672C6 FPGA [9]. A NIOS microprocessor core was instantiated alongside the neural network on the FPGA to interface the network to a host computer for inputting data vectors. To our best knowledge, this is the first FPGA implementation of a taste recognition system, the spiking neurons and learning.

To start with, a summary of the evolving spiking neural network is provided in Section 2. Section 3 gives the implementation details, including block diagrams of the network's modules. The implementation was tested on two real-world datasets and the results are presented in Section 4. Conclusions are given in Section 5.

2 Evolving Spiking Neural Network

The taste recognition system proposed in [7] comprises two modules: a GRF module with m equally spaced Gaussian receptive fields which perform population coding of input data vectors, and an evolving spiking neural network (ESNN) in which L1s represent taste receptors whose values are encoded using ROC and L2s are simple integrate-and-fire units sensitive to the order of the incoming spikes (Fig. 1). The L2 layer is initially empty and all L2 neurons are created during the evolving process. The number of L2 neurons differs from one ensemble to another, i.e. different tastants are learnt by a different number of L2 neurons ($G_l(C_k)$). For a given input vector, the membrane potential PSP_i of an output $L2_i$ is the sum of the postsynaptic potentials PSP_{ji} :

$$PSP_i = \sum_{j=1}^n PSP_{ji} = \sum_{j=1}^n (mod^{order_j} \times w_{ji}) \quad (1)$$

where n is the number of L1 neurons, $mod \in [0, 1]$ is a modulation factor, $order \in [0, n - 1]$, $order_j$ is the firing order of $L1_j$, w_{ji} is the synaptic weight between $L1_j$ and $L2_i$. The w_{ji} values are found during the evolving of the network by using

$$w_{ji} = mod^{order_j} \quad (2)$$

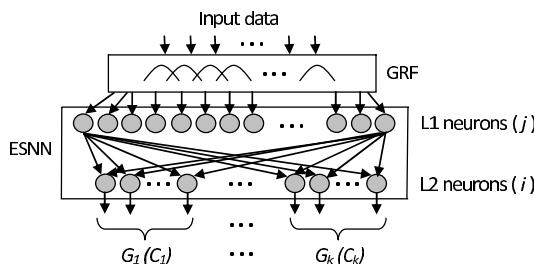


Fig. 1. Block diagram of the evolving spiking neural network

with the same convention as in (1). The $L2$ neurons exhibit a dynamic threshold where the threshold value of an $L2_i$ neuron is set to a proportion of its maximum membrane potential value ($PSP_{i_{max}}$).

$$PSP_{\theta_i} = c \times PSP_{i_{max}} . \quad (3)$$

During evolving of a network, a new $L2_n$ neuron is created for each training vector v_n . The neuron is trained using (2) and (3). Then $L2_n$ is compared against any other pre-existing neuron of the same class. If the Euclidean distance between their weight matrices is less than distance S_θ the neuron is considered similar to the existing neuron. If the new neuron is found similar to an existing neuron $L2_o$, the new neuron is aggregated into $L2_o$. The weights of $L2_o$ are updated by the arithmetic average of its old weights and weights of $L2_n$. The PSP_{θ_o} of $L2_o$ is updated likewise, then $L2_n$ is discarded. Note that only neurons representing the class of the input data vector are modified. For more details about this network, its software implementation and how the network was used for taste recognition see [7].

3 Implementation

The implementation (Fig. 2) comprises three functional modules, the population encoder, the temporal rank order coder and a layer of $L2_n$ neurons. Furthermore, a NIOS microprocessor core is instantiated alongside the neural network on the FPGA to interface the network to a remote host computer for inputting vectors, and to allow remote access to neuron variables and network structure metrics. The whole system was implemented around look-up-tables, minimising the need for multipliers and logic elements. Furthermore, a virtual neural based architecture was used where only one neuron with one synapse was instantiated and all other neurons used the same resources. While ensuring that resource consumption was as low as possible was the main challenge we made sure that numeric fidelity was not compromised by keeping quantization errors to a minimum. Data was represented with parallel 32-bit signed fixed-point words. Synaptic weights and normalised sensory data vectors may take values $\in [0, 1]$ hence these are

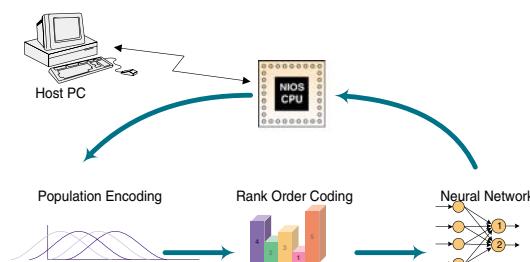


Fig. 2. Block diagram of the FPGA implementation

represented with Q1.30 (1 integer bit and 30 fractional bits). Post-synaptic potential and thresholds may take values $\in [0, n]$ where n is the number of synapses and limited to 64 for this implementation, so these values are represented with Q7.24.

3.1 Population Encoding

Data vectors first undergo population encoding of data features. Here vectors of f features are expanded to $n = f \times m$ temporally coded values for input to synapses of L2 by m overlapping Gaussian receptive fields. A finite state machine (FSM) steps through reading each feature of the input vector, and for each feature the FSM cycles through subtracting field centre values and presenting the difference to a look-up table (Gaussian curve LUT) (Fig. 3). The table contains 8192 pre-calculated 22-bit Gaussian curve points with step size 1.2E-4 and maximum error 0.12%. A zero is output for any feature further than 3 standard deviations from centre. Consequently L1 neurons conveying these values do not spike.

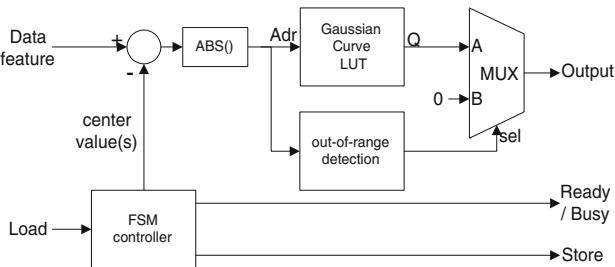


Fig. 3. LUT-based Population Encoder block diagram

3.2 Temporal Rank Order Coding

The data from the population encoder must be presented to the neural network in temporal order. For this a Rank Coder (Fig. 4) based on a FSM performs a top-down (from most-significant bit) search algorithm over all data values in

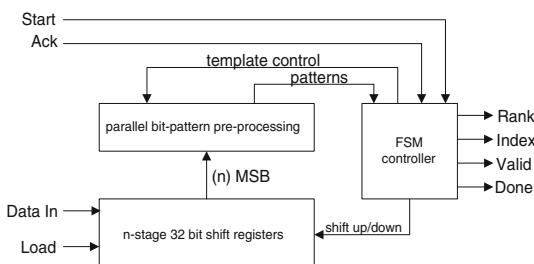


Fig. 4. Rank Order Coder block diagram

parallel. The population-encoded vectors of n features are loaded into n stages of shift registers, one per feature. By simultaneous rotation of all register contents any bit within the features can be set at the top of the registers. From here n -bit wide priority encoders and comparators summarise the feature bit patterns for the FSM. The FSM uses a template or system of logic gates to restrict analysis to selected register columns as it searches downwards for the highest value. Features with higher values represent earlier spiking times, the highest value being ranked 0 ($order=0$). This design determines spike order rapidly, identifying the first values often within a few clock cycles. Furthermore, the design permits neural processing to occur in parallel with ranking.

3.3 Evolving Spiking Neural Network

The neuron operates within a discrete time domain with intervals defined by $order$, where $order \in (0, n - 1)$, n is the number of synaptic inputs (L1 neurons). Given that the number of multipliers in an implementation must be kept to minimum, the multipliers required for the power calculations in (1) and (2) were avoided by storing the mod^{order} values into a look-up table (Mod^{rank} LUT)(Fig. 5). As a result, the calculation of PSP values was reduced from the power-multiply-and-add operations to one multiply-and-add operation per synapse. The PSP and PSP_θ values are held in the on-chip memory (PSP memory and Threshold memory blocks). At each time interval the rank coder presents the neurons with a pair of values indicating the current rank and the current synaptic index j . During classification ranking terminates once first spikes occur.

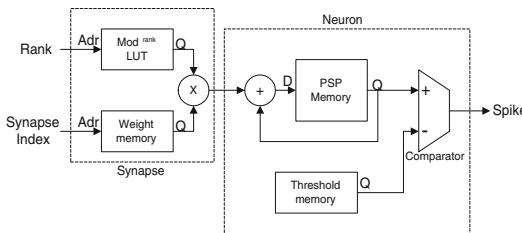


Fig. 5. Artificial spiking neuron block diagram (learning circuits not shown)

4 Results

The performance of the FPGA implementation was compared to the performance of the software implementation of the same network architecture. The software simulations were run on a 2 GHz Intel Core 2 with 1 GB RAM running Windows XP [7], whereas the operating frequency of the FPGA model was 50 MHz ($t_{CLK} = 20$ ns). Parameters mirrored those in [7] with most remaining fixed throughout testing ($mod = 0.95$, $c = 0.33$, and $\alpha = 0.333$). All datasets contained $f=7$ features per vector. Both m and S_θ parameters can be altered by simply recompiling VHDL source.

4.1 Classification Accuracy

With $m=6$ and $S_\theta=1.3$ average classification accuracy of 100% was achieved during repeated tests on a well-separated binary dataset of 26 samples. Tests on 20 samples from two less well-separated classes saw average accuracy of 91% for $m=6$ increase up to 100% average after a change to $m=8$. Average accuracy for 40 samples from a 4 class dataset with limited separation was 87% when $m=6$ and $S_\theta=0.5$. Average accuracy of 100% was again achieved on a test of a 10 times larger binary set with $m=8$ and $S_\theta=1.3$. These results are comparable to results from earlier computer simulations.

4.2 FPGA Resource Usage

The implementation used 76% of the EP2C35F672 FPGA memory, 45% of the logic elements and 23% of the dedicated multipliers (Table 1). Half of all memory space was used by the population encoding module (POP) and its LUTs. It is interesting to see that the processor itself uses only 11% of the memory. On the other hand, the on-chip learning circuit uses 14% of the multipliers and 4% of the logic elements. The modules were designed to support a temporally multiplexed network of at least 64 neurons and a total of 4096 synapses. While the network was kept within these dimensions it was possible to contain all synaptic weights, post-synaptic potentials and threshold values to within the FPGA internal memory.

Table 1. Available FPGA resources and their usage by the proposed implementation

Resources	Available	NIOS	POP	Ranking	Neuron	Learning	Usage%
Logic elements	33216	7000	252	5000	1200	1460	45
Memory (kbits)	483	52	180	0	138	0	76
9×9 Multipliers	70	4	0	0	2	10	23

4.3 Training and Classifying Speeds

Table 2 shows times required for the FPGA-based and software implementations to train (evolve) the network on one data sample and to classify one sample. Note that both implementations are faster during the classification because there is no need to build the neurons.

The population encoder outputs temporal coded data every t_{CLK} , encoding a sample feature in $m \times t_{CLK}$ where m is the number of Gaussian receptive fields (e.g. 1.6ns per feature when $m=8$, $1.12\mu s$ per data vector). This time is insignificant compared to the $3\mu s$ ($t_d = 3\mu s$) or more the microprocessor requires for sourcing the feature data. A sample vector of f features encodes in $f \times (m \times t_{CLK} + t_d)$ or $22\mu s$.

The rank coder performs temporal rank order determination of 56 32-bit values ($n=56$) in $6\mu s$ to $16\mu s$ (time somewhat dependant on the distribution of values) with first results available in as little as 60ns.

Training requires between $40\mu s$ and $80\mu s$ per sample vector (T Total). Creation of new neurons takes 160ns per synapse, or $9\mu s$ for $n=56$. To this is added $6\mu s$ to $16\mu s$ of ranking overhead and $25\mu s$ to $60\mu s$ for similarity testing and aggregation (times for a network of 6 neurons) to arrive at the per-sample time. The details and timing analysis of the on-chip learning can be found in [10].

Classification of a sample in a network of 6 neurons each of 56 synapses requires around $6\mu s$ (C Total), being $2\mu s$ for neural processing and the balance in population encoding and interleaved temporal ranking. The figure of $2\mu s$ arises from $n_a \times l \times s \times t_{CLK}$, where l is the total number of L2 neurons, n_a is the number of synapses activated prior to spiking, and $s=5.2$ is an approximation of the average of synaptic activation times of $5t_{CLK}$ without spiking occurring and $6t_{CLK}$ when spiking occurs. With 6 neurons each synaptic activation interval lasts 640ns or $(l \times s \times t_{CLK})$.

Table 2. Training and classification times for software and FPGA implementations $n=56$

	POP	Rank	1^{st} order	Training	T Total	Classification	C Total
Software (μs)	123	320	320	600	1040	105	445
FPGA (μs)	$1.12 + (f \times t_d)$	6-16	0.06-3	30-70	40-80	2	6
	$1.12 + (f \times t_d)$	2	0.06-3				

5 Conclusions

An evolving integrate and fire spiking neural network with structure adaptation was successfully implemented on a FPGA. A fast non-destructive learning algorithm is performed on-chip. The implementation performs well both in terms of classification accuracy and speed of operation. Training requires between $40\mu s$ and $80\mu s$ per sample vector and classification of a sample in a network of 6 neurons requires as little as $6\mu s$.

The processing time required by the LUT based population encoder is insignificant, however resolving ability in a LUT based coder relates directly to the size of the LUT and any lack of resolving ability in the population encoding reduces classification accuracy. To ensure desired accuracy the LUT is set to use 37% of the FPGA memory. The rank order coder uses 15% of the FPGA logic but the resultant design performs rapid interleaved ranking and neural processing with very short delays to first synaptic activations.

Acknowledgments. The authors wish to thank André C. P. L. F. de Carvalho for access to the water and wine datasets.

References

1. Merchant, S., Peterson, G.D., Ki Park, S., Kong, S.G.: FPGA Implementation of Evolvable Block-based Neural Networks. In: IEEE Congress on Evolutionary Computation, pp. 3129–3136. IEEE Press, Los Alamitos (2006)

2. Himavathi, S., Anitha, D., Muthuramalingam, A.: Feedforward Neural Network Implementation in FPGA Using Layer Multiplexing for Effective Resource Utilization. *IEEE Tran. on Neural Networks* 18(3), 880–888 (2007)
3. Damak, A., Krid, M., Sellami Masmoudi, D., Derbel, N.: FPGA implementation of programmable pulse mode neural network with on-chip learning. In: International Conference on Design and Test of Integrated Systems in Nanoscale Technology, pp. 159–164. IEEE Press, Los Alamitos (2006)
4. Lee, Y., Ko, S.-B.: FPGA Implementation of a Face Detector Using Neural Networks. In: IEEE Conference on Electrical and Computer Engineering, pp. 1914–1917. IEEE Press, Los Alamitos (2006)
5. Low, K.-S., Krishnan, V., Zhuang, H., Yau, W.-Y.: On-chip genetic algorithm optimized pulse based RBF neural network for unsupervised clustering problem. In: Jiao, L., Wang, L., Gao, X.-b., Liu, J., Wu, F. (eds.) ICNC 2006. LNCS, vol. 4222, pp. 851–860. Springer, Heidelberg (2006)
6. Roggen, D., Hofmann, S., Thoma, Y., Floreano, D.: Hardware spiking neural network with run-time reconfigurable connectivity in an autonomous robot. In: NASA/DoD Conference on Evolvable Hardware, pp. 189–198. IEEE Press, Los Alamitos (2003)
7. Soltic, S., Wysoski, S.G., Kasabov, N.K.: Evolving spiking neural networks for taste recognition. In: 2008 International Joint Conference on Neural Networks, pp. 2092–2098. IEEE Press, Los Alamitos (2008)
8. Wysoski, S.G., Benuskova, L., Kasabov, N.: On-Line Learning with Structural Adaptation in a Network of Spiking Neurons for Visual Pattern Recognition. In: Kollias, S.D., Staftlopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 61–70. Springer, Heidelberg (2006)
9. Cyclone II Device handbook, <http://www.altera.com/literature/lit-cyc2.jsp>
10. Soltic, S., Zuppich, A.: Comparison between software and FPGA implementation of ESNN in terms of processing speed. In: 15th Electronics New Zealand Conference, pp. 56–61. Electronics New Zealand Incorporated (2008)

HyFIS-Yager-gDIC: A Self-organizing Hybrid Neural Fuzzy Inference System Realizing Yager Inference

Sau Wai Tung¹, Chai Quek^{1,*}, and Cuntai Guan²

¹ Center for Computational Intelligence, Sch. of Comp. Engineering,
Nanyang Technological University, Singapore
ashcquek@ntu.edu.sg

² Institute for Infocomm Research, A*Star, Singapore

Abstract. The Hybrid neural Fuzzy Inference System (HyFIS) is a five layers adaptive neural fuzzy system for building and optimizing fuzzy models. In this paper, the fuzzy Yager inference scheme, which accounts for a firm and intuitive logical framework that emulates the human reasoning and decision-making mechanism, is integrated into the HyFIS network. In addition, a self-organizing gaussian Discrete Incremental Clustering (gDIC) technique is used to form the fuzzy sets in the fuzzification phase. This clustering technique is no longer limited by the need to have prior knowledge about the number of clusters needed in each input and output dimensions. The proposed self-organizing Hybrid neural Fuzzy Inference System based on Yager inference (HyFIS-Yager-gDIC) is benchmarked on two case studies to demonstrate its superiority as an effective neuro-fuzzy modelling technique.

1 Introduction

Neural fuzzy networks are hybrid systems that capitalize on the functionalities of fuzzy systems and neural networks. The main advantage of a neural fuzzy network is its ability to model a problem using vague information by means of a set of linguistic IF-THEN fuzzy rules while possessing learning abilities to self-adjust the parameters of the fuzzy rules. This is an improved tool over the individual systems of fuzzy logic and neural networks because the drawbacks of the individual systems such as the lack of learning abilities in a fuzzy logic system and the black-box phenomenon in a neural network are resolved.

The Hybrid neural Fuzzy Inference System (HyFIS) [4] is a five layers adaptive neural fuzzy network that is used to combine numerical and linguistic information into a common framework. It adopts a two phase learning scheme. In the first phase, a fuzzy technique of Wang and Mendel [9] is used to obtain the initial fuzzy rulebase and the initial structure of the neural fuzzy system. In the second phase, a parameter learning technique using a gradient descending approach is used to tune the memberships of the input and output dimensions.

* Corresponding author.

Chen and Chiang [2] proposed an intelligent control system, using the HyFIS architecture as the controller in the framework. Pasha et. al. [7] proposed an evolvable neural fuzzy network, which adopts the structure of HyFIS, for adaptive network anomaly detection. Both the frameworks mimic the human-like decision-making procedures, but they lack the logic framework present in the human reasoning and decision-making mechanism.

This paper presents the self-organizing Hybrid neural Fuzzy Inference System based on Yager inference (HyFIS-Yager-gDIC), which emulates the human reasoning and decision-making mechanism. The Discrete Incremental Clustering (DIC) [8] technique was first proposed for trapezoidal membership functions to overcome the need for prior knowledge on the number of clusters in partition-based clustering techniques. The use of the gaussian Discrete Incremental Clustering (gDIC) [6] technique for gaussian membership functions in the initialization phase of the HyFIS network allows robust handling of noisy data, while preserving the dynamism of partition-based clustering techniques. In addition, the realization of the fuzzy Yager inference scheme [3] in the HyFIS network offers a firm and intuitive logical framework. The main advantage of the Yager inference is that when the input perfectly matches the antecedents, the output will be the exact consequents of the fuzzy rule.

The rest of the paper is organized as follows: the gDIC technique is introduced in Sect. 2; the HyFIS network is briefly described in Sect. 3; the operations and learning process of the proposed HyFIS-Yager-gDIC network are presented in Sect. 4; the applications and results of HyFIS-Yager-gDIC on two benchmark case studies are described in Sect. 5; and Sect. 6 concludes the paper.

2 Gaussian Discrete Incremental Clustering

The gDIC [6] technique processes raw numerical training data and automatically determines the number of clusters in each of the input and output dimensions. Prior knowledge of the number of clusters is not required. This helps to maintain a consistent representation of the fuzzy sets on a local basis, such that the number of labels in the input and output dimensions need not be the same. In addition, the gDIC technique is able to determine the centroids and the widths of the gaussian fuzzy labels in the dimensions that best represent the data, which is not the case in the method proposed by Wang and Mendel where the fuzzy labels are evenly spaced in each of the dimensions with a fixed width.

3 HyFIS: Hybrid Neural Fuzzy Inference System

HyFIS [4] is a multilayer neural network based fuzzy system. The architecture of HyFIS is shown in Fig. 1 and the network consists of five layers of nodes. Layer 1 consists of the input nodes; layer 2 consists of the antecedent nodes; layer 3 is the rule nodes; layer 4 is the consequent nodes; and layer 5 consists of the output nodes. Each input node $IV_i, i \in \{1 \dots n_1\}$ in layer 1 takes in a single input value and the input vector is represented as $x = \{x_1, \dots, x_i, \dots, x_{n_1}\}$.

Each output node $OV_m, m \in \{1 \dots n_5\}$ in layer 5 produces a single output value and the output vector is represented as $y = \{y_1, \dots, y_m, \dots, y_{n_5}\}$. In addition, the vector $d = \{d_1, \dots, d_m, \dots, d_{n_5}\}$ represents the desired output during the parameter learning phase. For each input variable IV_i , it will consist of J_i number of fuzzy labels. Each antecedent node will be represented as $IL_{i,j_i}, j_i \in \{1 \dots J_i\}$ and the total number of nodes in layer 2 is $n_2 = \sum_i J_i$. For each output variable OV_m , it will consist of L_m number of fuzzy labels. Each consequent node will be represented as $OL_{l_m,m}, l_m \in \{1 \dots L_m\}$ and the total number of nodes in layer 4 is $n_4 = \sum_m L_m$. Layer 3 represents the rule nodes R_k where $k \in \{1 \dots n_3\}$. Fuzzy rules with certainty factors, represented as weights in the synapses connecting layers 3 and 4 of the network, are used in HyFIS. The training parameters are the centres and the widths of the gaussian membership functions present in layers 2 and 4 defined as

$$\mu(x) = e^{-((x-c)^2/\sigma^2)} \quad (1)$$

where c is the centre of the function and σ is the width of the function.

HyFIS adopts a two phase learning scheme. The first phase is the structure learning phase for knowledge acquisition. A fuzzy technique by Wang and Mendel [9] is used to derive an initial rulebase of the network. Prior knowledge on the number of clusters is required to produce an evenly spaced grid partition in the input and output dimensions. The second phase is the parameter learning phase. The input vector is propagated forward in the network following the Compositional Rule of Inference (CRI) scheme [10] and parameter tuning is performed by backpropagating the error using a gradient descending approach.

4 HyFIS-Yager-gDIC

The proposed HyFIS-Yager-gDIC has the same architecture as HyFIS as shown in Fig. 1. Similarly, it adopts a two phase learning scheme. In phase one, the self-organizing gDIC [6] is used to perform cluster partitioning in the input and output dimensions. Initial fuzzy rules are then formed using a method similar to Wang and Mendel [9]. In phase two, a supervised learning scheme based on a gradient descent learning is used to optimally tune the parameters of the membership functions. The two phase learning scheme allows HyFIS-Yager-gDIC to automatically formulate the initial fuzzy rules from raw numerical training data and subsequently evolve its structure through learning. In addition, the fuzzy Yager inference scheme [3] is integrated into HyFIS-Yager-gDIC, because it provides a firm and logical framework that closely emulates the human reasoning process. The key feature of adopting the fuzzy Yager inference scheme is its ability to produce output that are the exact consequents of the fuzzy rule when the input perfectly matches the antecedents of the fuzzy rule.

4.1 Implementation of Fuzzy Yager Inference Scheme

The generic operations of the proposed HyFIS-Yager-gDIC are defined as follows: the forward aggregation and activation functions of each layer $I \in \{1 \dots 5\}$

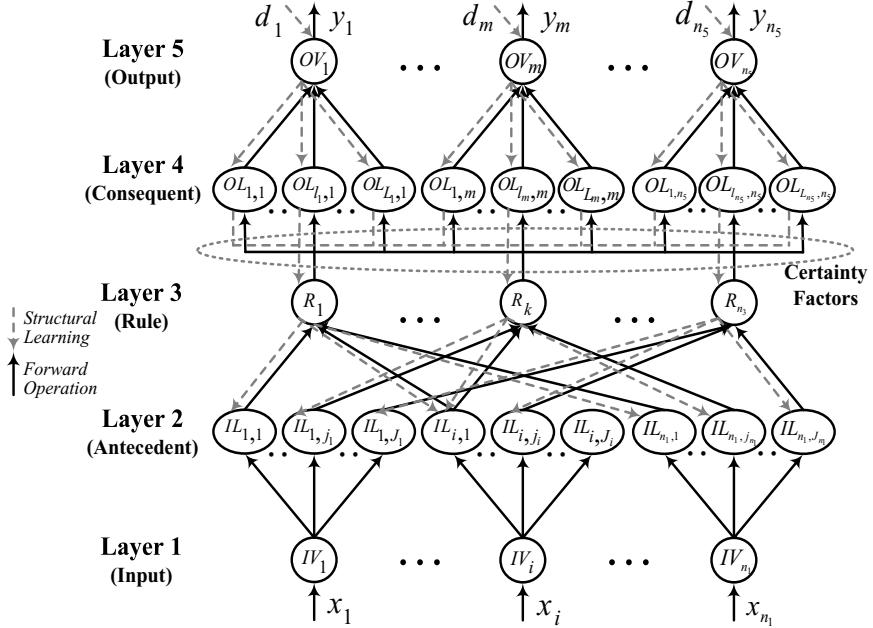


Fig. 1. Structure of HyFIS/HyFIS-Yager-gDIC

are denoted as $f^{(I)}$ and $a^{(I)}$ respectively. The aggregated input value to an arbitrary node is denoted as NET and the output of an arbitrary node is denoted as Z .

Layer 1

$$NET_{IV_i} = f^{(1)}(x_i) = x_i ; \quad (2)$$

$$Z_{IV_i} = a^{(1)}(NET_{IV_i}) = x_i . \quad (3)$$

Layer 2

$$NET_{IL_{i,j_i}} = f^{(2)}(Z_{IV_i}) = x_i ; \quad (4)$$

$$Z_{IL_{i,j_i}} = a^{(2)}(NET_{IL_{i,j_i}}) = 1 - \mu_{IL_{i,j_i}}(x_i) . \quad (5)$$

Layer 3

$$NET_{R_k} = f^{(3)}(Z_{IL_{(1,j_1)_k}}, \dots, Z_{IL_{(i,j_i)_k}}, \dots, Z_{IL_{(n_1,j_{n_1})_k}}) \quad (6)$$

$$= (Z_{IL_{(1,j_1)_k}}, \dots, Z_{IL_{(i,j_i)_k}}, \dots, Z_{IL_{(n_1,j_{n_1})_k}}) ;$$

$$Z_{R_k} = a^{(3)}(NET_{R_k}) = 1 - \max_i(Z_{IL_{(1,j_1)_k}}, \dots, Z_{IL_{(i,j_i)_k}}, \dots, Z_{IL_{(n_1,j_{n_1})_k}}) \quad (7)$$

where $Z_{IL_{(i,j_i)_k}}$ is the output of IL_{i,j_i} that is connected to R_k .

Layer 4

$$\begin{aligned} NET_{OL_{l_m,m}} &= f^{(4)}(Z_{R_1}^{(l_m,m)}, \dots, Z_{R_k}^{(l_m,m)}, \dots, Z_{R_K}^{(l_m,m)}) \\ &= (Z_{R_1}^{(l_m,m)}, \dots, Z_{R_k}^{(l_m,m)}, \dots, Z_{R_K}^{(l_m,m)}) ; \end{aligned} \quad (8)$$

$$Z_{OL_{l_m,m}} = a^{(4)}(NET_{OL_{l_m,m}}) = 1 - \max_k(Z_{R_1}^{(l_m,m)}, \dots, Z_{R_k}^{(l_m,m)}, \dots, Z_{R_K}^{(l_m,m)}) \quad (9)$$

where $Z_{R_k}^{(l_m,m)}$ is the output of R_k that is connected to $OL_{l_m,m}$.

Layer 5

$$\begin{aligned} NET_{OV_m} &= f^{(5)}(Z_{OL_{1,m}}, \dots, Z_{OL_{l_m,m}}, \dots, Z_{OL_{L_m,m}}) \\ &= (Z_{OL_{1,m}}, \dots, Z_{OL_{l_m,m}}, \dots, Z_{OL_{L_m,m}}) ; \end{aligned} \quad (10)$$

$$Z_{OV_m} = a^{(5)}(NET_{OV_m}) = \frac{\sum_{l_m} (1 - Z_{OL_{l_m,m}}) c_{l_m,m} / (1 + \sigma_{l_m,m})}{\sum_{l_m} (1 - Z_{OL_{l_m,m}}) / (1 + \sigma_{l_m,m})} \quad (11)$$

where $c_{l_m,m}$ and $\sigma_{l_m,m}$ are the centre and width of $\mu_{OL_{l_m,m}}$ respectively.

4.2 Parameter Learning Phase

The supervised learning algorithm of HyFIS-Yager-gDIC is based on a gradient descent approach to minimize the error function

$$E = \frac{1}{2} \sum_X \sum_m (d_m - y_m)^2 \quad (12)$$

where X is the training data set.

5 Experimental Results

HyFIS-Yager-gDIC is applied to two well-examined benchmark examples in this section. The strength and potential of the proposed HyFIS-Yager-gDIC is revealed through a comparison with the HyFIS model.

Prediction of Chaotic Dynamic System: Mackey-Glass. The chaotic time series is generated by a delay differential equation

$$\frac{\partial x(t)}{\partial t} = \frac{\alpha x(t - \tau)}{1 + x^\gamma(t - \tau)} - \beta x(t) \quad (13)$$

which was first investigated by Mackey and Glass [5]. In this study, 1000 input-output data samples which consist of four past values of $x(t)$ are used, i.e.,

$$[x(t - 18), x(t - 12), x(t - 6), x(t); x(t + 6)] . \quad (14)$$

There are four input dimensions and one output dimension to the system. The first 500 samples were used as the training set, while the remaining 500 pairs were used to test the performance of HyFIS-Yager-gDIC. Four different cases are considered in this study: the first case consists of data as described above, while noise of 5%, 10% and 15% of the power of the original signal are added in the remaining three cases. A measure of prediction accuracy is given by the root mean square error ($RMSE$) defined as

$$RMSE = \left[\frac{1}{n_5} \sum_{m=1}^{n_5} (d_m - y_m)^2 \right]^{1/2}. \quad (15)$$

In both the noise-free and noisy data, the proposed HyFIS-Yager-gDIC outperforms HyFIS in the number of rules formed and the $RMSE_{test}$ as shown in Table 1. Reductions of 14.7% and at least 4.55% in the number of rules formed are achieved in the noise-free and the noisy data respectively. In addition, there is a reduction of 21.6% and at least 8.35% in the $RMSE_{test}$ for the two cases respectively. This shows that HyFIS-Yager-gDIC has a higher noise tolerance level. A localized representation of the fuzzy sets is also achieved using gDIC as seen from Table 1. HyFIS-Yager-gDIC achieves a better training performance as can be seen from the faster and steeper descent in the training curves for HyFIS-Yager-gDIC as compared to those of HyFIS in Fig. 2. Lower converged minima for the $RMSE_{train}$ are achieved for HyFIS-Yager-gDIC in both the noise-free and noisy cases. In addition, the convergence rate of HyFIS-Yager-gDIC is faster than that of HyFIS, achieving a 2 folds, 2.5 folds, 2 folds, and 3 folds faster convergence rate than that of HyFIS in the four cases respectively.

Table 1. Comparison on Mackey-Glass prediction

Type of Data	Method	Num. of Labels		Num. of Rules	$RMSE_{test}$
		1	2	3	4
No Noise	HyFIS-CRI-Grid	5	5	5	5
	HyFIS-Yager-gDIC	4	5	4	4
5% Noise	HyFIS-CRI-Grid	5	5	5	5
	HyFIS-Yager-gDIC	4	5	4	4
10% Noise	HyFIS-CRI-Grid	5	5	5	5
	HyFIS-Yager-gDIC	4	5	4	4
15% Noise	HyFIS-CRI-Grid	5	5	5	5
	HyFIS-Yager-gDIC	4	5	5	4

Classification Problem: Iris Data. The Fisher's Iris data set [1] consists of 150 instances of Iris flowers equally distributed into three classes: Setosa, Virginica and Versicolor. Each instance consists of four physical attributes: sepal length, sepal width, petal length, and petal width. The data set is randomized and partitioned into a training and a testing set, with 51 (34%) points in the

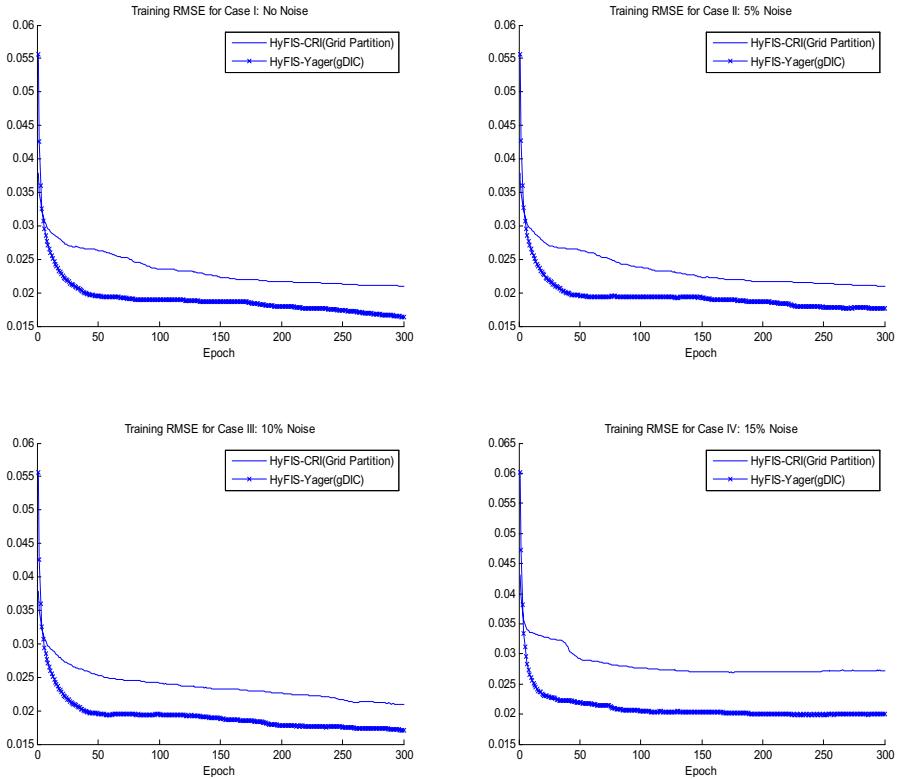


Fig. 2. Training curve for the four cases: (I) No noise (II) 5% noise (III) 10% noise (IV) 15% noise. As shown in the plots, the $RMSE_{train}$ for HyFIS-Yager-gDIC attains a lower convergent value than that of HyFIS in (I),(II),(III), and (IV). In addition, HyFIS-Yager-gDIC converges at least 2 folds faster than HyFIS in all the cases.

Table 2. Comparison on Iris classification

Method		Num. of Labels				Classification Rate (%)	
		Input		Output			
		1	2	3	4		
CV1	HyFIS-CRI-Grid	10	10	10	10	3	48 92.93
	HyFIS-Yager-gDIC	9	9	10	8	3	42(-12.5%) 98.99(+6.52%)
CV2	HyFIS-CRI-Grid	10	10	10	10	3	46 94.95
	HyFIS-Yager-gDIC	9	8	8	8	3	44(-4.35%) 96.97(+2.13%)
CV3	HyFIS-CRI-Grid	10	10	10	10	3	47 91.92
	HyFIS-Yager-gDIC	10	7	11	9	3	44(-6.38%) 94.95(+3.30%)

Table 3. Average Comparison on Iris classification

Method	Mean Classification Rate
HyFIS-CRI-Grid	93.27
HyFIS-Yager-gDIC	96.97(+3.97%)

training set and 99 (66%) points in the testing set. The training set comprises of 17 instances, and the testing set comprises of 33 instances from each of the three classes. The results of HyFIS-Yager-gDIC are subsequently cross-validated through three different groups of training and testing sets, labeled CV1, CV2 and CV3, as shown in Table 2. As can be seen from Table 2, the proposed HyFIS-Yager-gDIC outperforms HyFIS in the number of rules formed and the classification rate. A minimum reduction of 4.35% in the number of rules formed and a minimum improvement of 2.13% in the classification rate are attained in the three cross-validation folds, as achieved in CV2. As can be seen from Table 3, a mean classification rate of 96.97% is attained in the three-fold cross-validation for HyFIS-Yager-gDIC, achieving an average improvement of 3.97% over HyFIS.

6 Conclusion

This paper proposes a novel self-organizing Hybrid neural Fuzzy Inference System named HyFIS-Yager-gDIC. The fuzzy Yager inference scheme [3] is incorporated into the structure of HyFIS-Yager-gDIC to provide an intuitive logical network that emulates the human reasoning mechanism. In addition, HyFIS-Yager-gDIC requires no prior knowledge of the number of clusters and is able to automatically formulate the initial fuzzy rules from raw data. The system has been benchmarked on two case studies and superior performance is achieved.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, <http://www.ics.uci.edu/~mlearn/MLRepository.html>
2. Chen, L.H., Chiang, C.H.: New Approach to Intelligent Control Systems with Self-exploring Process. *IEEE Trans. Syst., Man, and Cybernetics* 33, 56–66 (2003)
3. Keller, J.M., Yager, R.R., Tahani, H.: Neural Network Implementation of Fuzzy Logic. *Fuzzy Sets and Syst.* 45, 1–12 (1992)
4. Kim, J., Kasabov, N.: HyFIS: Adaptive Neuro-fuzzy Inference Systems and Their Application to Nonlinear Dynamical Systems. *Neural Networks* 12, 1301–1319 (1999)
5. Mackey, M., Glass, L.: Oscillation and Chaos in Physiological Control Systems. *Science* 197, 287–289 (1977)
6. Nguyen, M.N., Shi, D., Quek, C.: Self-organizing Gaussian Fuzzy CMAC with Truth Value Restriction. *ICITA* 2, 185–190 (2005)
7. Pasha, M.F., et al.: EFIS: Evolvable-neural-based fuzzy inference system and its application for adaptive network anomaly detection. In: Yeung, D.S., Liu, Z.-Q., Wang, X.-Z., Yan, H. (eds.) *ICMLC 2005. LNCS*, vol. 3930, pp. 662–671. Springer, Heidelberg (2006)

8. Tung, W.L., Quek, C.: GenSoFNN: A Generic Self-organizing Fuzzy Neural Network. *IEEE Trans. Neural Networks* 13, 1075–1086 (2002)
9. Wang, L.X., Mendel, J.M.: Generating Fuzzy Rules by Learning from Examples. *IEEE Trans. Syst., Man, Cybern.* 22, 1414–1427 (1992)
10. Zadeh, L.A.: Calculus of Fuzzy Restrictions. *Fuzzy Sets and Their Appl. to Cog. and Decision Processes*, 1–39 (1975)

Parallel Ant Colony Optimizer Based on Adaptive Resonance Theory Maps

Hiroshi Koshimizu and Toshimichi Saito

Hosei University, Koganei, Tokyo, 184-0002 Japan

Abstract. This paper studies a parallel ant colony optimizer and its application to the traveling sales person problems. The parallel processing is based on the adaptive resonance theory map that divide the input space into subspaces. The ants are classified into two types: local ant for local search within either subspace and global ant for search of whole input space. Communication between local and global ants is a key for effective parallel processing. Applying the algorithm to basic bench marks, we can suggest that our algorithm realize fast and reasonable search.

1 Introduction

Ant colony optimizer (ACO) is a meta-heuristic optimizer algorithm inspired by pheromone effect of ants. Since the original ACO is presented, many variants and improved versions have been studied [1]-[3]. The ACO has variety of applications including self-organization of wireless sensor networks, fuzzy controller design, mechanical engineering systems and so on [4]-[6]. Also, the ACO can be effective to solve many combinatorial optimization problems [3] including graph coloring problem, quadratic assignment problem and traveling sales-person problems (TSP) [7]. In such applications, fast and reliable operation is required and parallel processing is an interesting technique to overcome the requests.

This paper presents a parallel ACO and considers its application to the TSP: a typical combinatorial optimization problem to find the shortest possible tour through the set of M cities that passes through each city exactly once. The parallel processing is based on the adaptive resonance theory maps (ART) that has ability to automatic classification of a data set on a feature space [8]-[11]. The ART divide of the input space of the city positions into some subspaces. We then classify all the ants into two types: local ant that is assigned within either of the subspaces for local search and global ant that searches whole input space. The ants drop pheromones on the path connecting cities. The pheromones are updated depending on behavior of ants to find a food source through path having strong pheromone deposits. Referring to the pheromone strength and communication among ants, the algorithm tries to find the optimal solution. Especially, communication between local and global ants are key for effective parallel processing. Applying the algorithm to basic bench marks and comparing the results with other series ACO, we can suggest that our parallel ACO can find sub-optimal solution faster and the solution fitness is competitive.

2 Paralleled ART-Based ACO Algorithm

Here we first explain the main-routine of the algorithm with definition of basic notations. Fig. 2(a) illustrates an objective input space consists of city positions for the TSP. Let S denote the input space of N city positions

$$S \equiv \{P_1, \dots, P_N\}, P_i \equiv (x_i, y_i) \in \mathbf{R}^2$$

where the input area is normalized as the unit square and P_i is the i -th city position.

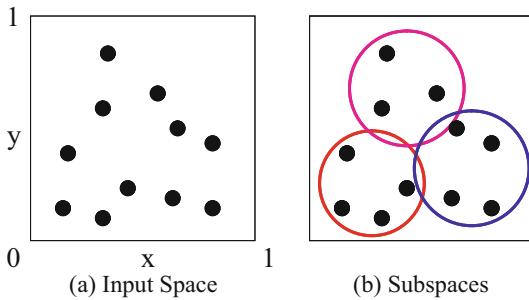


Fig. 1. Division of space. Dot denotes city position.

2.1 Main Routine

STEP 1: Applying the *ART*-subroutine defined afterward, S is divided into subspaces S_j :

$$S = \bigcup_{j=1}^{N_c} S_j, \quad S_j \cap S_k = \emptyset (j \neq k), \quad S_j = \{P_{j1}, \dots, P_{jn_j}\}$$

where N_c denotes the number of subspaces, P_{jk} denotes the k -th city position in the j -th subspace S_j and n_j denotes the number of cities in S_j .

STEP 2: We prepare M ants and classify them into two types: a local ant that can search either subspace and a global ant that can search whole input space. Let m be the number of global ants and let m_j be the number of local ants in the j -th subspace $M = (m + m_1 + m_2 + \dots + m_{N_c})$. Note that conventional ACO uses the global ant only.

STEP 3: The local ants try to find optimal tour in each subspace as shown in Fig. 2(a) by the EACO subroutine defined afterward.

STEP 4: Referring to the pheromone information of the local ants, the global ants try to find the optimal tour as shown in Fig. 2(b) by the EACO subroutine.

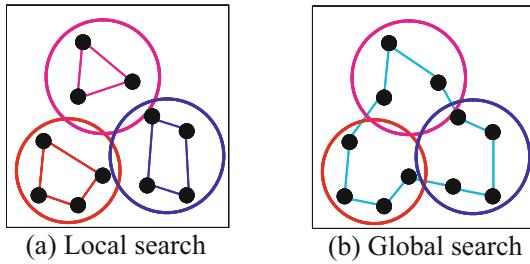


Fig. 2. (a) Search of local ant (b) Search of global ant

2.2 ART-Subroutine

The ART map consists of categories the number of which is time variant. Let $W_i(t) = (x_i(t), y_i(t), r_i(t))$ denote the i -th category at time t that is characterized by a circle at center (x_i, y_i) with radius r_i , $i = 1$ to $N_c(t)$ where $N_c(t)$ is the number of the categories at time t .

Step 1 (Initialization): Let $t = 0$, $N_c(t) = 1$ and let $W_i(t) \equiv (x_i(t), y_i(t), 0)$ where $i = 1$.

Step 2 (Selection): One city position $P_j = (x, y)$ is selected from S based on the order defined in the database and the city is applied as an input. If the input belongs to some category then the situation is preserved and goto Step 5. If the input does not belong to any category, we find the closest category form input.

$$d_c = \min_i(d_i), \quad d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (1)$$

If $d_c \leq \gamma$ then goto Step 3 where $\gamma \in [0, 1]$ is the vigilant parameter. If $d_c > \gamma$ then goto Step 4.

Step 3 (Learning): The closest category is updated to include the input as shown in Fig. 3(a) and goto Step 5.

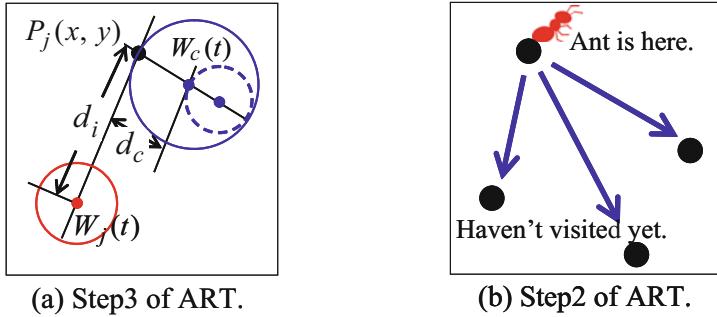
$$\begin{aligned} x'_c &= 0.5(r_c(x_c - x)/d_c + x_c + x) \\ y'_c &= 0.5(r_c(y_c - y)/d_c + y_c + y) \\ r'_c &= \sqrt{(x'_c - x)^2 + (y'_c - y)^2} \end{aligned} \quad (2)$$

Step 4 (Insertion of new category): New category with radius zero is generated at position of the input.

$$N_c = N_c + 1, \quad W_{N_c} = (x, y, 0) \quad (3)$$

Step 5 (End of search): Let $t = t + 1$. x_c , y_c and r_c are replaced with x'_c , y'_c and r'_c , respectively. Go to Step 2 and repeat until the maximum time limit $t = t_{ART}$.

If an input belongs to plural categories then the input is declared to belong to the closest category.

**Fig. 3.** Explanation of Subroutine

2.3 ACO-Subroutine

We have used ACO with elite strategy (EACO) where type pheromone deposits on the path in the current optimum tour is used. The elite strategy is known as a modified method of evolutionary computation. The local and global ants have different territory of search, however, can share some pheromone information. Here we define the local ants first and the definition is modified to the global ants. The local ants behavior in the l -th subspace is defined in the following 4 steps.

Step 1 (Initialization): Let τ_{ij} be the pheromone deposits on the path (i, j) between cities i and j . The $\tau_{ij}(t)$ is given by

$$\tau_{ij}(t) = (M + N)/\rho C^{NN} \quad (4)$$

where M is the total number of local and global ants. N is the number of cities in whole input space. $\tau_{ij}(t)$ is common for the local and global ants. And C^{NN} is the length of a tour generated by the nearest-neighbor heuristic in whole input space. ρ is a parameter of forgetting factor which models the evaporation over time of pheromone deposits. t denotes iteration number. Let $t = 0$.

Step 2 (Find a tour): For the k -th ant ($1 \dots m_l$) the visiting city is chosen by probability $\Phi_{ij,k}(t)$ as shown in Fig. 3(b) where the ant can visit a city in the subspace once. Then the k -th ant finds a tour based on $\Phi_{ij,k}(t)$. $\Phi_{ij,k}(t)$ is given by:

$$\Phi_{ij,k}(t) = \frac{\tau_{ij}(t)^\alpha \eta_{ij}^\beta}{\sum_{c \in C_{i,k}} \tau_{ij}(t)^\alpha \eta_{ij}^\beta} \quad (5)$$

where $1/\eta_{ij}$ is the distance of path (i, j) . α and β are adjustable parameters that control the weight of pheromone intensity and city information, respectively. $C_{i,k}$ is a set of cities such that ant k has not visited from city i yet. Referring to [1], we set $(\alpha, \beta) = (1, 5)$. The ants repeat choosing next city until all the cities are visited.

Step 3 (Pheromone update): Let $T_k(t)$ be a obtained tour by the k -th ant and $L_k(t)$ is the length of $T_k(t)$. Compute the $L_k(t)$. The pheromone deposits $\tau_{ij}(t)$ is updated by Eq. (6).

$$\tau_{ij}(t) \leftarrow (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}^l(t) + \Delta\tau_{ij}^g(t) \quad (6)$$

where $\Delta\tau_{ij}^l(t)$ and $\Delta\tau_{ij}^g(t)$ are pheromone deposits of the local and global ants, respectively. This equation realizes pheromone communication between local and global ants. $\Delta\tau_{ij}^g(t)$ is define afterward and $\Delta\tau_{ij}^l(t)$ is defined by:

$$\Delta\tau_{ij}^l(t) = \sum_{k=1}^{m_l} \Delta\tau_{ij,k}(t) + n_l \Delta\tau_{bs}^l$$

$\Delta\tau_{ij,k}(t)$ is the pheromon deposit for the local tour of the k -th ant:

$$\Delta\tau_{ij,k}(t) = \begin{cases} n_l / (N \times L_k(t)) & \text{if } (i, j) \in T_k(t) \\ 0 & \text{otherwise} \end{cases}$$

$\Delta\tau_{bs}^l$ is the pheromone deposit for the local best tour:

$$\Delta\tau_{bs}^l = \begin{cases} n_l / (N \times L_{bs}^l) & \text{if } (i, j) \text{ belongs to } T_{bs}^l \\ 0 & \text{otherwise} \end{cases}$$

where T_{bs}^l is the shortest tour in the l -th subspace, and L_{bs}^l is the length of T_{bs}^l .

Step 4 (End of search): Let $t=t+1$. Go to Step 2 and repeat until the maximum time limit $t = t_{max}$.

This definition is translated into the definition of global ants behavior. First, the "input subspace" is replaced with "whole input space". The Step1, 2, and 4 are the same. The pheromone deposit $\Delta\tau_{ij}^g(t)$ in Step 3 is defined as the following:

$$\Delta\tau_{ij}^g(t) = \sum_{k=1}^m \Delta\tau'_{ij,k}(t) + N \Delta\tau_{bs}^g$$

where

$$\Delta\tau'_{ij,k}(t) = \begin{cases} 1/L_k(t) & \text{if } (i, j) \in T_k(t) \\ 0 & \text{otherwise} \end{cases}$$

$\Delta\tau_{bs}^g$ is given as follows:

$$\Delta\tau_{bs}^g = \begin{cases} 1/L_{bs}^g & \text{if } (i, j) \text{ belongs to } T_{bs}^g \\ 0 & \text{otherwise} \end{cases}$$

The T_{bs}^g is the shortest tour and L_{bs}^g is the length of T_{bs}^g .

3 Experiments

In order to confirm the algorithm performance, we have tried to solve the problem 'eil51'- 51 cities in web site <http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsplib.html/>. The input space is shown in Fig 4(a). for which the each optimal tour length is known and is normalize as 1 in the experiment. Note that our algorithm is characterized by 5 parameters: the vigilance parameter γ , the maximum time limit t_{ART} , the number of ant M , the evaporation parameter ρ , iteration limit t_{max} . They are fixed after trial-and-errors:

$$\gamma = 0.40, t_{ART} = N, M = N/4, \rho = 0.1, t_{max} = 500.$$

where $\gamma = 0.40$ is from Ref. [11].

Fig 4(a) illustrates a result of the *ART-subroutine*. In each category, local ants try to find a tour and the result is shown in Fig 4(b), (d). Communication with the local ants, the global ants find a tour as shown in Fig 4(c), (e) and (f). Fig 5 shows search process of global ants. Table II summarizes the results in several measures after 20 runs for four algorithms: ART-EACO is the algorithm defined in Secsion 2, ART-EACO without elite strategy (without $\Delta\tau_{bs}^g$) is the ART-ACO. ART-EACO/ART-ACO without parallel strategy (without local ants) is the EACO/ACO. Comparing ART's processing time with ACO's processing time, ACO's processing time is too short time. Therefore, We assume that ART processing time = 0. In the four algorithm, total number of the ants are the same (M) (M consists of global ants only). In using ART-algorithm, global ant $m = M/2$, local ant in j -th subspace $m_j = mn_j/N$. We have tried similar numerical experiments for the other problem of 226 cities and the results are summarized in Table 1. These results suggest that the ART-EACO/ART-ACO realizes faster search than EACO/ACO, the series EACO/ACO can find briefly shorter tour than parallel ones and ART-EACO/EACO realizes more effective search than ART-ACO/ACO.

Table 1. The summary of each problem (Optimal tour length is normalized as 1, respectively). Ave. Length = average tour length, Best Length = the shortest length, ACO's processing time of each problem is normalized as 1, respectively.

Problem Name	ALG name	Ave. Length	Best Length	Processing Time
eil51	ACO	1.071	1.044	1.000
	ART-ACO	1.072	1.046	0.621
	EACO	1.023	1.008	0.993
	ART-EACO	1.038	1.017	0.639
pr226	ACO	1.052	1.040	1.000
	ART-ACO	1.056	1.045	0.555
	EACO	1.015	1.009	0.999
	ART-EACO	1.035	1.019	0.551

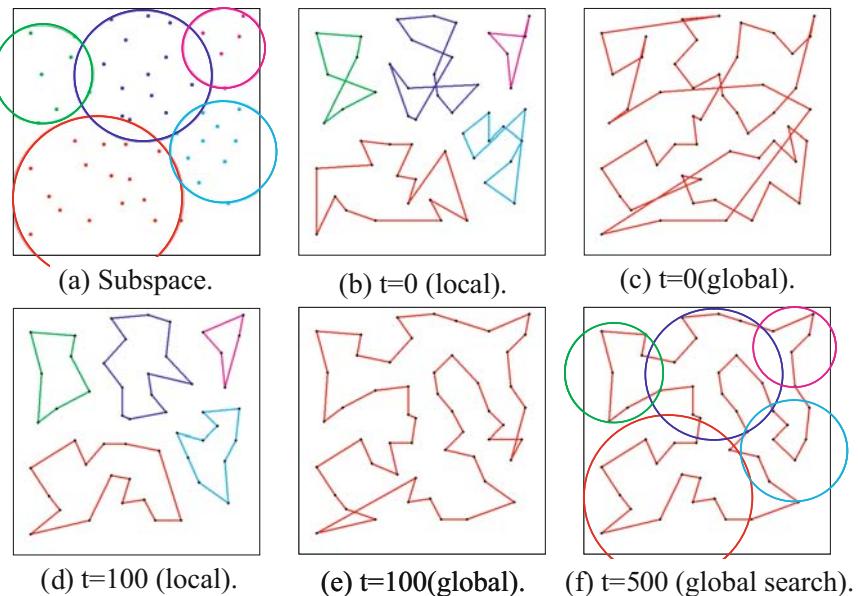


Fig. 4. Learning process for eil51 (5 subspaces, The obtained length is 1.017)

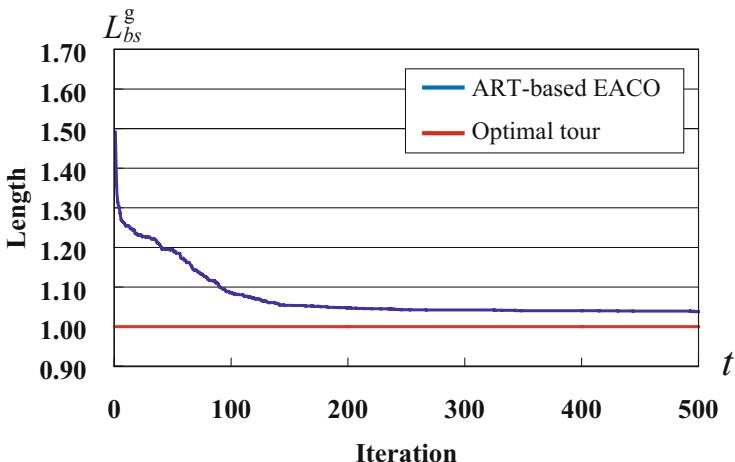


Fig. 5. Search process of global ants in ART-EACO Average for 20 trials (Optimal length = 1.000)

4 Conclusions

A parallel ACO and its application to the TSP is studied in this paper. The ART is used to divide the input space into subspaces for parallel processing.

The two kinds of ants, global and local ants, communicate to each other for efficient search. Performing basic numerical experiments, it is suggested that our algorithm can find tours much faster than serial ACO. In order to establish the algorithm, we must consider many problems including : 1) analysis of parameters role for automatic parameter setting, 2) analysis of search process, 3) development of effective series/parallel cooperation by ants grouping, 4) comparison of various algorithms in key measures and 5) application to practical problems and/or complex bench marks.

References

1. Dorigo, M., Stutzle, T.: *Ant Colony Optimization*. Bradford Books (2004)
2. Engelbrecht, A.P.: *Fundamentals of computational swarm intelligence*. Wiley, Chichester (2005)
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Inspiration for Optimization from Social Insect Behaviour. *Nature* 406, 39–42 (2000)
4. Rui, W., Yan, L., Ganqiang, Y., Chaoxia, L., Quan, P.: Swarm Intelligence for the Self-Organization of Wireless Sensor Network. In: Proc. of IEEE/CEC, pp. 3180–3184 (2006)
5. Juang, C., Lu, C., Lo, C., Wang, C.: Ant Colony Optimization Algorithm for Fuzzy Controller Design and Its FPGA Implementation. *IEEE Trans. Industrial Electronics* 55(3) (March 2008)
6. Viana, F.A.C., Kotinda, G.I., Rade, D.A., Steffen Jr., V.: Can Ants Design Mechanical Engineering System? In: Proc. of IEEE/CEC, pp. 3173–3179 (2006)
7. Hara, A., Ichimura, T., Fuita, N., Takahama, T.: Effective Diversification of Ant-Based Search using Colony Fission and Extinction. In: Proc. of IEEE/CEC, pp. 3173–3179 (2006)
8. Anagnostopoulos, G.C., Georgopoulos, M.: Ellipsoid ART and ARTMAPS for incremental clustering and classification. In: Proc. IEEE/INNS IJCNN, pp. 1221–1226 (2001)
9. Parsons, O., Carpenter, G.A.: ARTMAP neural networks for information fusion and data mining: Map production and target recognition methodologies. *Neural Networks* 16, 1075–1089 (2003)
10. Oshime, T., Saito, T., Torikai, H.: ART-based parallel learning of growing sOMs and its application to TSP. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) *ICONIP 2006. LNCS*, vol. 4232, pp. 1004–1011. Springer, Heidelberg (2006)
11. Takanashi, M., Torikai, H., Saito, T.: An approach to fusion of growing self-organizing maps and adaptive resonance theory maps. *IEICE Trans. Fundamentals* E90-A(9), 2047–2050 (2007)

Covariate Shift and Incremental Learning

Koichiro Yamauchi

Graduate School of Information Science and Technology, Hokkaido University,
Sapporo Hokkaido 060-0814, Japan

Abstract. Learning strategies under *covariate shift* have recently been widely discussed. Under covariate shift, the density of learning inputs is different from that of test inputs. In such environments, learning machines need to employ special learning strategies to acquire a greater capability to generalize through learning.

However, incremental learning methods are also for learning in non-stationary learning environments, which would represent a kind of covariate-shift. However, the relation between covariate shift environments and incremental learning environments has not been adequately discussed.

This paper focuses on the covariate shift in incremental learning environments and our re-construction of a suitable incremental learning method.

Keywords: Incremental Learning, Covariate shift, Radial Basis Function Neural Network (RBFNN), Generalization error, Student's-t distribution.

1 Introduction

Let the learning samples be (\mathbf{x}_b, y_b) ($b = 1, 2, \dots$), whose joint probability distribution is $P(\mathbf{x}, y) = P(y|\mathbf{x})P(\mathbf{x})$. To achieve successful online learning of the relation between \mathbf{x} and y : $P(y|\mathbf{x})$ using a model-based learning machine, we need independent and identically distributed (i.i.d) \mathbf{x} 's. In many cases, however, $P(\mathbf{x})$ is not stable so that online learning usually fails. To overcome this problem, many researchers have developed incremental learning algorithms [1] [2] [3] [4] [5] [6] [7] [8]. These algorithms allow $P(y|\mathbf{x})$ to be learned even if $P(\mathbf{x})$ is changing. In extreme cases, one-pass learning is also available.

However, learning strategies under *covariate shift* have recently been widely been discussed (e.g [9] [10]). In covariate shift, the learning input density $P(\mathbf{x})$ is not equivalent to that of the test samples. In such environments, learning machines need to adjust their parameters to minimize the following weighted error function to acquire greater generalization capabilities.

$$\hat{E} = \sum_{i=1}^N (F(\mathbf{x}_i) - f_\theta(\mathbf{x}_i))^2 W(\mathbf{x}_i), \quad (1)$$

where $W(\mathbf{x})$ is the weight for each sample and $W(\mathbf{x}) \equiv (q(\mathbf{x})/P(\mathbf{x}))^\lambda$, where $q(\mathbf{x})$ denotes the density of \mathbf{x} for test samples and $0 < \lambda \leq 1$ is the flattening

parameter. Here, $f_\theta(\mathbf{x})$ denotes the output of the learning machine and $F(\mathbf{x})$ denotes the target output.

Thus, we can see that the situation for the incremental-learning machine is similar to that for covariate shift. Therefore, in incremental learning, $q(\mathbf{x})$ corresponds to the input density for the new learning samples subsequently presented. In this study, let us assume that the incremental learning environments are also a subset of covariate shift, and try to redevelop a suitable incremental learning algorithm from the point of view of covariate shift.

The next section describes the incremental learning scheme assumed in this study. Section 3 presents a model of incremental learning environments. Section 4 proposes a suitable learning method. Section 5 presents the benchmark test results, and Section 6 is the conclusion.

2 Learning Scheme

Let us consider the simplified incremental learning scheme depicted in Fig. 1, which has a fundamental incremental learning architecture having a rehearsal process such as those proposed in Refs [1] [2] [4] [5] [6] [8].

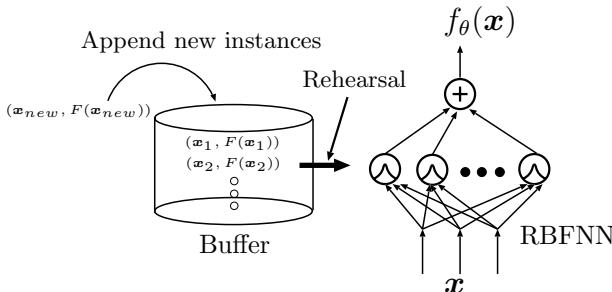


Fig. 1. Incremental learning scheme in this study

This system repeats two phases, i.e., those for recording and rehearsal, alternately. During the recording phase, the learning system obtains several new learning samples and stores them into a buffer with unlimited capacity. After the recording phase, the system begins the next rehearsal phase. During the rehearsal phase, entire samples in the buffer are presented to the learner.¹ Note that the learner rehearses not only the new novel learning samples but also old learning samples stored in the buffer so that it does not cause “catastrophic forgetting.” For simplicity, we were not concerned here with any strategies to reduce the computational power for the rehearsal process.

¹ Although incremental learning systems that include a rehearsal process would waste computational power, it would be more effective to acquire a compact network [11].

The learner used was a radial bases function neural network (RBFNN). Let $f_\theta(\mathbf{x})$ be the output value of the RBFNN. $f_\theta(\mathbf{x})$ is

$$f_\theta(\mathbf{x}) = \sum_{j=1}^M w_j \exp\left(-\frac{\|\mathbf{x} - \mathbf{u}_j\|^2}{2v_j^2}\right), \quad (2)$$

where M denotes the number of hidden units.

The aim of the learning system was to minimize the following evaluation function:

$$E = \int (F(\mathbf{x}) - f_\theta(\mathbf{x}))^2 q(\mathbf{x}) d\mathbf{x}, \quad (3)$$

where $F(\mathbf{x})$ denotes a target output value and $q(\mathbf{x})$ denotes the actual input density. Note that $q(\mathbf{x})$ is not equivalent to the learning input density $P(\mathbf{x})$.

3 Modeling of Incremental Learning Environments

To construct a learning method to minimize the weighted error function in Eq(1), we need to derive the actual input density $q(\mathbf{x})$ and the learning sample's input density $P(\mathbf{x})$. In almost all of the previous studies (eg. [10]), $q(\mathbf{x})$ was predicted from actual test samples. The test samples, however, correspond to the new samples, which will be given in the future under incremental learning environments. Therefore, we must somehow predict $q(\mathbf{x})$ from the given learning samples in advance.

The following predicted distribution of \mathbf{x} from N number of learning samples, which have been presented up till now, is used in this study. Let $\hat{q}(\mathbf{x})$ be the predicted $q(\mathbf{x})$.

$$\hat{q}(\mathbf{x}) = \int P(\mathbf{x}|S)P(S|\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) dS, \quad (4)$$

where S denotes the parameter vector that represents the input density function. In the simplest case, $q(\mathbf{x})$ would be a normal probability distribution. In such cases, according to Bayes theorem, $\hat{q}(\mathbf{x})$ should be approximated by a *Student's-t* distribution of $N - 1$ -degrees-of-freedom. Therefore,

$$\hat{q}(\mathbf{x}) = \frac{\Gamma[(N-1+p)/2]}{((N-1)\pi)^{p/2}\Gamma[(N-1)/2]|\Sigma|^{1/2}} \left[1 + \frac{(\mathbf{x} - \mathbf{u})^T \Sigma^{-1} (\mathbf{x} - \mathbf{u})}{N-1}\right]^{-(N-1+p)/2}, \quad (5)$$

where $p = \dim(\mathbf{x})$, $\mathbf{u} = E[\mathbf{x}]$ and Σ denote the variance-covariance matrix. Note that *Student's-t* distribution converges to actual normal distribution $q(\mathbf{x})$ while increasing the number of presented learning samples.

$P(\mathbf{x})$ is predicted by the learning samples presented up till now. Let $\hat{P}(\mathbf{x})$ be the predicted $P(\mathbf{x})$. We assume that $\hat{P}(\mathbf{x})$ can be represented by $\mathcal{N}(\Sigma, \mathbf{u})$, where $\Sigma = \frac{1}{N} \sum_{b=1}^N (\mathbf{x}_b - \mathbf{u})(\mathbf{x}_b - \mathbf{u})^T$ and $\mathbf{u} = \frac{1}{N} \sum_{b=1}^N \mathbf{x}_b$.

Therefore, $\hat{q}(\mathbf{x})/\hat{P}(\mathbf{x})$ is

$$\frac{\hat{q}(\mathbf{x})}{\hat{P}(\mathbf{x})} = \left(\frac{2}{N-1} \right)^{p/2} \frac{\Gamma[(N-1+p)/2]}{\Gamma[(N-1)/2]} \frac{\left[1 + \frac{(\mathbf{x}-\mathbf{u})^T \Sigma^{-1} (\mathbf{x}-\mathbf{u})}{N-1} \right]^{-(N-1+p)/2}}{\exp(-\frac{1}{2}(\mathbf{x}-\mathbf{u})^T \Sigma^{-1} (\mathbf{x}-\mathbf{u}))}. \quad (6)$$

In many cases, however, $P(\mathbf{x})$ and $q(\mathbf{x})$ have complex shapes so they are difficult to approximate by using a single normal or *Student's t* distribution. To overcome this difficulty, the given learning samples in this study were divided into several clusters by k -means clustering, and prediction was achieved in each cluster. The number of clusters k was determined automatically by using the method proposed in [12]. Therefore, the system tries all k -means clustering for $k = 1 \cdots K$ and finds $k^* = \arg \min_k \rho(k)$, which is defined as

$$\rho(k) \equiv \frac{1}{k-1} \sum_{i=1}^{k-1} \max_{i < j \leq k} \left(\frac{\eta_i + \eta_j}{\xi_{ij}} \right) \quad \eta_i = \frac{1}{N_i} \sum_j D(\mathbf{x}_j, \mathbf{u}_i), \quad \xi_{ij} = D(\mathbf{u}_i, \mathbf{u}_j), \quad (7)$$

where $\mathbf{u}_i = \frac{1}{N_i} \sum_{b \in C_i} \mathbf{x}_b$ and $D(\mathbf{x}, \mathbf{u}) = \|\mathbf{x} - \mathbf{u}\|$.

Then, the estimate is applied to each cluster. Therefore, if \mathbf{x} is the i -th cluster, the corresponding $W_i(\mathbf{x})$ is

$$W_i(\mathbf{x}) = \left\{ \left(\frac{2}{N_i-1} \right)^{p/2} \frac{\Gamma[(N_i+p-1)/2]}{\Gamma[(N_i-1)/2]} \frac{\left[1 + \frac{(\mathbf{x}-\mathbf{u}_i)^T \Sigma_i^{-1} (\mathbf{x}-\mathbf{u}_i)}{N_i-1} \right]^{-(N_i+p-1)/2}}{\exp(-\frac{1}{2}(\mathbf{x}-\mathbf{u}_i)^T \Sigma_i^{-1} (\mathbf{x}-\mathbf{u}_i))} \right\}^\lambda, \quad (8)$$

where N_i , Σ_i , and \mathbf{u}_i correspond to the number of samples, the variance-covariance matrix, and mean \mathbf{x} of the i -th cluster.

4 Suitable Learning Method

The RBFNN learns all samples stored in the buffer in each rehearsal phase. As discussed in Section 1, the RBFNN has to maximize the weighted log likelihood in Eq.(1). We can employ various learning methods for the RBFNN to maximize Eq.(1). In this study, the learning system maximizes Eq.(1) using the online learning method proposed by Platt [13]. To do this, instead of using the original error function, the following weighted error function is used:

$$\hat{E}(\mathbf{x}_b) \equiv (F(\mathbf{x}_b) - f_\theta(\mathbf{x}_b))^2 W(\mathbf{x}_b), \quad (9)$$

where $(\mathbf{x}_b, F(\mathbf{x}_b))$ denotes a learning sample stored in buffer B and $W(\mathbf{x}_b)$ denotes the weight of the learning sample. Note that $W(\mathbf{x})$ is

$$W(\mathbf{x}) = W_i(\mathbf{x}) \quad \text{if } \mathbf{x} \in \text{cluster}_i \quad (10)$$

Although the learning method includes a method of increasing the number of hidden units, we restricted the maximum number of hidden units to a certain number, M' , to accurately evaluate what effect the weighted error function had.

The pseudo-code for this learning method is shown in Fig. 2. Note that the learning algorithm is accomplished when the system encounters m number of new learning samples.

Input: new learning samples $(\mathbf{x}_n, F(\mathbf{x}_n))$ $n = 1 \cdots m$.

Output: parameter vector θ for the RBFNN.

```

//Recording phase

for each  $\mathbf{x}_n$ 
     $B \leftarrow (\mathbf{x}_n, F(\mathbf{x}_n))$ 
end
Calculate  $\rho(k)$  for  $k = 1 \cdots K$ . ( $\rho(k)$  is defined by Eq(7))
 $k_{best} \leftarrow \arg \min_k \rho(k)$ 
Divide  $B$  into  $k_{best}$  clusters and detect  $N_i$ ,  $\Sigma_i$  and  $\mathbf{u}_i$  for  $i = 1 \cdots k_{best}$ .
Detect the weight  $W_i(\mathbf{x})$  for all samples stored in  $B$ . (Eq(8))

//Rehearsal phase

repeat until  $\frac{1}{N} \sum_b \hat{E}(\mathbf{x}_b) < a$  ( $\hat{E}(\mathbf{x}_b)$  is defined by Eq(9)(10))
    choice  $(\mathbf{x}_b, F(\mathbf{x}_b))$  from  $B$  randomly.
    if  $M < M'$  and  $\arg \min_j \|\mathbf{x}_b - \mathbf{u}_j\| > \delta$  and  $\sqrt{\hat{E}(\mathbf{x}_b)} > \epsilon$ 
        a new hidden unit with  $\mathbf{u}_{M+1} \leftarrow \mathbf{x}_b$ ,  $\sigma_{M+1} \leftarrow \kappa \min_\alpha (\|\mathbf{x}_b - \mathbf{u}_\alpha\|)$  is allocated.
         $M \leftarrow M + 1$ 
    end
     $\theta \leftarrow \theta - \eta \nabla_\theta \{(F(\mathbf{x}_b) - f_\theta(\mathbf{x}_b))^2 W(\mathbf{x}_b)\}$ 
end
```

Fig. 2. Pseudo-code for learning algorithm

5 Experiments

The system was tested using one synthetic dataset and four benchmark datasets. Although this learning method repeats the *recording* of new learning samples and *rehearsal* for the recorded samples alternately, only one set of recording and rehearsal phases was carried out because this was sufficient to check system behavior.

5.1 Illustrative Example in One-Dimensional Synthetic Dataset

To evaluate system behavior accurately, the following simple dataset was used.

$$(x, y) = (x, 1.5) \quad \text{where, } x \sim \frac{1}{3}\mathcal{N}(-20, 2) + \frac{1}{3}\mathcal{N}(0, 2) + \frac{1}{3}\mathcal{N}(20, 2) \quad (11)$$

Note that $F(x) = y = 1.5$. There were 150 learning samples and the maximum number of hidden units was set to 6. The error threshold, a , and the maximum number of clusters, K , in Fig. 2 correspond to 0.0002 and 20. λ was set to 1. In the experiment, we compared how well the proposed learning method and a conventional method performed. Note that the conventional learning method used following error function instead of using Eq.(9).

$$\hat{E}(\mathbf{x}_b) \equiv (F(\mathbf{x}_b) - f_{\theta}(\mathbf{x}_b))^2, \quad (12)$$

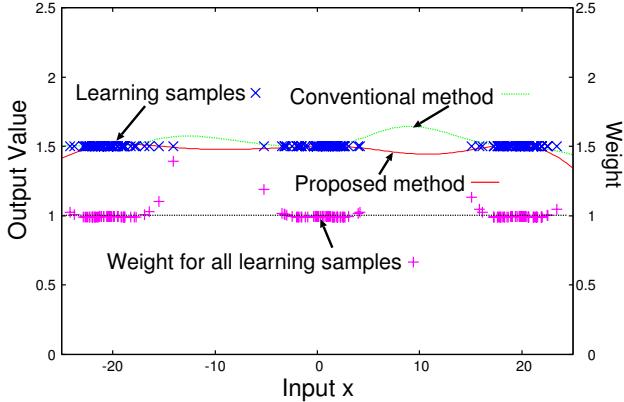


Fig. 3. Output curves for proposed and conventional methods

Fig. 3 plots the output curves for the two methods. We can see that the weight for each learning sample is larger than 1 if x is close to edges. This means that if the frequency of the appearance of the learning samples is low, the corresponding weight increases large. However, the learner does not learn such learning samples well with conventional learning due to their low frequency of appearance. With the proposed method, the leaner learns such patterns well by increasing the corresponding weights. Consequently, we can see from Fig. 3 that the output curve for the proposed method fits better than that for the conventional method at edges.

5.2 System Behavior with Benchmark Datasets

To investigate the effect of λ and the maximum number of hidden units M' (see Fig 2), the system performances for the four benchmark datasets were evaluated with varying λ and M' . The datasets used were CPU-performance, mackay-glass-chaos, housing and servo in the UCI machine learning repository. The selected values for λ were 0, 0.1, 0.25, 0.5, 0.75 and 1. Note that $\lambda = 0$ corresponds to the conventional learning method. N was varied around the minimum number that the network can learn the target function. To avoid the learning process being unstable, the weight for each sample was restricted to be less than 10.

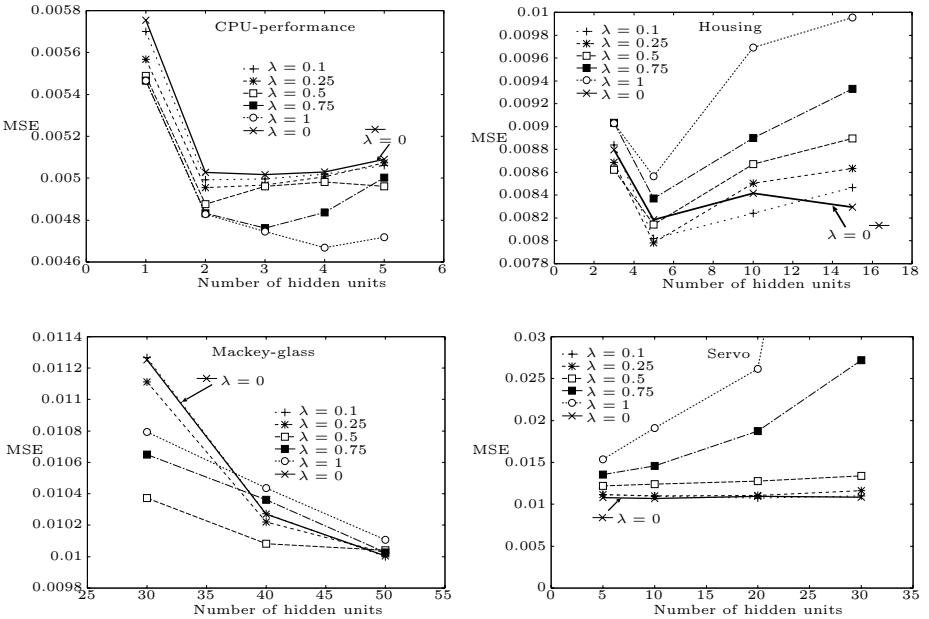


Fig. 4. System performances to benchmark datasets ($\lambda = 0$ corresponds to the conventional method.)

The parameters for the resource allocating network were $\kappa = 2.5$, $\epsilon = 0.001$, $\delta = 0.0001$ and $a = 0.003$ for all datasets.

Fig 4 shows the mean square error to test samples after the learning of 100, 200, 100 and 80 samples of CPU-performance, mackay-glass-chaos, housing and servo, respectively. Each test was repeated 50 trials with changing the learning sample randomly. The test samples were whole samples for each dataset except for mackay-glass-chaos. In the case of mackay-glass chaos, the test samples were randomly selected 500 successive samples. The results were averaged over the 50 trials. Note that the number of hidden units was reached to M' in all test.

In the case of cpu-performance, we can see that the proposed method outperformed the conventional one in MSE for all conditions (see Fig 4). In the case of mackay-glass chaos and housing, the proposed one outperformed the conventional one when the number of hidden units is small and λ is a certain value which is less than 1. This suggests that the proposed method is effective to get a better learning results when the network size is restricted.

In the case of servo dataset, however, the proposed system failed to evaluate the distribution of inputs correctly so that the almost all of MSE's were larger than those of the original one. The servo dataset includes several symbolic attributes so that it may be hard to approximate $q(\mathbf{x})$ as the mixture of normal distributions.

6 Conclusion and Discussion

This study pointed out that incremental learning environments are similar to covariate shift and it tried to redevelop an incremental learning method that would be suitable for covariate shift.

The new method can predict the input density for new learning samples, which will be presented in the later incremental learning steps, and it makes the learner undergo proactive learning according to the predicted input density. Therefore, the method reinforces the learning effect of novel isolated learning samples.

In this study, we investigated system behaviors under several benchmark datasets. The results suggests that the method is effective to get better result under the situation where the network size is small and the distribution of given learning samples can be approximated as a mixture of normal distributions. To do this, we also have to choice the best λ . If we require enough computational resources, we can use an information criterion IC_w proposed by Shimodaira [9] to find the optimal λ as well as the optimal network size.

For next work, we want to build a prediction method for $q(\mathbf{x})$, which can support various kinds of distributions, and establish an effective model selection method with low computational complexity.

References

- Yoneda, T., Yamanaka, M., Kakazu, Y.: Study on optimization of grinding conditions using neural networks – a method of additional learning. *Journal of the Japan Society of Precision Engineering/Seimitsu kogakukaishi* 58(10), 1707–1712 (1992)
- Yamakawa, H., Masumoto, D., Kimoto, T., Nagata, S.: Active data selection and subsequent revision for sequential learning with neural networks. In: World congress of neural networks (WCNN 1994), vol. 3, pp. 661–666 (1994)
- Schaal, S., Atkeson, C.G.: Constructive incremental learning from only local information. *Neural Computation* 10(8), 2047–2084 (1998)
- Yamauchi, K., Yamaguchi, N., Ishii, N.: Incremental learning methods with retrieving interfered patterns. *IEEE Transactions on Neural Networks* 10(6), 1351–1365 (1999)
- French, R.M.: Pseudo-recurrent connectionist networks: An approach to the “sensitivity stability” dilemma. *Connection Science* 9(4), 353–379 (1997)
- Ans, B., Roussert, S.: Neural networks with a self-refreshing memory: knowledge transfer in sequential learning tasks without catastrophic forgetting. *Connection Science* 12(1), 1–19 (2000)
- Kasabov, N.: Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *IEEE Transactions on Systems, Man, and Cybernetics* 31(6), 902–918 (2001)
- Ozawa, S., Toh, S.L., Abe, S., Pang, S., Kasabov, N.: Incremental learning of feature space and classifier for face recognition. *Neural Networks* 18, 575–584 (2005)
- Hidetoshi, S.: Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90(2), 227–244 (2000)

10. Sugiyama, M., Nakajima, S., Kashima, H., von Bünau, P., Kawanabe, M.: Direct importance estimation with model selection and its application to covariate shift adaptation. In: Twenty-First Annual Conference on Neural Information Processing Systems (NIPS2007) (December 2007)
11. Yamauchi, K., Hayami, J.: Incremental learning and model selection for radial basis function network through sleep. IEICE Transactions on Information and Systems E90-D(4), 722–735 (2007)
12. Ngo, C.-W., Pong, T.-C., Zhang, H.-J.: On clustering and retrieval of video shots through temporal slice analysis. IEEE Transactions on Multimedia 4(4), 446–458 (2002)
13. Platt, J.: A resource allocating network for function interpolation. Neural Computation 3(2), 213–225 (1991)

A Novel Incremental Linear Discriminant Analysis for Multitask Pattern Recognition Problems*

Masayuki Hisada¹, Seiichi Ozawa¹, Kau Zhang¹,
Shaoning Pang², and Nikola Kasabov²

¹ Graduate School of Engineering, Kobe University, Kobe 657-8501, Japan

073t248t@stu.kobe-u.ac.jp, ozawasei@kobe-u.ac.jp

² Knowledge Engineering & Discover Research Institute

Auckland University of Technology

Private Bag 92006, Auckland 1020, New Zealand

shaoning.pang@aut.ac.nz, nik.kasabov@aut.ac.nz

Abstract. In this paper, we propose a new incremental linear discriminant analysis (ILDA) for multitask pattern recognition (MTPR) problems in which training samples of a specific recognition task are given one after another for a certain period of time and the task is switched to another related task in turn. The Pang et al.'s ILDA is extended such that a discriminant space of the current task is augmented with effective discriminant vectors that are selected from other related tasks based on the class separability. We call the selection and augmentation of such discriminant vectors *knowledge transfer of feature subspaces*. In the experiments, the proposed ILDA is evaluated for the four MTPR problems, each of which consists of three multi-class recognition tasks. The results demonstrate that the proposed ILDA outperforms the ILDA without the knowledge transfer with regard to both the class separability and recognition accuracy. From the experimental results, we confirm that the proposed knowledge transfer mechanism works well to construct effective discriminant feature spaces incrementally.

1 Introduction

The main purpose of multitask learning (MTL) is to learn unknown tasks efficiently by utilizing the knowledge of related tasks as an inductive bias [1]. A practical goal is to improve the generalization performance of a learning system even if training samples of a new task are not sufficiently provided. To make up a shortage of training samples, it is particularly important how an inductive bias is earned from other related tasks and how it is transferred to the system for learning.

* The authors would like to thank Professor Shigeo Abe for his helpful comments and discussions. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C) 205002205.

Caruana [1] first proposed an MTL model in which multiple tasks are learned in parallel by a feedforward neural network. Silver and Mercer [2] extend the Caruana's MTL model so that sequentially given tasks can be learned stably without the catastrophic forgetting. In the two models, the inductive bias corresponds to the internal representation in the hidden layer and the knowledge transfer is mainly realized by sharing the hidden units among different tasks.

On the other hand, the dimensional reduction of inputs by feature extraction plays an important role in pattern recognition, prediction, data mining, etc. It seems natural that useful features for a specific task are also useful for other related tasks if the tasks have the same input domain. However, there have been so far proposed only a few approaches to the knowledge transfer for feature extraction [3].

In this paper, we propose a novel approach to online feature extraction for a multitask pattern recognition (MTPR) problem [4] in which multiple pattern recognition tasks are defined for the same input domain and given to learn multiple descriptions (class labels) of objects. To earn useful features in such an MTPR problem, we extend the Pang et al.'s incremental linear discriminant analysis (ILDA) [5] such that a discriminant feature space model is incrementally updated not only by learning the current task samples but also by combining effective discriminant vectors that are selected from the discriminant feature space models for other related tasks based on the class separability.

This paper is organized as follows. Section 2 gives a quick review of the conventional linear discriminant analysis (LDA) and the Pang et al.'s ILDA. In Section 3, we first define the learning environment of MTPR problems, and present a condition of knowledge transfer that leads to the enhancement in the class separability of discriminant features. Then, we propose an extended ILDA algorithm for MTPR problems. In Section 4, the performance of the proposed model is evaluated for the four MTPR problems which are defined from the UCI data sets. Section 5 gives our conclusions.

2 ILDA for Single-Task Pattern Recognition

Assume that we have a set of N samples and they belong to one of C classes. The set of class c samples is denoted as $\mathbf{X}_c = \{\mathbf{x}_{c1}, \dots, \mathbf{x}_{cn_c}\}$ ($c = 1, \dots, C$), where $\mathbf{x}_{cj} \in \mathcal{R}^n$ is the j th sample of class c and n_c is the number of class c samples. Then, the whole sample set is denoted as $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_C\}$. For these samples, we can define the following between-class scatter matrix \mathbf{S}_B and within-class scatter matrix \mathbf{S}_W :

$$\mathbf{S}_B \stackrel{\text{def}}{=} \sum_{c=1}^C n_c (\bar{\mathbf{x}}_c - \bar{\mathbf{x}})(\bar{\mathbf{x}}_c - \bar{\mathbf{x}})^T \quad (1)$$

$$\mathbf{S}_W \stackrel{\text{def}}{=} \sum_{c=1}^C \mathbf{V}_c \quad (2)$$

where

$$\mathbf{V}_c = \sum_{j=1}^{n_c} (\mathbf{x}_{cj} - \bar{\mathbf{x}}_c)(\mathbf{x}_{cj} - \bar{\mathbf{x}}_c)^T. \quad (3)$$

Here, $\bar{\mathbf{x}}_c$ and $\bar{\mathbf{x}}$ are the mean vectors of the class c samples and all the samples, respectively.

In LDA, a discriminant vector \mathbf{w} is obtained by maximizing the following separability measure:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}. \quad (4)$$

In general, up to $(C - 1)$ discriminant vectors can be obtained for a C -class recognition problem due to the rank of \mathbf{S}_B . Let us define a set of such discriminant vectors as $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_{C-1}\}$ and call the feature space spanned by \mathbf{W} a *discriminant feature space*. To obtain all the discriminant vectors \mathbf{W} , the following objective function has to be maximized:

$$J(\mathbf{W}) = \text{tr} \left(\mathbf{W}^T \mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{W} \right). \quad (5)$$

It is well known that \mathbf{W} is computed by solving the following generalized eigenvalue problem:

$$\mathbf{S}_B \mathbf{W} = \mathbf{S}_W \mathbf{W} \Lambda \quad (6)$$

where Λ is a diagonal matrix whose diagonal element λ_i is the i th eigenvector \mathbf{w}_i . For notational convenience, let us denote the calculated discriminant space model as $\Omega = \{\mathbf{W}, \Lambda, \bar{\mathbf{x}}_c, n_c, \mathbf{V}_c \mid c = 1, \dots, C\}$.

Assume that L training samples \mathbf{Y} are given and the number of classes increases from C to C' ($C \leq C' \leq C + L$). The new training samples are denoted as $\mathbf{Y} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_{C'}\}$ where $\mathbf{Y}_c = \{\mathbf{y}_{c1}, \dots, \mathbf{y}_{cl_c}\}$ ($c = 1, \dots, C'$). Here, \mathbf{y}_{cj} is the j th training sample of class c and l_c is the number of class c samples. Note that $L = \sum_c l_c$. Then, the between-class scatter matrix \mathbf{S}_B and the within-class scatter matrix \mathbf{S}_W are updated by the following equations [5]:

$$\mathbf{S}'_B = \sum_{c=1}^{C'} n'_c (\bar{\mathbf{x}}'_c - \bar{\mathbf{x}}') (\bar{\mathbf{x}}'_c - \bar{\mathbf{x}}')^T \quad (7)$$

$$\mathbf{S}'_W = \sum_{c=1}^{C'} \mathbf{V}'_c \quad (8)$$

where

$$\begin{aligned} \mathbf{V}'_c &= \mathbf{V}_c + \frac{l_c^2}{n_c'^2} (\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c) (\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c)^T + \frac{n_c^2}{n_c'^2} \sum_{j=1}^{l_c} (\mathbf{y}_{cj} - \bar{\mathbf{x}}_c) (\mathbf{y}_{cj} - \bar{\mathbf{x}}_c)^T \\ &\quad + \frac{l_c(l_c + 2n_c)}{n_c'^2} \sum_{j=1}^{l_c} (\mathbf{y}_{cj} - \bar{\mathbf{y}}_c) (\mathbf{y}_{cj} - \bar{\mathbf{y}}_c)^T \end{aligned} \quad (9)$$

$$\bar{\mathbf{x}}'_c = \frac{n_c}{n'_c} \bar{\mathbf{x}}_c + \frac{1}{n'_c} \sum_{j=1}^{l_c} \mathbf{y}_{cj} \quad (10)$$

$$\bar{\mathbf{x}}' = \frac{N}{N+L} \sum_{c=1}^{C'} n'_c \bar{\mathbf{x}}'_c. \quad (11)$$

Here, $n'_c = n_c + l_c$ and $n_c = 0$ for $c = C+1, \dots, C'$.

In the Pang et al.'s ILDA [5], first a new eigenvalue problem is defined by substituting Eqs. (7) and (8) into Eq. (6). Then, the new eigenvalue problem is solved to obtain the updated discriminant vectors $\mathbf{W}' = \{\mathbf{w}'_1, \dots, \mathbf{w}'_{C'-1}\}$ and the eigenvalues $\Lambda' = \{\lambda'_1, \dots, \lambda'_{C'-1}\}$. After all the components of the discriminant feature space model Ω are updated, the training samples \mathbf{Y} are discarded and only Ω is kept in the system. The above procedures are repeated whenever new training samples \mathbf{Y} are given.

3 ILDA for Multitask Pattern Recognition

3.1 Basic Idea

As explained in Section 2, discriminant vectors that span the feature space are obtained such that training samples of different classes are as linearly separable as possible. For this purpose, the class separability of discriminant features in Eq. (5) is maximized. However, the number of discriminant vectors in LDA is generally restricted to $(C - 1)$ where C is the number of classes. Thus, for a two-class problem, only a one-dimensional discriminant feature space is constructed, and the class information in the complementary subspace is not usually exploited.

In some cases, however, the information in the complementary subspace might be useful for classifying classes. To find such useful discriminant vectors, Xiang et al. [6] propose *recursive Fisher linear discriminant* (RFLD) algorithm that has been developed for single task and batch learning problems. In RFLD, extra discriminant vectors are recursively searched in the complementary subspace of the original discriminant feature space by using the ordinal LDA.

The idea of this paper comes from the following simple thought: If tasks are mutually related, there could exist some discriminant vectors of the previous tasks that are helpful for learning the current recognition task. As will be described later, the knowledge transfer in the proposed ILDA is carried out by searching for useful discriminant vectors in the complementary space. In this sense, we can view the proposed method as an online implementation of the Xiang's RFLD under the multitask learning environment defined next.

3.2 Multitask Learning Environment

Figure 1 illustrates an MTPR problem we assume here. Training samples of a specific task are given in a chunk for a certain period of time called *epoch* and discarded after training. Recognition tasks are switched one after another and

could appear repeatedly. Therefore, we assume *one-pass incremental learning* and *sequential task learning* as a learning environment. We further assume that training samples have both class and task information; thus, a task change is explicitly informed by the environment.

In the case of Fig. 1, a data chunk of Task 1 is given at the 1st learning stage and another chunk of Task 1 comes at the 2nd stage. After the 3rd chunk is trained, the task to be learned is changed to Task 2 with an explicit notice of a task change to the system. Five data chunks of Task 2 are given between the 4th and the 8th learning stages one after another. Then, the task is switched to Task 3 at the 9th stage. Note that Tasks 1 and 3 appear again at the 13th and the 19th stages.

3.3 Proposed ILDA Algorithm

Assume that $(R - 1)$ recognition tasks are already trained and the discriminant feature space models $\Omega^{(r)} = \{\mathbf{W}^{(r)}, \mathbf{A}^{(r)}, \bar{\mathbf{x}}_c^{(r)}, n_c^{(r)}, \mathbf{V}_c^{(r)} \mid c = 1, \dots, C^{(r)}\}$ ($r = 1, \dots, R - 1$) are stored in the long-term memory (LTM). Now assume that a new task is presented to the system and the conventional LDA is applied to the first chunk of training samples, each of which belongs to one of $C^{(R)}$ classes. By solving the eigenvalue problem in Eq. (6), the R th discriminant feature space model $\Omega^{(R)} = \{\mathbf{W}^{(R)}, \mathbf{A}^{(R)}, \bar{\mathbf{x}}_c^{(R)}, n_c^{(R)}, \mathbf{V}_c^{(R)} \mid c = 1, \dots, C^{(R)}\}$ is obtained.

The purpose of the knowledge transfer is to find effective discriminant vectors from $\mathbf{W}^{(r)}$ ($r = 1, \dots, R - 1$) that are computed in the past tasks. To augment the current discriminant feature space spanned by $\mathbf{W}^{(R)}$, extra discriminant vectors should be searched in the complementary subspace of $\mathbf{W}^{(R)}$. Hence, the discriminant vectors $\hat{\mathbf{W}}^{(r)}$ of the r th task are once transformed into the residue vectors $\hat{\mathbf{w}}_i^{(r)} = \{\hat{\mathbf{w}}_1^{(r)}, \dots, \hat{\mathbf{w}}_{C^{(r)}-1}^{(r)}\}$ ($r = 1, \dots, R - 1$) such that they are orthogonal to $\mathbf{W}^{(R)} = \{\mathbf{w}_1^{(R)}, \dots, \mathbf{w}_{C^{(R)}-1}^{(R)}\}$. The residue vectors $\hat{\mathbf{w}}_i^{(r)}$ ($i = 1, \dots, C^{(r)} - 1$) are calculated by

$$\hat{\mathbf{w}}_i^{(r)} = \frac{\hat{\mathbf{w}}_i^{(r)\prime}}{\|\hat{\mathbf{w}}_i^{(r)\prime}\|} \quad \text{where} \quad \hat{\mathbf{w}}_i^{(r)\prime} = \mathbf{w}_i^{(r)} - \mathbf{W}^{(R)T} \mathbf{w}_i^{(r)}. \quad (12)$$

Based on the principle of LDA, the effectiveness of a discriminant vector $\hat{\mathbf{w}}_i^{(r)}$ is evaluated by the class separability $J(\hat{\mathbf{w}}_i^{(r)})$ in Eq. (4). Hence, we propose the following condition to transfer a discriminant vector $\hat{\mathbf{w}}_i^{(r)}$ of the r th task:

$$J(\hat{\mathbf{w}}_i^{(r)}) > \eta \left\{ \min_k (J(\mathbf{w}_k^{(R)})) \right\} \quad (13)$$

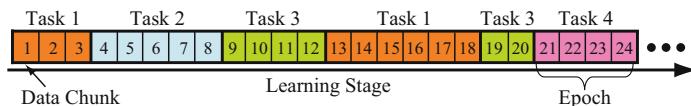


Fig. 1. Sequential presentation of tasks and training samples in an MTPR problem

where η is a positive constant between 0 and 1. This condition means that a discriminant vector of the past task is transferred only if it enhances the total class separability more than a certain value that is dynamically determined in proportion to the lowest class separability of the current discriminant vector. If this condition is satisfied, the i th discriminant vector $\hat{\mathbf{w}}_i^{(r)}$ is transferred to the current task, and it is combined with $\mathbf{W}^{(R)}$ as follows:

$$\mathbf{W}^{(R)'} = [\mathbf{W}^{(R)}, \hat{\mathbf{w}}_i^{(r)}]. \quad (14)$$

This knowledge transfer process is applied to all the discriminant vectors $\hat{\mathbf{w}}_i^{(r)}$ ($i = 1, \dots, C^{(r)} - 1$) of the r th task.

The above process is repeated for other discriminant vectors $\mathbf{W}^{(r')}$ ($r' = r, \dots, R - 1$). That is, the residue vectors of $\mathbf{W}^{(r')}$ are first calculated against the updated discriminant vectors $\mathbf{W}^{(R)'}$ in Eq. (14). Then, the condition in Eq. (13) is checked for each of the residue vectors, and it is transferred if satisfied.

On the other hand, the given task might be a known task. In this case, the corresponding discriminant space model is retrieved from LTM, and the Pang et al.'s ILDA is carried out to update the model.

4 Performance Evaluation

4.1 Experimental Setup

The performance of the proposed ILDA is evaluated for the four MTPR problems which are defined from the UCI data sets in Table 1. Table 2 shows the correspondence between the labels of the original data and those of the defined tasks for ‘Segmentation’ data. As seen from Table 2, the MTPR problem consists

Table 1. Evaluated UCI data sets

	#attrib.	#classes	#training	#test
Vehicle	18	4	188	658
Segmentation	18	7	210	2100
Yeast	8	10	744	740
Vowel-context	10	11	528	462

Table 2. An MTPR problem defined from ‘Segmentation’ data. In the tables, the correspondence between the class labels of original data and the three tasks is shown.

Original	1	2	3	4	5	6	7
Task 1	1	2	3	4	5	6	7
Task 2	8	8	8	9	9	9	9
Task 3	10	10	11	11	12	12	12

of three multi-class tasks. Let us explain how to generate the training samples of Task 2. Table 2 shows that Class 8 corresponds to Class 1, 2, and 3 in the original data set and Class 9 corresponds to Class 4, 5, 6, and 7. Hence, the training samples whose original classes are either Class 1, 2, or 3 are set to Class 8 in Task 2, and the same rule is applied to Class 9. We define MTPR problems for the other data sets in Table 1. However, due to space limitations, we omit the label correspondence tables.

In the following experiments, we assume that five training samples of a task are given in a chunk over the eight successive learning stages in an epoch; thus, forty samples are trained in the epoch of one task. Then, the task is changed to another task and another 40 training samples of the task are given in the same way. The same training sample is never given to the system once it is trained. However, the same task appears several times and different training samples of the task are given to the system at different epochs.

Since the performance could depend on the presentation order of tasks and training samples, we generate fifty different sequences per MTPR problem to evaluate the average performance. Two performance measures are used for the evaluation: class separability and recognition accuracy. To evaluate the classification accuracy, we adopt the nearest neighbor method as a classifier.

4.2 Experimental Results

To study the effectiveness of the knowledge transfer, the proposed ILDA is compared with the ILDA without the knowledge transfer. For notational simplicity, these models are denoted as “ILDA with KT” and “ILDA without KT.”

First, we examine how the class separability is improved by introducing the knowledge transfer. Figure 2 shows the time evolutions of the class separability obtained by ILDA with KT and ILDA without KT. The evaluated data set is ‘Vowel-context.’ The horizontal axis shows the learning stages and a chunk of five samples are trained at every stage. As seen from Fig. 2, the class separability of

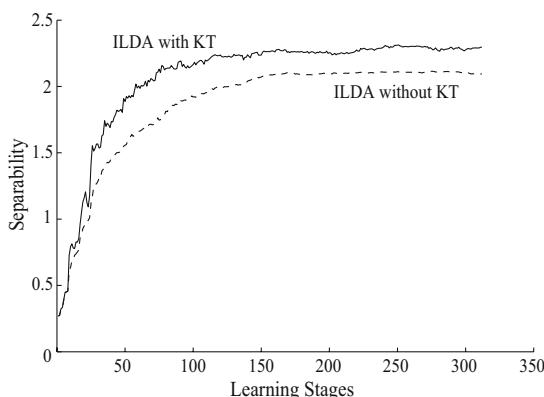


Fig. 2. Time evolutions of the class separability for ‘Vowel-context’

Table 3. Class separability of discriminant features at the final stage

	Vehicle	Segmentation	Yeast	Vowel
Without KT	2.18	5.76	1.51	2.10
With KT	2.33	12.0	1.60	2.30

Table 4. Final classification accuracies [%]

	Vehicle	Segmentation	Yeast	Vowel
Without KT	81.2±0.7	83.3±0.6	50.9±1.0	55.0±1.2
With KT	81.8±0.8	86.3±1.9	52.7±0.9	59.9±3.1

the proposed ILDA is enhanced by the knowledge transfer at the early learning stages, and this performance advantage is maintained throughout the learning stages. Table 3 shows the average separability at the final learning stage for the four MTPR problems. From the results, it is confirmed that the class separability of discriminant features is clearly improved by the knowledge transfer.

Finally, we evaluate the classification accuracy when extracted discriminant features are applied to the k -nearest neighbor method. Table 4 shows the classification accuracy at the final learning stage. In order to check the statistical significance of the difference in the average accuracies of the two ILDAs, we perform the Welch's t test. As a result, we confirm that the proposed ILDA outperforms the ILDA without KT for all the MTPR problems.

5 Conclusions

In this paper, we proposed an ILDA algorithm for MTPR problems in which training samples of a specific recognition task are given one after another and the task is switched to another related task in turn. In the proposed ILDA, a discriminant space model was updated not only by the Pang et al.'s ILDA but also by the knowledge transfer. The knowledge transfer was implemented by augmenting the discriminant space of the current task with effective discriminant vectors selected from other related tasks based on the class separability.

In the experiments, the proposed ILDA was evaluated for the four MTPR problems, each of which consisted of three multi-class recognition tasks. The results demonstrated that the proposed ILDA outperformed the ILDA without knowledge transfer with regard to both the class separability and recognition accuracy. From the experimental results, we conclude that the proposed knowledge transfer works well to build an effective discriminant space incrementally.

References

1. Caruana, R.: Multitask Learning. Machine Learning 28, 41–75 (1997)
2. Silver, D.L., Mercer, R.E.: Selective Function Transfer: Inductive Bias from Related Task. In: Proc. of IASTED Int. Conf. on AI and Soft Computing, pp. 182–189 (2001)

3. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-Task Feature Learning. In: Advances in Neural Information Processing Systems, vol. 19, pp. 41–48. MIT Press, Cambridge (2007)
4. Ozawa, S., Roy, A.: Incremental Learning for Multitask Pattern Recognition Problems. In: Proc. of 17th Int. Conf. on Machine Learning and Applications (in press)
5. Pang, S., Ozawa, S., Kasabov, N.: Incremental Linear Discriminant Analysis for Classification of Data Streams. IEEE Trans. on SMC - Part B 35(5), 905–914 (2005)
6. Xiang, C., Fan, X.A., Lee, T.H.: Face Recognition Using Recursive Fisher Linear Discriminant. IEEE Trans. on Image Processing 15(8), 2097–2105 (2006)

Soft Sensor Based on Adaptive Local Learning

Petr Kadlec and Bogdan Gabrys

Computational Intelligence Research Group, Smart Technology Research Centre,
Bournemouth University, Fern Barrow, Poole, BH12 5BB, United Kingdom
pkadlec@bournemouth.ac.uk

Abstract. When it comes to application of computational learning techniques in practical scenarios, like for example adaptive inferential control, it is often difficult to apply the state-of-the-art techniques in a straight forward manner and usually some effort has to be dedicated to tuning either the data, in a form of data pre-processing, or the modelling techniques, in form of optimal parameter search or modification of the training algorithm. In this work we present a robust approach to on-line predictive modelling which is focusing on dealing with challenges like noisy data, data outliers and in particular drifting data which are often present in industrial data sets. The approach is based on the local learning approach, where models of limited complexity focus on partitions of the input space and on an ensemble building technique which combines the predictions of the particular local models into the final predicted value. Furthermore, the technique provides the means for on-line adaptation and can thus be deployed in a dynamic environment which is demonstrated in this work in terms of an application of the presented approach to a raw industrial data set exhibiting drifting data, outliers, missing values and measurement noise.

1 Introduction

In this work we propose an adaptive regression model for on-line prediction. The predictor is applied in the process industry where this type of model is called data-driven *Soft Sensor* [1]. Data-driven Soft Sensors are trained using *historical process data*. For many industrial applications this data is automatically recorded and can be retrieved from the process databases. After the training, the models are launched in the real-life application where it is the task of the Soft Sensor to provide an on-line prediction of the target values. The on-line data stream is often unlabelled or the labels are delayed (e.g. due to manual evaluations in chemical laboratories). In the latter case the delayed labels can be used for the adaptation of the models.

The most common data-driven techniques applied for empirical Soft Sensor modelling are: (i) Principle Component Regression [2] and Partial Least Squares method [3], these techniques gained their popularity due to their statistical background, ease of interpretability of the model, and due to the fact that the methods intrinsically deal with data co-linearity which is a common problem among industrial data sets. Examples of Soft Sensor applications based on PCA/PLS are [4][5]; (ii) Artificial Neural Networks [6], this modelling technique is very popular in predictive modelling and due to its ability to model non-linear problems it has also found broad applications as Soft Sensor

modelling tool (see e.g. [7]); (iii) several other methods like Support Vector Machines [8] and Neuro-Fuzzy Systems [9] for on-line prediction [10][11].

This work is based on the local learning approach. The pioneering work of local learning was presented by Jacobs and Jordan in [12], where a gating network is used to decide which model from a set of available local experts is responsible for the prediction of the given input sample. Another approach related to our work is the Locally Weighted Learning (LWL) technique [13] which was proposed as an effective way of dealing with the bias/variance dilemma [14] and negative interference [15]. Based on LWL the Locally Weighted Projection Regression has been proposed in [16] where additionally a local dimensionality reduction has been applied in order to deal with high dimensional data with locally low dimensional manifold. Furthermore, the LWPR algorithm is restricted to the application of linear regression models as predictors. In contrast to LWPR the approach presented in this work provides higher flexibility by modelling the receptive fields using the Parzen method for distribution approximation [17] and allows to use any regression technique to be applied as local expert. Also, the receptive field descriptors are built in the reduced low dimensional space rather than in the original high dimensional input space.

2 Soft Sensor for On-Line Prediction

This section presents the main contribution of this work which is a Soft Sensor algorithm. The algorithm can be split into the following steps: (i) Receptive fields construction; (ii) Local models/experts training; (iii) Receptive field descriptors building (iv) Local experts combination; (v) Soft Sensor adaptation during the run-time. The first three steps are performed off-line using the historical data while the last two steps are carried out during the exploitation (run-time) phase and thus using the on-line data.

Receptive fields construction: The aim of the receptive field building is to divide the historical data into partitions representing the particular concepts within the historical data. The notion of the concepts is linked to the area where a model, called landmarker, provides constant performance.

Provided the historical data set $S^{hist} = (\mathcal{X}^{hist}, \mathcal{Y}^{hist})$, where S^{hist} consists of an $M \times N$ matrix of input data \mathcal{X}^{hist} , consisting of M variables and N observations, and a vector of N target labels \mathcal{Y}^{hist} , the first step of the algorithm is training the landmarker. The landmarker is trained using samples from an initial window S^{init} which is a subset of the historical data:

$$S_i^{init} = (\mathcal{X}_i^{init}, \mathcal{Y}_i^{init}) := \{(\mathbf{x}_n, y_n); n = k, \dots, k + l\}, \quad (1)$$

where i identifies the current receptive field, n is the index of the samples within the receptive field, k is the index of the first sample in the current receptive field, l is the length of initial window, \mathbf{x}_n is M-dimensional input sample and y_n is the corresponding target value.

Provided the initial set, the landmarker f_i^{lm} can be trained and the residual vector \mathbf{r}_i^{init} of the trained landmarker on the training data can be calculated:

$$\mathbf{r}_i^{init} = \mathcal{Y}_i^{init} - f_i^{lm}(\mathcal{X}_i^{init}), \quad (2)$$

The next step is shifting the window one step forwards, while keeping the size of window constant:

$$S^{shifted} = (\mathcal{X}_i^{shifted}, \mathcal{Y}_i^{shifted}) := \{(\mathbf{x}_n, y_n); n = k + s, \dots, k + s + l\} \quad s = 1, \quad (3)$$

and calculating the new residual values $\mathbf{r}_i^{shifted}(s)$ of landmarker's prediction using the new data window.

As next the two residual vectors (\mathbf{r}_i^{init} and $\mathbf{r}_i^{shifted}(s)$) are tested for a statistically significant difference using the one-sided t-test [18]. This significance test was chosen because the residuals can be, ideally, assumed as normally distributed. The null hypothesis is that there is no significant difference in the mean values of the two samples which in this case means that there is no significant difference between the landmarker's performance on the initial window and on the shifted window, i.e. the data from S^{init} and $S^{shifted}$ belong to the same concept. This procedure is repeated as long as the null hypothesis of the significance test remains valid:

$$s^{final} = \operatorname{argmin}_{s \in [1, \dots, N-k]} (ttest(\mathbf{r}_i^{init}, \mathbf{r}_i^{shifted}(s)) == 1), \quad (4)$$

where s^{final} corresponds to the first sample for which the t-test rejects the null hypothesis.

Finally, the receptive field is constructed in the following way:

$$S_i^{RF} = (\mathcal{X}_i^{RF}, \mathcal{Y}_i^{RF}) := \{(\mathbf{x}_n, y_n); n = k, \dots, k + l + s^{final} - 1\} \quad (5)$$

and the algorithm can move to the next receptive field by constructing new initial window S_{i+1}^{init} .

The outcome of this stage is a set of receptive fields \mathcal{S}^{RF} , each corresponding to a concept of the historical data.

Local experts training: After identifying of the receptive fields, a local expert f^{le} can be trained for each of the receptive fields. At this point one can apply any computational learning technique for the local experts with the same or higher than the landmarker's model complexity. This constrain originates from the fact that the notion of the concept is related to the landmarker's prediction capabilities and thus in order to be able to model the concept using the local expert it has to posses the same or better prediction ability.

After this step there is a set of trained local experts $\mathcal{F}^{le} := \{(f_i^{le}); i = 1, \dots, I\}$.

Receptive field descriptors building: The next step towards the final model is building descriptors of the receptive fields which will be later sampled to estimate the correspondence of the test samples to the particular receptive fields.

The aim of the descriptors is to describe the area of expertise of the particular local experts. This task is approached by building a set of two-dimensional descriptors, which are a combination of the input variables distribution $D(\mathcal{X}_i^m)$ and of the distribution of the target variable $D(\mathcal{Y}_i)$ within the i th receptive field. This leads to a two-dimensional probability density function $D_i^m(\mathcal{X}_i^m, \mathcal{Y}_i)$. The mutual descriptor $D_i^m(\mathcal{X}_i^m, \mathcal{Y}_i)$ is constructed using a two-dimensional Parzen method [17]. The final descriptor $\mathcal{D} := \{D_i^m\}$

is a set of $M \times I$ two-dimensional pdfs, with M being the number of input variables and I the number of receptive fields.

Local experts combination: During the run-time (on-line) phase the task of the Soft Sensor is to make an on-line prediction of the target variable given the unlabelled test samples. To obtain the final prediction y^{final} these predictions have to be combined. This is done by making a weighted sum of the local experts' predictions:

$$y^{final} = \sum_{i=1}^I v_i(\mathbf{x}, f_i^{le}) f_i^{le}(\mathbf{x}), \quad (6)$$

where $v_i(\mathbf{x}, f_i^{le})$ is the weight of the i th local expert's prediction and $f_i^{le}(\mathbf{x})$ is the actual prediction given the input vector \mathbf{x} . In the case of the presented approach the purpose of the weights is to predict the performance of the local expert given the input sample and the local expert's prediction and thus to select the correct local expert for making the prediction. This can be expressed as the posterior probability of the i th receptive field given the test sample \mathbf{x} and the local expert prediction $f_i^{le}(\mathbf{x})$:

$$v_i(\mathbf{x}, f_i^{le}) = p(i|\mathbf{x}, f_i^{le}) = \frac{p(\mathbf{x}, f_i^{le}|i)p(i)}{p(\mathbf{x}),, f_i^{le}}, \quad (7)$$

where $p(i)$ is the a priori probability of the i th receptive field, $p(\mathbf{x}, f_i^{le})$ is the normalisation factor to normalise $p(i|\mathbf{x}, f_i^{le})$ and $p(\mathbf{x}|i)$ the likelihood of \mathbf{x} given the receptive field and f_i^{le} which can be calculated by reading the mutual pdfs at the positions defined by the sample \mathbf{x} :

$$p(\mathbf{x}, f_i^{le}|i) = \prod_{m=1}^M p(x^m, f_i^{le}|i) = \prod_{m=1}^M D_i^m(x^m, f_i^{le}(\mathbf{x}))p(m), \quad (8)$$

with $p(m)$ as a priori probability for the different variables calculated as its entropy:

$$p(m) = - \sum_{m=1}^M D_i^m(x^m, f_i^{le}) \log(D_i^m(x^m, f_i^{le})), \quad (9)$$

in this way the discriminant power of the variables is measured and used as a weighting factor.

Soft Sensor Adaptation: The adaptation is performed at the level of the combination of the local experts while the actual local experts are kept untouched. The adaptation is done by modifying the receptive field descriptors.

The descriptors are modified each time a correct target value y_n is received. Having the correct target value, the error e_i of the i th local expert's prediction f_i^{le} can be calculated and normalised:

$$e_i = (y_n - f_i^{le}(\mathbf{x}_n))^2, e_i^{rel} = \frac{e_i}{\sum_{i=1}^I e_i}. \quad (10)$$

The relative error values are further on applied as weighting factor for the adaptation masks ΔD_i^m . These are two-dimensional Gaussian functions which are used to adapt the neighbourhood of the sampling point $(x_n^m, f_i^{le}(\mathbf{x}_n))$ of the descriptors D_i^m :

$$\Delta D_i^m(x, y) = \alpha e_i^{rel} ((\Delta D_i^m(x))^T * \Delta D_i^m(y)) \quad (11a)$$

$$\Delta D_i^m(x) = \frac{1}{h^x \sqrt{2\pi}} \exp\left(-\frac{(\mathbf{x}^{range} - x_n^m)^2}{2(h^x)^2}\right) \quad (11b)$$

$$\Delta D_i^m(y) = \frac{1}{h^y \sqrt{2\pi}} \exp\left(-\frac{(\mathbf{y}^{range} - y_n)^2}{2(h^y)^2}\right), \quad (11c)$$

where $\Delta D_i^m(x, y)$ is a two-dimensional adaptation mask for the m variable and i th receptive field, $\Delta D_i^m(x)$ and $\Delta D_i^m(y)$ are the marginal masks for the m th input variable and the target variable respectively, α is the adaptation strength parameter, which can be used for modifying the power of the adaptation, further on h^x is the variance, i.e. the width, of the input variable Gaussian kernel and h^y the variance of the target variable kernel respectively, these two values define the size of the neighbourhood of the current sampling point which is being modified by the adaptation mask ΔD_i^m . This parameter influences the adaptation properties and can be tuned in order to obtain optimal adaptation capability, and \mathbf{x}^{range} and \mathbf{y}^{range} are the ranges within which the descriptor D_i^m stores the distributions.

Finally, the descriptors can be adapted using the modification masks in the following way:

$$D_i^m(t+1) = D_i^m(t) \cdot \Delta D_i^m, \quad (12)$$

where \cdot is the Hadamard product.

3 Experiments

In this section the proposed adaptive Soft Sensor will be evaluated in terms of an industrial data set. The raw data set provided by Evonik Industries consists of 1066 raw samples (covering six months of the operation of the process) each having 19 input features, like temperatures and pressures measured within the process plant. The target values of this data set are laboratory measurements of the residual humidity of the process product. The data exhibit typical problems of industrial data like missing values, outliers, measurements noise.

The available data is split into two parts. The first 30% of the samples form the historical data which is used for training of the three models. The remaining 70% of the data simulate the on-line data and are used as on-line testing data. This split was chosen because the focus of this work is on the assessment of the adaptivity capabilities of the Soft Sensor. All the tests presented in this section are strictly following the out-of-sample principle, i.e. the models are tested on unseen test samples.

Based on the analysis of the models' sensitivity to the input parameter(s), following values of the input parameters for the non-adaptive/adaptive local Soft Sensor have been selected: initial window size $l = 50$ (see Eq. 10), adaptation rate $\alpha = 1000$ (see Eq. 11a). The applied landmarks are multiple linear regression models and the local expert are

Table 1. MSE and correlation coefficient performance of the three tested approaches

	Soft Sensor type	MSE	Corr. coeff.
non-adaptive	MLP Committee	0.0062	0.40
	Proposed local learning non-adaptive	0.0053	0.41
adaptive	LWPR	0.0042	0.48
	Proposed local learning adaptive	0.0040	0.50

committee of ten multiple regression models combined using the LMS method. Prior to the training of the local experts the data are locally pre-processed using the Principle Component Analysis (PCA) with eight principle components (the optimal number of principle components was obtained using the cross-validation approach). There are two version of the proposed models discussed, a non-adaptive version without any adaptation capabilities and an adaptive version based on the adaptation of the receptive field descriptors.

Additionally for performance comparison, the non-adaptive version of the proposed model is compared to a traditional Soft Sensor, namely an Multi-Layer Perceptron ANN. The MLP Soft Sensor is a committee of ten randomly initialised models each having five hidden units in one hidden layer. These parameters were found as optimal in terms of Mean Squared Error and correlation coefficient and were also obtained using cross-validation approach.

Additionally, the adaptive Soft Sensor is compared to the Locally Weighted Projection Regression (LWPR) which is an established adaptive modelling technique showing some similarities to the proposed algorithm. The parameters of the LWPR method were optimised for this modelling task and the optimal performance was achieved with following values: init-D=4; init-alpha=100; diag-only=0; w-gen=0.9; w-prune=0.9; penalty=1e-7; meta=0; update-D=0; kernel=Gaussian.

Experiment results: The local learning Soft Sensor partitions the historical, i.e. training, data into four receptive fields and builds four local experts. The results of the experiments are presented in Fig. I and in Tab. I. In order to provide a fair comparison, Fig. I(a) firstly presents the predictions of the non-adaptive models (MLP and non-adaptive local learning approach) and separately to this the adaptive Soft Sensor and the LWPR predictions. In the second part of Fig. I(a), the adaptive version of the proposed Soft Sensor, as expected, further improves the performance and it follows the target values more accurately. Additionally it performs slightly better than the LWPR model.

The previous conclusions are more obvious in Fig. I(b) which shows the predictions residuals of the four Soft Sensors after smoothing with a 30 samples long averaging filter. In ideal case the curves should get as close to the zero-line as possible. Comparing the two non-adaptive models, one can see that the local learning Soft Sensor (black-line) gets much closer to this goal than the MLP committee. The benefit of the adaptation technique proposed in this work can be also observed, i.e. the residuals of the local learning adaptive Soft Sensor are the closest to the zero-line.

In summary Table I shows the Mean Squared Errors (MSE) and the correlation coefficients of the four Soft Sensors. The presented values confirm the previous results. One

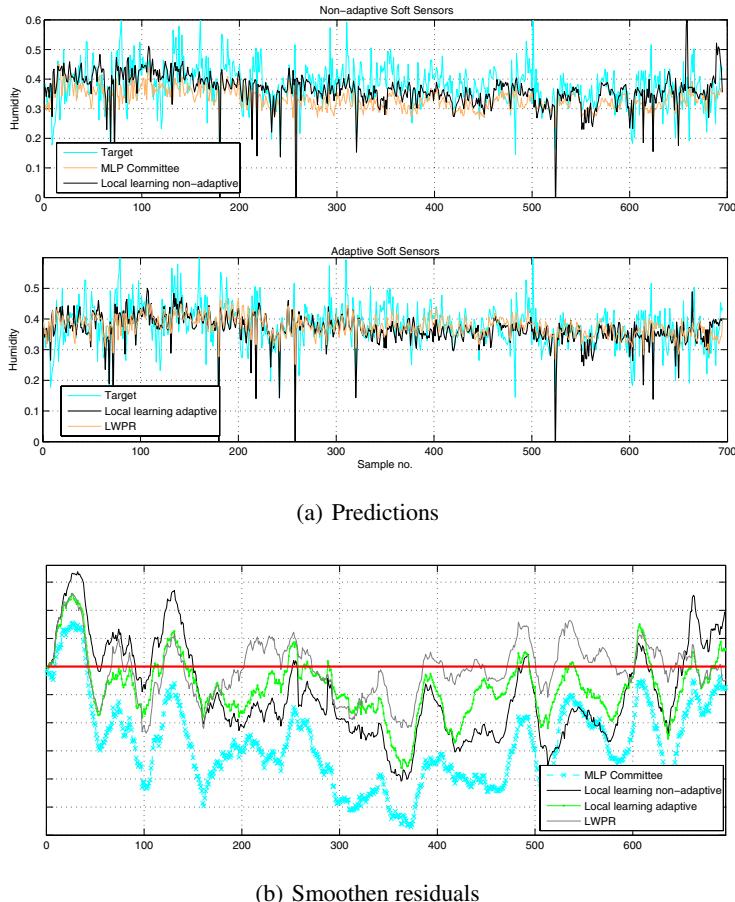


Fig. 1. Experimental results of the discussed four Soft Sensors

can see a slight gain in MSE of the non-adaptive local learning model in comparison to the MLP which is achieved at lower model complexity at the same time. Further on the benefit of the two adaptive models can be observed. The best performance is achieved by the adaptive version of the proposed algorithm. Despite some similarities between the our algorithm and the LWPR, our model shows slightly better performance for this modelling problem.

4 Conclusions

The main contribution of this work is an adaptive Soft Sensor which is based on the local learning approach. Unlike the majority of so far presented approaches to soft sensing, this algorithm also limits the effort which needs to be spent on the data pre-processing, model selection and Soft Sensor maintenance. The robustness of the Soft Sensor is

achieved by deploying a set of local models. The partitioning of the input space further allows individual pre-processing and modelling of the receptive fields depending on the local complexity. The limitation of the maintenance requirements is achieved by applying a fully incremental adaptive approach. This approach is based on the modification of the combination weights during the run-time of the Soft Sensor and thus does not require the implementation of any model-specific adaptation technique. This factor further increases the flexibility of the model as any regression technique can be applied as local expert. The application possibilities of the proposed Soft Sensor are wide, it can be for example deployed to support the process control in form of an adaptive inferential control.

References

1. Fortuna, L.: *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer, Heidelberg
2. Jolliffe, I.T.: *Principal Component Analysis*. Springer, Heidelberg
3. Abdi, H.: Partial least squares (pls) regression. *Encyclopedia of Social Sciences, Research Methods*. Sage, Thousand Oaks (2003)
4. Dong, D., McAvoy, T.J., Chang, L.J.: Emission monitoring using multivariate soft sensors. In: *Proceedings of the American Control Conference*, vol. 1 (1995)
5. Lin, B., Recke, B., Knudsen, J., Jrgensen, S.B.: A systematic approach for soft sensor development. *Computers and Chemical Engineering* 31(5) (2007)
6. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, USA
7. Fortuna, L., Graziani, S., Xibilia, M.G.: Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Engineering Practice* 13(4) (2005)
8. Vapnik, V.N.: *Statistical learning theory*. Wiley, New York
9. Jang, J.S.R., Sun, C.T., Mizutani, E.: *Neuro-fuzzy and soft computing*. Prentice-Hall, Upper Saddle River, NJ
10. Warne, K., Prasad, G., Siddique, N.H., Maguire, L.P.: Development of a hybrid pca-anfis measurement system for monitoring product quality in the coating industry. In: *2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4 (2004)
11. Desai, K., Badhe, Y., Tambe, S.S., Kulkarni, B.D.: Soft-sensor development for fed-batch bioreactors using support vector regression. *Biochemical Engineering Journal* 27(3) (2006)
12. Jacobs, R.: Adaptive mixtures of local experts. *Neural Computation* 3(1) (1991)
13. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. *Artificial Intelligence Review* 11(1) (1997)
14. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. *Neural Computation* 4(1) (1992)
15. French, R.: Catastrophic forgetting in connectionist networks: Causes, consequences and solutions. *Trends in Cognitive Sciences* 3(4) (1999)
16. Vijayakumar, S., D'Souza, A., Schaal, S.: Incremental online learning in high dimensions. *Neural Computation* 17(12) (2005)
17. Parzen, E.: On estimation of a probability density function and mode. *Annals of Mathematical Statistics* 33(3) (1962)
18. Gosset, W.S.: The probable error of a mean. *Biometrika* 6(1) (1908)

Directly Optimizing Topology-Preserving Maps with Evolutionary Algorithms

José Everardo B. Maia¹, André L.V. Coelho², and Guilherme A. Barreto³

¹ Department of Statistics and Computing, State Univ. of Ceará

² Graduate Program in Applied Informatics, Univ. of Fortaleza

³ Department of Teleinformatics Engineering, Federal Univ. of Ceará

jmaia@uece.br, acoelho@unifor.br, guilherme@deti.ufc.br

Abstract. Recently, the formation of topographic maps has been approached via a direct optimization strategy involving the use of heuristic search techniques. In this paper, we move a step further in this line of research by devising and empirically assessing the performance of six different evolutionary algorithms (EA) towards the automatic generation of high-quality maps. Besides, we also report an analysis over the convexity profiles exhibited by different realizations of the adopted cost function in a manner as to testify its inherent search complexity. The simulation results reveal that, although the EA schemes do not distinguish so much in terms of the average quality of the maps they form, there is sometimes a significant difference of performance in terms of robustness (variance of the quality indices) and efficiency (number of iterations to converge to a good solution).

1 Introduction

Roughly speaking, topographic maps (a.k.a self-organizing feature maps) are biologically-inspired data representations which combine the aspects of vector quantization and topological properties preservation. Topographic map (TM) formation is an important component of both artificial and biological networks and, in what concerns its theory, several approaches have already been proposed in the literature, among which we can mention those based on Markovian models [1], Information Theory [6], kernel methods [2], constrained optimization techniques [3], and the Expectation-Maximization algorithm [4,5]. In this context, one distinctive model is the Self-organizing Map (SOM) [8], which has become one of the most influential neural networks with a number of important applications in pattern recognition and machine learning, specially as a tool for data visualization and dimensionality reduction.

Some studies on TM contemplate their formation as an optimization problem. The main problem with this approach is that it requires a differentiable cost function, something not always feasible to achieve. For instance, it has been shown that a cost function for the original SOM algorithm would be highly discontinuous and with many local minima [3]. In this case, a natural alternative would be to develop topographic maps by means of heuristic search techniques

like evolutionary algorithms (EA) [2], an approach that, in our point of view, has not been fully explored yet.

In this context, it is worth mentioning that the issue of map formation using genetic algorithms (GA) has already been addressed in a few studies [11,7,10]. In [7], map formation is achieved via a two-step procedure. In the first step, the authors apply the K -means algorithm for cluster formation, while in the second step, with the weight vectors frozen, they use a GA to build a topologically-ordered map. Thus, according to this approach, weight adaptation is separated from weight ordering. To the best of our knowledge, Curry and Morgan [4] pioneered in proposing a single-step GA-based direct optimization approach for TM formation, tuning the weights in concert with the map ordering.

In a previous paper [9], we have also tackled the TM formation problem by direct optimization via a GA. Through computer simulations, we have experimentally shown that the direct optimization process can lead to topology-preserving mappings, which also display robustness to outliers. In that study, the resulting maps were assessed taking as reference those generated by the SOM algorithm.

In the current paper, we report on a follow-up work in which six EA schemes, with components recruited from genetic algorithm and evolution strategies (ES) dialects [2], are empirically assessed and have their performance compared in terms of effectiveness and efficiency criteria when coping with the automatic generation of topographic maps. In this respect, our purpose here is to enlarge the scope of investigation of EA-based solutions to TM formation by varying some of the EA components and also incorporating more customized ones.

The remainder of the paper is organized as follows. In Section 2 we introduce the proposed EA-based approach to SOFM formation, describing in detail the components that compose the six evolutionary schemes investigated here. In Section 3, we present the obtained results and discuss them accordingly. The paper is concluded in Section 4.

2 Topographic Map Formation via Evolutionary Algorithms

Evolutionary algorithms comprehend a family of stochastic search and optimization algorithms inspired from the mechanics of Natural Selection and concepts of population genetics [2]. According to the general EA framework, candidate solutions to a given problem play the role of individuals in a population, while a fitness function determines the environment within which the solutions “live” and have their levels of adaptation (quality) measured. Here, optimal solutions emerge through the evolution of the population, which takes place after the repeated application of some operators mimicking well-known natural phenomena: parent (mating) selection, recombination, mutation, and survivor (environmental) selection.

The most significant advantages of using evolutionary algorithms lie in the gain of flexibility and adaptability to the task at hand, in combination with robust performance and global search characteristics. The key to successful implementation of EA most often lies in the choice of a representation and a fitness

function that mirror the problem well and the use of ‘intelligent’ genetic operators, which, again, have a natural affinity to the problem of interest [2]. In the sequel, we discuss about the properties of the fitness function devised to cope with the TM formation problem and then provide details of the other EA components that compose the EA schemes evaluated in Section 3.

2.1 The Fitness Function

To measure the quality of the formed map, Kirk and Zurada [7] used the Weighted Topological Error (WTE) index, defined as

$$WTE = \frac{1}{L} \sum_{l=1}^L \frac{|i_1(l) - i_2(l)| - 1}{N - 1}, \quad (1)$$

where $i_1(l)$ and $i_2(l)$ are the indices of the neurons whose weight vectors are, respectively, the first and second closest ones to the l -th input vector. The authors compared the WTE values obtained by their GA-based approach with those obtained by Kohonen’s SOM, the former showing clear advantages over the latter. However, their results were achieved solely for the one-dimensional SOM.

Conversely, Curry and Morgan [1] used as fitness function (FF) the Locally Weighted Distortion Index (LWDI):

$$LWDI = E \left[\sum_{\forall j} h_{lj} \| \mathbf{x}_l - \mathbf{w}_j \| \right], \quad (2)$$

where $E[\cdot]$ denotes the expectation operator taken over the whole set of input vectors. A comparison between their GA-based approach and Kohonen’s SOM in terms of the obtained LWDI values favored the former.

Another choice for assessing how good is a topographic map formed is to use the Quantization Error (QE) and the Topographic Error (TE) indices, defined respectively as

$$QE = \frac{1}{L} \sum_{l=1}^L \| \mathbf{x}_l - \mathbf{w}_j \|, \quad (3)$$

and

$$TE = \frac{1}{L} \sum_{l=1}^L \Psi(\mathbf{x}_l, \mathbf{w}_1, \dots, \mathbf{w}_N), \quad (4)$$

where $\Psi(\mathbf{x}_l, \mathbf{w}_1, \dots, \mathbf{w}_N) \in [0, 1]$.

The QE index assesses how good is the TM as a vector quantizer, whereas the TE index captures the proportion of data points for which the closest and second-closest weight vectors are not adjacent on the map grid, which is determined by Ψ . The relative importance of these two indices depends on the application in sight and it is very difficult to manage using Kohonen’s SOM algorithm.

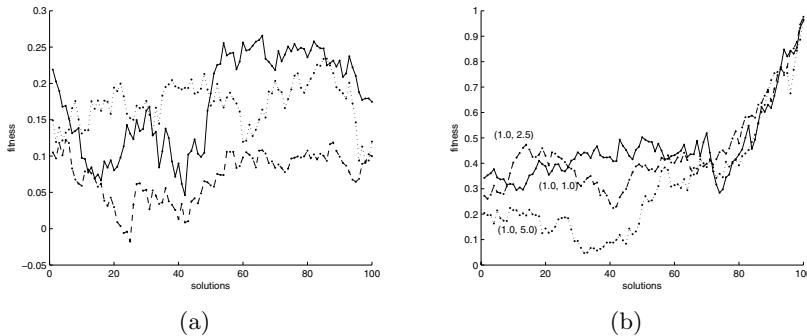


Fig. 1. (a) Convexity analysis of the fitness landscape for a particular realization of the fitness function ($\alpha = \beta = 1$) considering distinct mutation step sizes. (b) Convexity analysis of the fitness landscape considering distinct combinations of (α, β) in Eq. (5).

In the experiments reported here, we have adopted a fitness function formed by the simple combination of the QE and the Pearson correlation coefficient (PCC) indices, which is given by

$$\text{Fitness}(\widetilde{\mathbf{W}}) = \alpha \cdot \text{PCC}(\widetilde{\mathbf{W}}) - \beta \cdot \text{QE}(\widetilde{\mathbf{W}}), \quad (5)$$

where $\widetilde{\mathbf{W}}$ denotes the whole set of weight vectors, and the parameters $\alpha, \beta \in [0, 10]$ weigh the relative importance of the indices with respect to each other. The PCC index is computed as the cross-correlation between pairs of distances $[d(\mathbf{r}_m, \mathbf{r}_n), d(\mathbf{w}_m, \mathbf{w}_n)]$, where $(\mathbf{r}_m, \mathbf{r}_n)$ are the coordinates of pairs of neurons in the output array and $(\mathbf{w}_m, \mathbf{w}_n)$ are the corresponding pairs of weight vectors. It is worth noting that PCC is an index of the type *the larger, the better*, while QE is of the type *the lesser, the better*.

Before delving into details about the other EA components, it is worth analyzing graphically the convexity properties underlying the fitness landscape of a typical realization of Eq. (5). For this purpose, we provide in Fig. 1(a) three different trajectories obtained by resorting to different mutation operators, which vary in terms of magnitude of the step size and scope of application. On the other hand, with Fig. 1(b), we perform a different analysis considering different realizations (i.e., combinations of α and β) of the fitness function in Eq. (5). Starting from different random map configurations and by applying mutation as the unique source of variation, the three trajectories eventually converge up to a well-formed map. The shape of the trajectories pictured in both figures evidence the hurdles inherent to the direct optimization of the conceived fitness function. It is quite noticeable the high levels of multi-modality (local minima) and nonlinearities involved in the underlying search landscapes, something that hampers too much the direct application of gradient-based optimization methods.

2.2 Representation and Genetic Operators

In the EA schemes considered in this study, each individual is represented as a $N \times P$ matrix of real numbers, where P denotes the number of weight (prototype) vectors in a map and N denotes the input dimension.

Regarding recombination, two crossover operators were conceived, taking the topological-preservation issue into account. The first, referred to as point-based crossover (PbCross), randomly chooses a position in the map grid and then swaps the corresponding prototype vectors of the two parents selected for mating. On the other hand, the second recombination operator (NbCross) centers upon the concept of “neighborhood” among prototypes. During its application, not only the corresponding prototypes are swapped as in PbCross, but also all the neighbors of these prototypes lying within a vicinity of radius r . (In the experiments reported here, $r = 1$ was adopted.) By observing Fig. 2(a), one can visualize three different possibilities for recombining blocks of prototypes as performed by the NbCross operator over a bi-dimensional grid.

In what concerns mutation, its scope of application is at the prototype level. That is, for each weight vector Y pertaining to an EA individual, a random number (taken from a uniform distribution) is generated. If its magnitude is below p_m then a random weight vector X , extracted from a Gaussian distribution with mean zero and covariance matrix C (i.e. $X \sim N(0, C)$), is added to the target weight vector Y . Otherwise, Y is not altered. Although we have also considered the application of mutation at a prototype component level and with a uniform distribution for the mutation vector X , the EA configurations discussed in the next section do not contemplate these options. Besides, it is worth mentioning that, in two EA schemes, the strategy of deterministically controlling the magnitude of the mutation rate and step size (σ) along the evolutionary process has also been adopted [2]. The schedule imposed was to start with high values for p_m and σ and then decreasing them gradually.

2.3 Mating and Survivor Selection

In the selection of individuals for reproduction, three types of strategies have been employed: deterministic (in which all individuals are selected), stochastic (in which individuals are randomly selected), and fitness-proportionate (in which fitter individuals have higher chance to be selected). Conversely, with respect to survivor selection, we have investigated three possibilities. The first involves the generational model by means of which the whole population is replaced by its offspring. The other two relate to the replacement strategies usually employed in ES, namely, $(\mu + \lambda)$, where the survivors are deterministically selected from the set composed of parents and offspring, and (μ, λ) , where the survivors are selected exclusively from the offspring (in this case, $\lambda \gg \mu$).

3 Results

The basic components described in Section 2 were systematically combined in order to give birth to different EA schemes towards the formation of high-quality

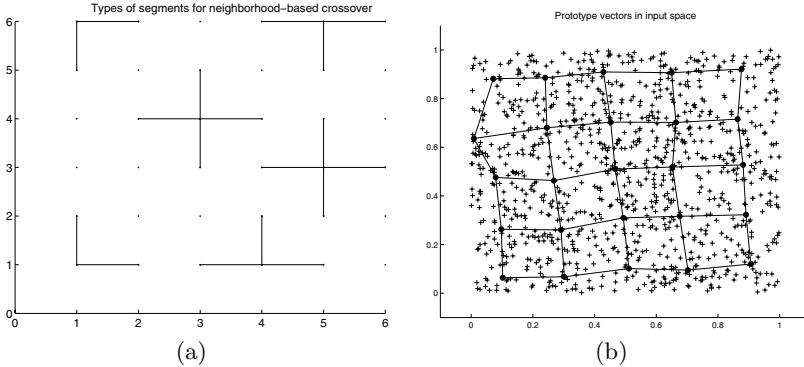


Fig. 2. (a) Three types of recombination of blocks or prototypes as performed by the NbCross operator. (b) A high-quality 5×5 map formed by an EA ($\alpha = \beta = 1$).

maps. Although we have been experimenting with several EA variants, we focus the analysis here only on six of them, whose configurations are conveyed in Table II. In all, the same fitness function, given by Eq. (5), was adopted, although sometimes varying the coefficients α and β . Likewise, the same stop criteria were used: reaching the maximum number of generations (2,500 in most of the cases) or reaching a solution with a fitness value greater than or equal to 0.9 (for instance, for a 5×5 grid, this mark would represent a map having 0.98 as PCC and 0.08 as QE). While some schemes are more compliant with a GA standpoint, others much resemble an evolution strategy. This can be easily inspected by considering the different emphasis given on the mutation and crossover operators. Indeed, the first EA refers to the GA adopted in the experiments reported in [9]. As mentioned before, two of the schemes were endowed with an automatic control over the mutation rate and step size.

Table 1. Component configurations of the different EA schemes

Component	EA1	EA2	EA3	EA4	EA5	EA6
Representation	matrix	matrix	matrix	matrix	matrix	matrix
Fitness Function	Eq. (5)	Eq. (5)	Eq. (5)	Eq. (5)	Eq. (5)	Eq. (5)
Recombination ($p_c = 0.01$)	NbCross	NbCross	NbCross	—	PbCross	PbCross
Mutation $p_m(\sigma)$	0.01 (0.05)	0.30 (0.1)	0.3 (0.1)	0.15 (0.4)	0.01 (0.1)	0.15 (0.4)
Mutation rate and step size control	no	no	no	yes	yes	no
Mating Sel.	fit-propriet.	determ.	determ.	random	fit-propriet.	determ.
Survivor Sel.	generat.	$(\mu + \lambda)$	(μ, λ)	$(\mu + \lambda)$	$(\mu + \lambda)$	generat.
Population size	40	80	80	80	80	40

Table 2. Average and standard deviation values of QE, WTE, and NG for a 5×5 map ($\alpha = 1$)

Algorithm -	QE	WTE	NG
EA1 ($\beta = 2.0$)	av 0.0765 sd 0.0024	0.012 0.010	1,517.10 217.64
EA2 ($\beta = 4.0$)	av 0.0821 sd 0.0044	0.021 0.034	1,345.08 220.20
EA3 ($\beta = 4.0$)	av 0.0863 sd 0.0025	0.015 0.014	1,980.45 214.32
EA4 ($\beta = 2.5$)	av 0.0844 sd 0.0029	0.011 0.012	1,252.34 109.60
EA5 ($\beta = 2.5$)	av 0.0868 sd 0.0019	0.012 0.016	2,317.10 417.64
EA6 ($\beta = 2.0$)	av 0.0794 sd 0.0026	0.012 0.011	1,817.10 201.64
SOM	av 0.0745	0.015	-
-	sd 0.0016	0.010	-
EA1 ($\beta = 6.5$)	av 0.0806 sd 0.0045	0.558 0.073	3,491.34 15.55
EA2 ($\beta = 8.0$)	av 0.0781 sd 0.0096	0.645 0.034	3,414.82 36.00

Table 2 contrasts the performance levels achieved by the six EA, having QE, WTE, and NG (i.e. the number of generations elapsed until locating the optimal map for the first time) as figures of merit. In this table, we also provide the values achieved by SOM to serve as yardstick. Several runs have been performed varying the metaparameter values as well as the values of α and β in Eq. (5). The particular values shown in the table (except given to the two last cases) represent configurations for which high-quality maps could be formed regularly. The two configurations at the bottom represent aberrant cases wherein high values of β deeply precluded the formation of topologically-conserving maps by the EA.

The results conveyed in Table 2 indicate that there is no significant difference between the WTE values presented by the maps formed by all six contestants. Conversely, considering average QE as the only figure of merit, EA1 and EA6 schemes have outperformed the others, although the latter has displayed higher values for NG (lower efficiency). If efficiency is a issue, EA3 and EA5 are not good options, with EA4 prevailing significantly. Figure 2(b) illustrates one of the high-quality maps formed by the EA schemes considered.

4 Final Remarks

In this paper, we have devised and empirically assessed the performance of six different evolutionary algorithms in order to tackle the automatic generation of high-quality topographic maps via direct optimization. All EA schemes share

the same fitness function, which is a weighted combination of indices that measure the aspects of vector quantization and topological preservation, but vary in terms of the other components. The simulation results reveal that, although the different EA schemes do not distinguish so much in terms of the average quality of the maps they form, there is a significant difference of their performance in terms of robustness (variance of the quality indices) and efficiency (number of iterations to converge to a good solution). Moreover, since the different EA provide alternative tradeoffs to the quality indices under investigation, there is a range of possibilities for their adoption according to the necessities of the application scenario in sight. As future work, we plan to explore other types of combination of quality indices to compose the fitness function. The handling of the TM formation problem through an evolutionary multiobjective perspective seems also to be a very promising initiative and is currently under investigation.

References

- Curry, B., Morgan, P.H.: Evaluating Kohonen's learning rule: An approach through genetic algorithms. *European Journal of Operational Research* 154(1), 191–205 (2004)
- Eiben, A.E., Smith, J.E.: *Introduction to Evolutionary Computing*. Springer, Heidelberg (2003)
- Erwin, E., Obermayer, K., Schulten, K.: Self-organizing maps: ordering, convergence properties and energy functions. *Biological Cybernetics* 67(1), 47–55 (1992)
- Graepel, T., Burger, M.M., Obermayer, K.: Self-organizing maps: Generalizations and new optimization techniques. *Neurocomputing* 21, 173–190 (1998)
- Heskes, T.: Self-organizing maps, vector quantization and mixture modeling. *IEEE Transactions on Neural Networks* 12(6), 1299–1305 (2001)
- Kamimura, R.: Cooperative information maximization with gaussian activation functions for self-organizing maps. *IEEE Transactions on Neural Networks* 17(4), 909–918 (2006)
- Kirk, J.S., Zurada, J.M.: Motivation for a genetically-trained topography-preserving map. In: *Proceedings of the 2002 International Joint Conference on neural Networks (IJCNN 2002)*, vol. 1, pp. 394–399 (2002)
- Kohonen, T.: *Self-Organizing Maps*. Springer, Heidelberg (2001)
- Maia, J.E.B., Barreto, G.A., Coelho, A.L.V.: On self-organizing feature map (SOFM) formation by direct optimization through a genetic algorithm. In: *Procs. of 8th International Conference on Hybrid Intelligent Systems (HIS 2008)*, Barcelona, Spain, September 2008, vol. 1, pp. 661–666. IEEE Computer Society Press, Los Alamitos (2008)
- Polani, D.: On the optimization of self-organizing maps by genetic algorithms. In: Oja, E., Kaski, S. (eds.) *Kohonen Maps*, pp. 157–169. Elsevier, Amsterdam (1999)
- Ritter, H., Schulten, K.: Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimension selection. *Biological Cybernetics* 60, 59–71 (1988)
- Van Hulle, M.M.: Maximum likelihood topographic map formation. *Neural Computation* 17(3), 503–513 (2005)
- Wiskott, L., Sejnowski, T.J.: Constrained optimization for neural map formation: A unifying framework for weight growth and normalization. *Neural Computation* 10, 671–716 (1998)

RBF NN Based Adaptive PI Control of Brushless DC Motor

Jie Xiu¹, Yan Xiu², and Shiyu Wang³

¹ School of Electrical Engineering and Automation, Tianjin University,
300072 Tianjin, China
xiujie@tju.edu.cn

² Department of Fundamental Subject, Tianjin Institute of Urban Construction,
300384 Tianjin, China

³ School of Mechanical Engineering, Tianjin University,
300072 Tianjin, China

Abstract. The inherent nonlinear of brushless DC motor (BLDCM) makes it hard to get a good performance by using the conventional PI controller to the speed control of BLDCM. In this paper, a radial basis function (RBF) artificial neural network (NN) based adaptive PI controller for BLDCM is developed. The RBF NN has a strong ability of adaptive, self-learning and self-organization. At the same time, the nonlinear mapping property and high parallel operation ability of NN make it suitable to be applied to perform parameter identification. In this paper, the RBF NN is employed to predict the Jacobian information and tune the gains. Compared with back propagation (BP) type NN with sigmoid activation function, the RBF NN has a more fast convergence speed and can avoid getting stuck in a local optimum. Through parameter prediction, response speed of the system can be improved. The experimental results demonstrate that a high control performance is achieved. The system responds quickly with little overshoot. The steady state error is zero. The system shows robust performance to the load torque disturbance.

1 Introduction

The BLDCM drive is a kind of variable speed drive that has been widely applied. It has a relatively simple structure. It has permanent magnet on the rotor for exciting the magnet field. By this way, the usually exciting current and winding are eliminated. The exciting losses are saved. So, it has a high efficient. It is also suitable for high speed operation.¹

In essence, the BLDCM is a highly nonlinear plant. The conventional PI controller is widely applied to the control of electric machines. But it is hard to get a good performance when applied to control the speed of highly nonlinear BLDCM.

¹ This work is supported by Natural Science Foundation of National Natural Science Foundation of China(50705062) and National Basic Research Program of China (973 Program)(No. 2009CB219700).

In the conventional PI controller, once the parameters are tuned according to a certain condition, it will not be changed again. So, when the characteristic of the nonlinear plant changed, the parameters tuned formerly will not do. In order to get a high performance, the adaptive PI control strategy should be adopted.

NN is formed by the interconnection of artificial neurons. It has a strong ability of adaptive, self-learning, and self-organization. It also has a strong ability of nonlinear mapping and fault tolerance. Such, combining it with the conventional PI controller, an adaptive PI controller based on neural network can be constructed.

In the application of NN, the parameters of the plant should be identified. The performance of the controller can be improved though the prediction of parameters of the system. Compared with sigmoid activation function BP NN, the RBF NN has a more fast convergence speed and can avoid getting stuck in a local optimum. By using the RBF NN to identify the system parameters, the system can respond quickly.

Many researches have been done to implement the control of BLDCM drives based on NN. Wavelet NN is used to the control of BLDCM servo system [1]. Simultaneous online identification and control is designed for brushless dc motor drives. The dynamics of the motor/load are modeled online and controlled using artificial neural network [2]. NN is applied to the speed control of permanent magnet synchronous motors [3]. The vector control of brushless dc motor using a neural network is presented in [4]. In the proposed method, a neural network is employed as on-line estimator of the nonlinear dynamic equations of brushless dc motor. The application of NN to identification and control of BLDCM and other electric machine is reported [5]-[6]. In summary, the application of NN to the control of BLDCM can improve the performance of BLDCM. In order to get a high performance, an adaptive PI controller based on RBF NN for BLDCM is developed in this paper.

2 The BLDCM Drive System

The BLDCM drive consists of the brushless DC motor, the three-phase symmetric bridge power converter, DSP and rotor position sensor. This BLDCM is controlled by a TMS320LF2407 type DSP.

The BLDCM used in this paper is a three phase 4 poles motor with the permanent magnet mounted on the rotor surface. The power converter circuit used to supply this BLDCM is a three-phase full bridge power converter, which has six power MOSFETs and six diodes. There are two power MOSFETs and two diodes in each phase. Each of the inverter power MOSFETs conducts the motor current for 120° electrical per cycle.

3 The Adaptive PI Controller Based on RBF NN for BLDCM Drive

The structure of the RBF NN based adaptive PI controller for BLDCM is shown in Fig. 1.

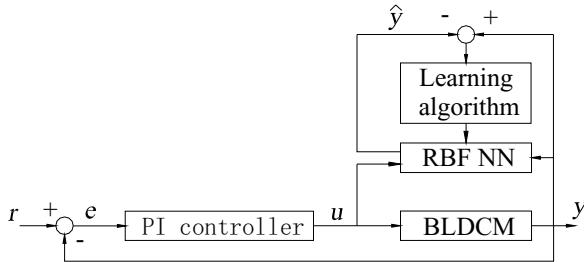


Fig. 1. The structure of the RBF NN based adaptive PI controller for BLDCM

In the system, the adaptive PI controller performs the closed loop control of BLDCM and its parameters are tuned on line. The NN is a RBF NN, which identify the Jacobian information on line and adaptively adjusts the parameters of the PI controller in order to get a set of optimal parameters. Thus an optimal performance of the system is reached.

3.1 The Adaptive PI Controller

The equation of adaptive increment PI controller can be expressed as

$$u(k) = u(k-1) + K_p(k)[e(k) - e(k-1)] + K_I(k)e(k) \quad . \quad (1)$$

where K_p and K_I is the proportional factor and integration factor respectively.

In conventional increment PI controller, the proportional factor and integration factor will not be changed again during the control process. But in the adaptive PI controller, the proportional factor and integration factor will be adjusted according to different error between command signal and output of the system.

The error between the command $r(k)$ and the system output $y(k)$ is defined as

$$e(k) = r(k) - y(k) \quad . \quad (2)$$

The adjustment of the parameters of the adaptive PI controller is according to the gradient descent algorithm as

$$\begin{aligned} \Delta K_p &= -\eta \frac{\partial E}{\partial k_p} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial \Delta u} \frac{\partial \Delta u}{\partial k_p} = \eta e(k) \frac{\partial y}{\partial \Delta u} (e(k) - e(k-1)) \\ \Delta K_I &= -\eta \frac{\partial E}{\partial k_I} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial \Delta u} \frac{\partial \Delta u}{\partial k_I} = \eta e(k) \frac{\partial y}{\partial \Delta u} e(k) \end{aligned} \quad . \quad (3)$$

where the $\frac{\partial y}{\partial \Delta u}$ is called Jacobian matrix.

Then the parameters of adaptive PI controller is updated as

$$\begin{aligned} K_p(k) &= K_p(k-1) + \Delta K_p \\ K_I(k) &= K_I(k-1) + \Delta K_I \end{aligned} \quad . \quad (4)$$

To speed up the adjustment of the parameters of the adaptive PI controller, the information of $\frac{\partial y}{\partial \Delta u}$ needs to be identified on line. This can be done by a RBF NN.

3.2 The Structure of RBF NN

The structure of the RBF NN is shown in Fig. 2. There are i neurons in the input layer, j neurons in the hidden layer and k neurons in the output layer. The input neurons are corresponding to variables that reflect the system operation condition. The output neurons are corresponding to the parameters of the system. The activation function of the neuron in the output layer is a liner activation function. The activation function of the neurons in the hidden layer is a Gaussian type activation function.

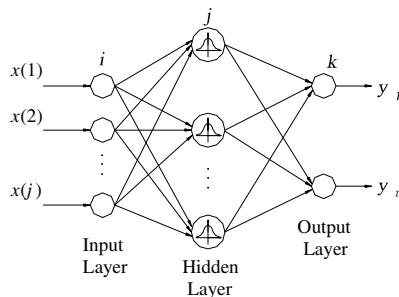


Fig. 2. The structure of the RBF neural network

3.3 The Identification of Jacobian Information Based on RBF NN

In order to adjust the parameters of adaptive PI controller, the value of $\frac{\partial y(k)}{\partial \Delta u(k)}$ in Eqs 3 should be identified and predicted on line. For BLDCM is a severely nonlinear plant, a nonlinear prediction model is used to perform the parameter prediction.

In this paper, the BLDCM can be expressed by a MISO mathematical equation as

$$y(k) = f[\Delta u(k-1), y(k-1), y(k-2)] . \quad (5)$$

In order to predict the value of $\frac{\partial y(k)}{\partial \Delta u(k)}$, in this paper, a three-layer RBF NN with three neurons in the input layer, five neurons in the hidden layer and one neuron in the output layer is employed to perform parameter prediction.

In this paper, the input vector of the neurons in the input layer of the RBF NN is $X=[\Delta u(k-1), y(k-1), y(k-2)]^T$. The vector used to construct the RBF NN is $H=[h_1, h_2, \dots, h_j, \dots, h_m]^T$, in the vector, where h_j is Gaussian type function

$$h_j(x) = \exp\left(-\frac{\|X - C_j\|^2}{2b_j^2}\right) \quad j=1,2,\dots,m \quad (6)$$

where C_j is the center of the receptive field of the j^{th} neuron, $j=1,2,\dots,m$; b_j is the width of the receptive field.

The output of RBF NN is

$$\hat{y}(k) = y_m(k) = w_1 h_1 + w_2 h_2 + \dots + w_m h_m \quad (7)$$

where w_j is the weight between the hidden layer and output layer.

The objective function of the RBF NN to be minimized is defined as

$$J_1 = \frac{1}{2} (y(k) - y_m(k))^2 \quad (8)$$

where y is the output of plant; y_m is the output of RBF NN.

According to the gradient descent method, the update equation of the weight between the hidden layer and the output layer, the center of the receptive field and the width of the receptive field can be expressed as

$$w_j(k) = w_j(k-1) + \eta'(k)(y(k) - y_m(k))h_j + \alpha'(w_j(k-1) - w_j(k-2)) \quad (9)$$

$$\Delta b_j = (y(k) - y_m(k))w_j h_j \frac{(X - C_j)^2}{b_j^3} \quad (10)$$

$$b_j(k) = b_j(k-1) + \eta'(k)\Delta b_j + \alpha'(b_j(k-1) - b_j(k-2)) \quad (11)$$

$$\Delta c_{ji} = (y(k) - y_m(k))w_j \frac{x_j - c_{ji}}{b_j^2} \quad (12)$$

$$c_{ji}(k) = c_{ji}(k-1) + \eta'(k)\Delta c_{ji} + \alpha'(c_{ji}(k-1) - c_{ji}(k-2)) \quad (13)$$

where $\eta'(K)$ is the learning rate; α' is momentum factor.

The partial derivative of $y(k)$ with respect to $u(k)$ called Jacobian matrix can be expressed as

$$\frac{\partial y(k)}{\partial u(k)} \approx \frac{\partial y_m(k)}{\partial u(k)} = \sum_{j=1}^m w_j h_j \frac{c_{ji} - x_i}{b_j^2} \quad (14)$$

where $x_i = u(k)$.

3.4 Adaptive Tuning of Learning Rate

To increase the convergence speed of NN, an adaptive learning algorithm is adopted in this paper that is to adjust the learning rate according to the error. When the current error is large than the former, the learning rate will not be changed, otherwise, the learning rate will be increased.

3.5 On-Line Training and Off-Line Training

To reduce computational complexity and improve initial response performance, off-line training is done. After off-line training, the control system is applied to on-line training and on-line control. Through on-line training, effect on the performance of the system caused by the change of system parameters can be eliminated. This helps to increase the robustness of the system.

4 Experimental Results

The experimental results are got under the control of the adaptive PI controller based on RBF ANN. The block diagram of the hardware control system for the BLDCM which based on the TMS320LF2407 type DSP is shown in Fig. 3.

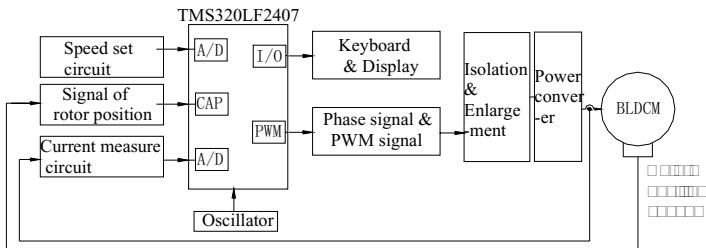


Fig. 3. The block diagram of hardware control system for BLDCM

In Fig. 4, the curve stands for the step response of the system under the control of the conventional PI controller. The parameters of the conventional PI controller are tuned as $K_P=10.6$, $K_I=0.15$.

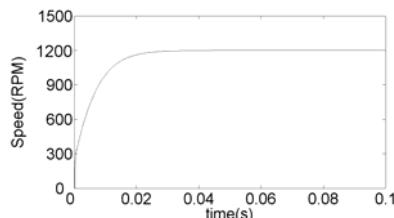


Fig. 4. Step response of BLDCM under the control of the conventional PI controller

With no load, the step response of the BLDCM under the control of the proposed controller is shown in Fig. 5. When under the control of the proposed controller, the system responds quickly with little overshoot. The steady state error is zero. From Fig. 5, it can be seen that after using the RBF NN to predict and identify the parameters of the system, the convergence speed of the adaptive PI controller is increased; the dynamic response of the system is much quick.

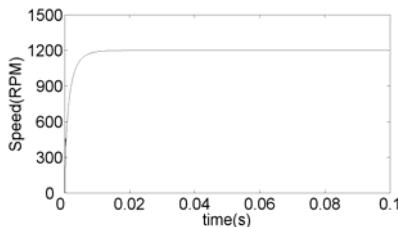


Fig. 5. Step response of BLDCM under the control of the proposed adaptive PI controller

The step response of the BLDCM under the control of the BP NN based PI controller is shown in Fig. 6.

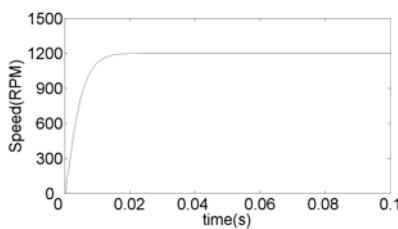


Fig. 6. Step response of BLDCM under the control of the BP NN based PI controller

In the dynamic response, the adjustment of the parameters of the proposed adaptive PI controller is shown in Fig. 7.

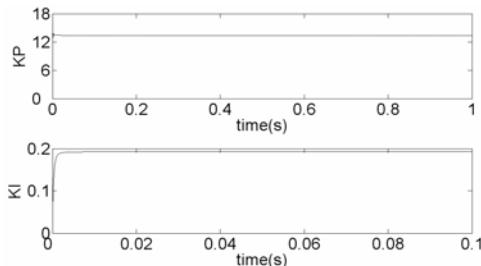


Fig. 7. Adjustment of parameters of the adaptive PI controller in dynamic response

When the system is disturbed, the response of the system is shown in Fig. 8. Curve 1 is the response of the system under the control of the proposed adaptive PI controller. Curve 2 is the response of the system under the control of the conventional PI controller. From Fig. 8, it can be seen that under the control of the proposed controller, the speed drop is little and the recover time is short. Under the control of the proposed controller, the system shows a robust performance to the torque disturbance.

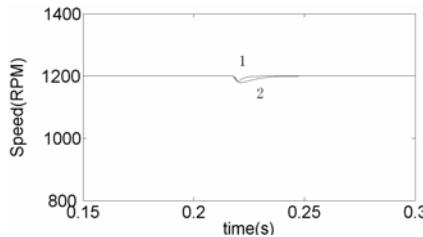


Fig. 8. Response of BLDCM under torque disturbance

5 Conclusions

In this paper, an adaptive PI controller of BLDCM based on RBF NN is developed. By using the strong ability of adaptive, self-learning, and self-organization of ANN and combining it with the PI control algorithm, an adaptive PI controller based on RBF NN to identify and predict system parameter is developed. Applying it to the speed control of BLDCM, the performance of BLDCM can be improved. By using the RBF NN to identify the nonlinear BLDCM and perform parameter prediction, the response of the BLDCM is improved. To increase the convergence speed of ANN, an adaptive learning algorithm is adopted in this paper. This makes the system response much quick. Experimental results prove that a high performance is achieved under the control of the proposed controller. The system responds quickly with little overshoot. Steady state error is zero. The system shows robust performance to the load torque disturbance.

References

1. Li, X.-X., Zhang, Q.-J., Xiao, H.-J.: The design of brushless DC motor servo system based on wavelet ANN. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol. 2, pp. 929–933 (2004)
2. Tipsuwanporn, V., Piyarat, W., Tarasantisuk, C.: Identification and control of brushless DC motors using on-line trained artificial neural networks. In: Proceedings of the Power Conversion Conference, vol. 3, pp. 1290–1294 (2002)
3. Saleh, K.I., Badr, M.A., Elwer, A.S., et al.: Analysis of controlled permanent magnet synchronous motor using artificial neural network. In: Proceedings of the Fifth International Conference on Electrical Machines and Systems, vol. 2, pp. 791–795 (2001)
4. Senju, T., Urasaki, N., Uezato, K.: Vector control of brushless DC motors using neural network. In: Proceedings of the International Conference on Power Electronics and Drive Systems, vol. 1, pp. 291–296 (1997)
5. Rubaai, A., Kotaru, R., Kankam, M.D.: A continually online-trained neural network controller for brushless DC motor drives. IEEE Transactions on Industry Applications 36(2), 475–483 (2000)
6. Senju, T., Shingaki, T., Omoda, A., et al.: High efficiency drives for synchronous reluctance motors using neural network. In: IECON 2000, vol. 2, pp. 777–782 (2000)

Incremental Principal Component Analysis Based on Adaptive Accumulation Ratio*

Seiichi Ozawa¹, Kazuya Matsumoto¹,
Shaoning Pang², and Nikola Kasabov²

¹ Graduate School of Engineering, Kobe University, Kobe 657-8501, Japan
ozawasei@kobe-u.ac.jp

² Knowledge Engineering & Discover Research Institute
Auckland University of Technology
Private Bag 92006, Auckland 1142, New Zealand
shaoning.pang@aut.ac.nz, nik.kasabov@aut.ac.nz

Abstract. We have proposed an online feature extraction method called Chunk Incremental Principal Component Analysis (Chunk IPCA) where a chunk of data is trained at a time to update an eigenspace model. In this paper, we propose an extended version of Chunk IPCA in which a proper threshold for the accumulation ratio is adaptively determined such that the highest classification accuracy is maintained for a validation data set. Whenever a new chunk of training data is given, the validation set is updated in an online fashion by using the k -means clustering or through the prototype selection based on the classification results. The experimental results show that the extended version of Chunk IPCA can determine a proper threshold on an ongoing basis, resulting in keeping higher classification accuracy than the original Chunk IPCA.

1 Introduction

In classification systems, feature extraction is very important for both good generalization performance and fast learning of classifiers. Thus, incremental learning should be considered not only for classifiers but also for feature extraction. However, many approaches to incremental learning have aimed for classifiers so far. As for the incremental learning of feature extraction, there have been proposed several incremental algorithms for principal component analysis [1,2,3,4] and linear discriminant analysis [5]. Hall and Martin [6] have proposed an incremental principal component analysis (IPCA) based on eigenvalue decomposition. Ozawa et al. [7] have extended this IPCA algorithm such that an eigen-axis is augmented based on the accumulation ratio in order to control the dimensionality of an eigenspace easily. Recently, we have proposed a new scheme of incremental learning in which feature extraction and classifier learning are simultaneously carried out, and developed an online face recognition system based

* The authors would like to thank Professor Shigeo Abe for his helpful comments and discussions. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C) 205002205.

on this scheme [8]. It is verified that the classification accuracy of this system is improved constantly even if a small set of training samples are provided at a starting point. Furthermore, to enhance the learning speed, we have proposed an extension of IPCA called Chunk IPCA [9] in which an eigenspace is updated for a chunk of data by performing single eigenvalue decomposition.

In Chunk IPCA, the dimensional augmentation is carried out if the accumulation ratio becomes lower than a threshold, which should be preset to a proper value in order to obtain good classification accuracy. In order to select a proper threshold automatically, the cross-validation method is introduced in the threshold selection using an initial training set [9]. However, once the threshold is determined, it is fixed over the entire learning stages. Obviously, the optimality for such a threshold is not ensured in incremental environments where the distribution of given training samples could be changed over time.

In this paper, we propose two methods to determine a proper threshold value adaptively in Chunk IPCA such that the classification accuracy for validation data is always maximized. In the first method, the prototypes of a validation data set are updated by applying k -means clustering to the following candidate prototype set: a chunk of training data and the existing prototypes. In the second method, the prototypes are updated such that some of them are selected from misclassified training data in the above candidate set.

This paper is organized as follows. Section 2 gives a quick review of Chunk IPCA in which a chunk of data is trained at one time. Then, Section 3 presents two methods to update an optimal threshold of accumulation ratio in Chunk IPCA. In Section 4, the evaluation for two UCI data sets are conducted and the effectiveness of updating the threshold is studied. Finally, Section 5 summarizes this work.

2 Chunk Incremental Principal Component Analysis

Hall et al.'s IPCA [6] and the extended IPCA [7] are applied to one sample at a time, and the intermediate eigenproblem must be solved repeatedly for every training sample. Hence, the learning may get stuck in a deadlock if a large chunk of training samples is given to learn in a short term. To overcome this problem, we have further extended the IPCA algorithms so that an eigenspace model could be updated with a chunk of training samples in a single operation. Let us call this extended algorithm *Chunk IPCA* [9].

2.1 Update of Mean Vector and Eigen-Axes

Let us assume that N training samples $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in \mathcal{R}^{n \times N}$ have been given so far and they were already discarded. Instead of keeping actual training samples, we hold an eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_d, \Lambda_d, N)$ where $\bar{\mathbf{x}}$, \mathbf{U}_d , and Λ_d are a mean input vector, an $n \times d$ eigenvector matrix, and a $d \times d$ eigenvalue matrix, respectively. Here, d is the number of eigen-axes spanning the eigenspace (i.e., eigenspace dimensionality). Now, assume that a chunk of L training samples $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_L\} \in \mathcal{R}^{n \times L}$ is presented.

Without the previous training samples \mathbf{X} , the updated mean vector $\bar{\mathbf{x}}'$ is easily obtained as follows:

$$\bar{\mathbf{x}}' = \frac{1}{N+L} \left(\sum_{i=1}^N \mathbf{x}_i + \sum_{j=1}^L \mathbf{y}_j \right) = \frac{1}{N+L} (N\bar{\mathbf{x}} + L\bar{\mathbf{y}}). \quad (1)$$

Suppose that l eigen-axes must be augmented to avoid the serious loss of essential input information when a chunk of L training samples \mathbf{Y} is provided; that is, the eigenspace dimensions are increased by l . Let us denote the augmented eigen-axes as follows:

$$\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_l] \in \mathcal{R}^{n \times l}. \quad (2)$$

Then the updated eigenvector matrix \mathbf{U}'_{d+l} is represented by using a rotation matrix \mathbf{R} and the current eigenvector matrix \mathbf{U}_d .

$$\mathbf{U}'_{d+l} = [\mathbf{U}_d, \mathbf{H}] \mathbf{R}. \quad (3)$$

The rotation matrix \mathbf{R} and the eigenvalue matrix Λ'_{d+l} are obtained by solving the following intermediate eigenproblem [9]:

$$\begin{aligned} & \left(\frac{N}{N+L} \begin{bmatrix} \Lambda_d & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0} \end{bmatrix} + \frac{NL^2}{(N+L)^3} \begin{bmatrix} \bar{\mathbf{g}}\bar{\mathbf{g}}^T & \bar{\mathbf{g}}\bar{\gamma}\bar{\gamma}^T \\ \bar{\gamma}\bar{\mathbf{g}}^T & \bar{\gamma}\bar{\gamma}^T \end{bmatrix} + \frac{N^2}{(N+L)^3} \sum_{i=1}^L \begin{bmatrix} \mathbf{g}'_i \mathbf{g}'_i^T & \mathbf{g}'_i \gamma'_i \gamma'^T \\ \gamma'_i \mathbf{g}'_i^T & \gamma'_i \gamma'_i \gamma'^T \end{bmatrix} \right. \\ & \left. + \frac{L(L+2N)}{(N+L)^3} \sum_{i=1}^L \begin{bmatrix} \mathbf{g}''_i \mathbf{g}''_i^T & \mathbf{g}''_i \gamma''_i \gamma''^T \\ \gamma''_i \mathbf{g}''_i^T & \gamma''_i \gamma''_i \gamma''^T \end{bmatrix} \right) \mathbf{R} = \mathbf{R} \Lambda'_{d+l} \end{aligned} \quad (4)$$

where

$$\bar{\mathbf{g}} = \mathbf{U}_d^T (\bar{\mathbf{y}} - \bar{\mathbf{x}}), \quad \mathbf{g}'_i = \mathbf{U}_d^T (\mathbf{y}_i - \bar{\mathbf{x}}), \quad \mathbf{g}''_i = \mathbf{U}_d^T (\mathbf{y}_i - \bar{\mathbf{y}}),$$

$$\bar{\gamma} = \mathbf{H}^T (\bar{\mathbf{y}} - \bar{\mathbf{x}}), \quad \gamma'_i = \mathbf{H}^T (\mathbf{y}_i - \bar{\mathbf{x}}), \quad \gamma''_i = \mathbf{H}^T (\mathbf{y}_i - \bar{\mathbf{y}}).$$

2.2 Augmentation of Eigen-Axes

In Chunk IPCA, the number of eigen-axes to be augmented is determined by finding a minimum d such that the accumulation ratio $A(\mathbf{U}_d)$ satisfies the following condition: $A(\mathbf{U}_d) \geq \theta$ where θ is a constant between 0 and 1. The update equation of $A(\mathbf{U}_d)$ must be derived so that it can be updated with a chunk of training samples in one-pass [9], and it is given by

$$A'(\mathbf{U}_d) \stackrel{\text{def}}{=} \frac{\sum_{i=1}^d \lambda'_i}{\sum_{i=1}^n \lambda'_i} = \frac{\sum_{i=1}^d \lambda_i + \frac{L}{N+L} \|\bar{\mathbf{g}}\|^2 + \frac{1}{N} \sum_{j=1}^L \|\mathbf{g}''_i\|^2}{\sum_{i=1}^n \lambda_i + \frac{L}{N+L} \|\bar{\mathbf{x}} - \bar{\mathbf{y}}\|^2 + \frac{1}{N} \sum_{j=1}^L \|\mathbf{y}_j - \bar{\mathbf{y}}\|^2}. \quad (6)$$

As seen from Eq. (6), no past sample is needed to update $A(\mathbf{U}_d)$.

In IPCA, a new eigen-axis is selected so as to be perpendicular to the existing eigenvectors which are given by the column vectors of \mathbf{U}_d . A straightforward way to get new eigen-axes is to apply Gram-Schmidt orthogonalization technique to the given chunk of training samples. If the chunk training samples are represented

Algorithm 1. Eigen-axis Selection in Chunk IPCA [9]**Input:**

- Eigenspace model: $\Omega = (\bar{x}, \mathbf{U}_d, \mathbf{A}_d, N)$.
- Threshold of accumulation ratio: θ .
- Training samples: \mathbf{y}_l ($l = 1, \dots, L$).

$$\mathbf{H} = \{ \}$$

Calculate the accumulation ratio $A'(\mathbf{U}_d)$ in Eq. (6).

if $A'(\mathbf{U}_d) \geq \theta$ **then**

 Terminate this algorithm.

end if

Obtain a set \mathcal{L} of the following residue vectors \mathbf{h}_l :

$$\mathbf{h}_l = \frac{\mathbf{r}_l}{\|\mathbf{r}_l\|} \quad (5)$$

where

$$\mathbf{r}_l = (\mathbf{y}_l - \bar{x}) - [\mathbf{U}_d, \mathbf{H}] [\mathbf{U}_d, \mathbf{H}]^T (\mathbf{y}_l - \bar{x}).$$

repeat

 Find the following residue vector \mathbf{h}_{l^*} which gives the maximum accumulation ratio $A'([\mathbf{U}_d, \mathbf{H}, \mathbf{h}_l])$:

$$l^* = \arg \max_l A'([\mathbf{U}_d, \mathbf{H}, \mathbf{h}_l]).$$

 Update $\mathbf{H}' = [\mathbf{H}, \mathbf{h}_{l^*}]$ and remove \mathbf{h}_{l^*} from \mathcal{L} .

until $A'([\mathbf{U}_d, \mathbf{H}']) < \theta$

Output: Augmented eigen-axes: \mathbf{H}' .

by \tilde{L} linearly independent vectors, the maximum number of eigen-axes to be augmented is also \tilde{L} . However, the feature space spanned by all of the augmented eigen-axes is redundant in general. Therefore, we need to select informative eigen-axes from the \tilde{L} eigen-axes efficiently.

To construct a compact feature space, we should find a smallest set of augmented eigen-axes such that the eigenspace includes as much the energy of the given chunk data as possible. A straightforward way to find the set is to select an eigen-axis one by one, each of which gives a maximum increment of accumulation ratio (see [9] for details). The algorithm of the eigen-axis selection is summarized in Algorithm 1.

3 Method to Generate Validation Data

In Chunk IPCA, the number of eigen-axes to be augmented is determined by finding a minimum d such that $A'(\mathbf{U}_d) \geq \theta$ holds. To determine an optimal threshold θ at every learning stage, a cross-validation method can be introduced; that is, θ can be changed over learning stages such that the classification accuracy for a validation data set is always maximized.

Algorithm 2. Generate validation data set in CIPCA-kmeans**Input:**

- The number of classes: C .
- Initial training data set:
 $\mathbf{X}(0) = \{\mathbf{X}^1(0), \dots, \mathbf{X}^C(0)\}$.
- k -means clustering method.

// Initial Validation Set

for $c = 1, \dots, C$ **do** Apply an initial training data set $\mathbf{X}^c(0)$ of class c to k -means clustering method. Obtain the k prototype vectors:

$$\mathbf{P}^c(0) = \{\mathbf{p}_1^c(0), \dots, \mathbf{p}_k^c(0)\}.$$

end forDefine a whole set of $\mathbf{P}^c(0)$ ($c = 1, \dots, C$) as initial validation set $\mathbf{V}(0)$.

// Update Validation Set

 $t = 1$ **repeat****for** $c = 1, \dots, C$ **do** Apply a new training data set $\mathbf{X}^c(t)$ and the prototype set $\mathbf{P}^c(t-1)$ to k -means clustering method. Obtain the k prototype vectors:

$$\mathbf{P}^c(t) = \{\mathbf{p}_1^c(t), \dots, \mathbf{p}_k^c(t)\}.$$

end forDefine a whole set of $\mathbf{P}^c(t)$ ($c = 1, \dots, C$) as validation set $\mathbf{V}(t)$. $t \leftarrow t + 1$.**until** No training data is given.

In this paper, we propose two methods to generate a validation data set on an ongoing basis. Let $\mathbf{P}^c(t) = \{\mathbf{p}_1^c(t), \dots, \mathbf{p}_k^c(t)\}$ be a validation data set of class c ($c = 1, \dots, C$) at the t th learning stage where C and k are respectively the numbers of classes and prototypes. In the first method, the validation data set $\mathbf{P}^c(t)$ of class c at the t th learning stage is obtained by selecting k prototypes from the previous validation data $\mathbf{P}^c(t-1)$ and a chunk of given training samples $\mathbf{X}^c(t) = \{\mathbf{x}_1^c(t), \dots, \mathbf{x}_{L^c}^c(t)\}$ where L^c is the number of class c training samples given at the t th learning stage. This can be done by applying the conventional k -means clustering to those data. Therefore, in the first method, the validation data set $\mathbf{P}^c(t)$ is updated such that the distribution of training samples is approximated by k prototypes as accurately as possible.

In the second method, the validation data $\mathbf{P}^c(t)$ are selected such that some of the previous validation data $\mathbf{P}^c(t-1)$ are replaced with hard-to-classify training samples. Since such hard-to-classify samples are generally located around class boundaries, it is expected that the threshold θ is determined such that extracted features are effective in classifying classes correctly.

For the notational convenience, the Chunk IPCA algorithms based on the first and second methods are called *CIPCA-kmeans* and *CIPCA-classify*, respectively. The algorithms to generate a validation data set are summarized in Algorithm 2 and 3.

Algorithm 3. Generate validation data set in CIPCA-classify**Input:**

- The number of classes C .
- Initial training data set:
 $\mathbf{X}(0) = \{\mathbf{X}^1(0), \dots, \mathbf{X}^C(0)\}$.
- Nearest neighbor (NN) classifier.

// Initial Validation Set

for $c = 1, \dots, C$ **do**

Classify an initial training data set $\mathbf{X}^c(0)$ of class c with NN classifier based on the leave-one-out method.

Separate $\mathbf{X}^c(0)$ into the two data sets which consist of the classified and misclassified data.

Select the k prototype vectors:

$\mathbf{P}^c(0) = \{\mathbf{p}_1^c(0), \dots, \mathbf{p}_k^c(0)\}$ from the two sets randomly.

end for

Define a whole set of $\mathbf{P}^c(0)$ ($c = 1, \dots, C$) as initial validation set $\mathbf{V}(0)$.

// Update Validation Set

 $t = 1$ **repeat****for** $c = 1, \dots, C$ **do**

Classify a new training data set $\mathbf{X}^c(t)$ and the prototype set $\mathbf{P}^c(t-1)$ with NN classifier based on the leave-one-out method.

Separate $\mathbf{X}^c(t)$ and $\mathbf{P}^c(t-1)$ into the two data sets which consist of the classified and misclassified data.

Select the k prototype vectors $\mathbf{P}^c(t) = \{\mathbf{p}_1^c(t), \dots, \mathbf{p}_k^c(t)\}$ from the two sets randomly.

end for

Define a whole set of $\mathbf{P}^c(t)$ ($c = 1, \dots, C$) as validation set $\mathbf{V}(t)$.

 $t \leftarrow t + 1$.**until** No training data is given.

4 Experiments

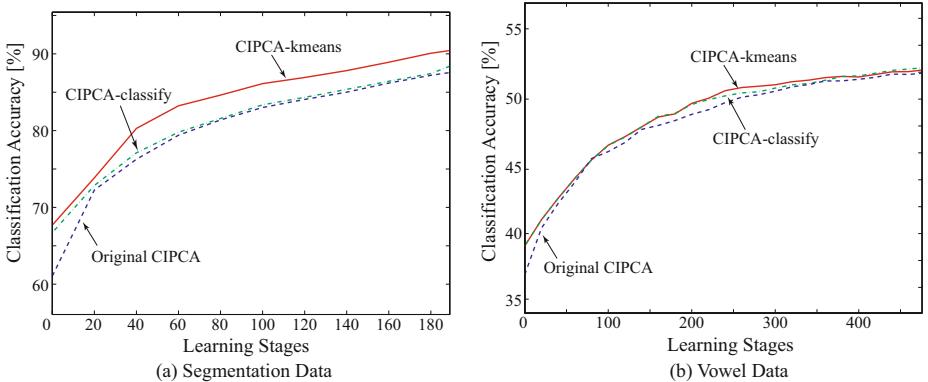
Two data sets in the UCI Machine Learning Repository, Segmentation and Vowel-context, are used for the evaluation. The data information is shown in Table I.

Assume that a part of training samples are provided as initial training data. Then, an initial eigenspace is calculated for the initial data by using the conventional PCA. In the experiments, 10% of training data in the original UCI data set are randomly selected as initial data. The remaining 90% of data are sequentially provided in chunks as training samples. Although the proposed Chunk IPCA works for any size of data chunk, for the sake of simplicity, the chunk size is fixed at 20 over the entire learning period.

A chunk of training samples is randomly selected and it has no overlap with other chunks; hence all the training samples are presented only once because

Table 1. Evaluated datasets

Data sets	#Attrib.	#Classes	#Train.	#Test
Segmentation	19	7	2,100	210
Vowel-context	10	11	528	462

**Fig. 1.** Time evolution of classification accuracy for three Chunk IPCA algorithms: Original CIPCA, CIPCA-kmeans, and CIPCA-classify

we assume one-pass incremental learning environments. Since the performance of incremental learning generally depends on the sequence of training samples, fifty trials with different sequences of training samples are carried out to evaluate the average performance for the test datasets in Table 1.

The evolutions of classification accuracy are shown in Figs. 1 (a) and (b). In the original Chunk IPCA, the threshold value is determined at the initial leaning stage and it is fixed over a learning session. As seen from Fig. 1 (a), the classification accuracy of the proposed CIPCA-kmeans for Segmentation data is largely improved as compared with the other two methods. For Vowel data, although there is no significant difference in the final accuracy of the three methods, the accuracies of CIPCA-kmeans and CIPCA-classify are quickly going up as compared with that of the original Chunk IPCA. Therefore, we conclude that the online performance of CIPCA-kmeans is better than those of CIPCA-classify and the original Chunk IPCA for the two UCI datasets.

5 Conclusions

This paper presents a new Chunk IPCA algorithm where the threshold of accumulation ratio is adaptively determined such that the classification accuracy is maximized for a validation data set. We proposed two methods to obtain the validation set in an online fashion. In the first method, the prototypes of a validation data set are updated by applying k -means clustering to a chunk of training

data and the existing prototypes. In the second method, several prototypes are replaced with the misclassified training samples to keep hard-to-classify data at every learning stage. The two Chunk IPCA algorithms whose validation data sets are obtained by the first and second methods are called CIPCA-kmeans and CIPCA-classify, respectively.

In the experiments, the evaluation of the proposed Chunk IPCA algorithms was made for the two UCI data sets and the effectiveness of updating the threshold was discussed. The experimental results showed that the extended Chunk IPCA could determine a proper threshold adaptively so as to keep high classification accuracy. For Segmentation data, CIPCA-kmeans outperforms both CIPCA-classify and the original Chunk IPCA in terms of classification accuracy.

References

1. Oja, E., Karhunen, J.: On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix. *J. Math. Analysis and Application* 106, 69–84 (1985)
2. Sanger, T.D.: Optimal Unsupervised Learning in a Single-layer Linear Feedforward Neural Network. *Neural Networks* 2(6), 459–473 (1989)
3. Chattejee, C., Kang, Z., Roychowdhury, V.: Algorithms for Accelerated Convergence of Adaptive PCA. *IEEE Trans. on Neural Networks* 11(2), 338–355 (2000)
4. Weng, J., Zhang, Y., Hwang, W.-S.: Candid Covariance-free Incremental Principal Component Analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25(8), 1034–1040 (2003)
5. Pang, S., Ozawa, S., Kasabov, N.: Incremental Linear Discriminant Analysis for Classification of Data Streams. *IEEE Trans. on Systems, Man, and Cybernetics, Part B* 35(5), 905–914 (2005)
6. Hall, P., Martin, R.: Incremental Eigenanalysis for Classification. In: Proc. of British Machine Vision Conference, vol. 1, pp. 286–295 (1998)
7. Ozawa, S., Pang, S., Kasabov, N.: A Modified Incremental Principal Component Analysis for On-line Learning of Feature Space and Classifier. In: Zhang, C., Guesgen, H.W., Yeap, W.-K. (eds.) PRICAI 2004. LNCS (LNAI), vol. 3157, pp. 231–240. Springer, Heidelberg (2004)
8. Ozawa, S., Toh, S.L., Abe, S., Pang, S., Kasabov, N.: Incremental Learning of Feature Space and Classifier for Face Recognition. *Neural Networks* 18(5-6), 575–584 (2005)
9. Ozawa, S., Pang, S., Kasabov, N.: Incremental Learning of Chunk Data for On-line Pattern Classification Systems. *IEEE Trans. on Neural Networks* 19(6), 1061–1074 (2008)

Ontology Based Personalized Modeling for Chronic Disease Risk Analysis: An Integrated Approach

Anju Verma, Nikola Kasabov, Elaine Rush, and Qun Song

Knowledge Engineering and Discovery Research Institute,
Auckland University of Technology, Auckland
anju.verma@aut.ac.nz

Abstract. A novel ontology based chronic disease risk analysis system framework is described, which allows the creation of global knowledge representation (ontology) and personalized modeling for a decision support system. A computerized model focusing on organizing knowledge related to three chronic diseases and genes has been developed in an ontological representation that is able to identify interrelationships for the ontology-based personalized risk evaluation for chronic diseases. The personalized modeling is a process of model creation for a single person, based on their personal data and the information available in the ontology. A transductive neuro-fuzzy inference system with weighted data normalization is used to evaluate personalized risk for chronic disease. This approach aims to provide support for further discovery through the integration of the ontological representation to build an expert system in order to pinpoint genes of interest and relevant diet components.

1 Introduction

Populations are aging and the prevalence of chronic diseases which persists for many years is increasing. Chronic diseases are illnesses that are caused by many factors (genes, environment, life style e.g. tobacco use, lack of physical activity, and poor eating habits). Chronic diseases generally cannot be prevented by vaccines or cured by medication, nor do they just disappear but may be cured or modified by changing life style. The most common chronic diseases in developed countries are cardiovascular disease, type 2 diabetes, obesity, arthritis and cancer (breast and colon). The chronic diseases such as cardiovascular disease, type-2 diabetes and obesity have high global prevalence, have multifactorial aetiology and are mainly caused by interactions of a number of common factors including genes, nutrition and life-style. For ontology based personalized risk evaluation for chronic diseases, a Protégé-based ontology has been developed for entering data for three chronic diseases (type 2 diabetes; CVD and obesity) and linking and building relationships among concepts. The ontological representation provides the framework into which information on individual patients for disease symptoms, gene maps, diet and life history details can be inputted, and risks, profiles, and recommendations derived.

A personalized risk evaluation system has been used for building the personalized model. Global models capture trends in data that are valid for the whole problem space, and local models, capture local patterns which are valid for clusters of data.

Both models contain useful information and knowledge. Local models are also adaptive to new data as new clusters and new functions that capture patterns of data in these clusters. A local model can be incrementally created. Usually, both global and local modeling approaches assume a fixed set of variables and if new variables, along with new data, are introduced with time, the models are very difficult to modify in order to accommodate these new variables. However new variables be accommodated in personalized models, as they are created “on the fly” provided that there is data for them [1]. The next two sections will describe a chronic disease ontology and personalized risk evaluation method (TWNFI) which is used to build personalized models , its comparison with global and local models and the framework for integration of chronic disease ontology and personalized modeling.

2 Chronic Disease Ontology

Ontology is a systematic account of being or existence. Ontology in terms of bioinformatics can be interpreted as the representation of the existing domain of the knowledge of life. Ontology is used to reason and make inferences about the objects within the domain [2]. Ontology is concerned with making information and knowledge explicit; it includes descriptions of concepts and their relationships. Ontology describes a hierarchical structure of concepts and the relationships built in order to extract new knowledge. Ontology is generally written as a set of definitions of the formal vocabulary of objects and relationships in the given domain. It supports the sharing and reuse of formally represented knowledge among systems [3, 4]. As a database technology, ontologies are commonly coded as triple stores (subject, relationship, object), where a network of objects is formed by relationship linkages, as a way of storing semantic information [5, 6]. A standardized ontology framework makes data easily available for advanced methods of analysis, including artificial intelligence algorithms, that can tackle the multitude of large and complex datasets by clustering, classification, and rule inference for biomedical and bioinformatics applications. The main advantages of building ontology are to extract and collect knowledge; share knowledge; manage terminology; store, retrieve and analyze; find relationships between the concepts; discover new knowledge and reuse knowledge for decision support systems.

Chronic diseases develop from conception over the life course in the presence of a number of interrelated factors including genetic predisposition, nutrition and life-style or interaction with the environment. With the development and completion of human genome sequencing, we are able to trace the genes responsible for proteins and metabolites that are linked with these diseases. The chronic disease ontology database developed contains information about three chronic diseases; cardiovascular disease, type-2 diabetes and obesity; their related genes and mutations, as well as health, nutrition and life history data. Personal genome and health data could provide a guide to design and build a medical health administration system to apply to annual relevant medical tests, e.g. gene expression level changes for health surveillance. Chronic disease ontology consists of five major domains namely; organism domain, molecular domain, medical domain, nutritional domain and a biomedical informatics map

domain. These domains or classifications contain further subclasses and instances. Each subclass has a set of slots which provide information about each instance and have relationships among other slots, instances and concepts. The chronic disease ontology also contains information about the genes involved in cardiovascular disease, type-2 diabetes and obesity. There are about 76 genes in the ontology. Each gene instance has different information associated with the gene and also has relationships with other domains. The advantage of this chronic disease ontology is that it can be updated manually and regularly with new knowledge and information providing a framework to keep an individual's specific information (medical, genetic, clinical and nutritional), to discover new knowledge and to adapt as required for personalized risk prediction and advice.

3 Personalized Risk Evaluation System for Chronic Diseases

With the emergence of personalized medicine and nutrition, we are able to build tools to predict personalized risk and advice systems. There have been several widely used methods for cardiovascular risk evaluation such as the Anderson formula [7], the New Zealand cardiovascular risk charts [8, 9], and the PREDICT-CVD method [10]. These methods assess risk by using only clinical variables but as each person is different and responds to each nutrient and medicine differently, it is important to build a tool that can predict personalized or individualized risk for cardiovascular disease for a discrete person. So in order to achieve this different methods of personalized risk evaluation (Weighted distance, Weighted variables K-nearest neighbors (WWKNN)) [1] have been with Transductive neuro-fuzzy inference systems with weighted data normalization (TWNFI) [11]. TWNFI uses a Zadeh-Mamdani type fuzzy inference engine as its frame and it applies the transductive technique and weighted data normalization method for its learning. TWNFI performs an improved local generalization to include new data because it develops an individual model for each data vector that takes into account the new input vector location in the space. This is an adaptive model, in the sense that input-output pairs of data can be added to the data set continuously and be immediately available for transductive inference of local models. TWNFI creates a unique sub-model for each data sample and usually needs more time to compute than an inductive model, especially in training and simulation using large data sets. Creating a personalized model for an individual to predict risk for cardiovascular disease, could increase the accuracy of the prediction, especially when more variables such as clinical, genetic, nutritional factors are included in the prognosis. This personalized model has the advantage that not only variables used in the past to create a generalized model are used. The Anderson's formula [7] for example was derived from clinical variables collected from a North American population in the 1980s to predict a cardio-vascular event. To our knowledge nutritional variables have not been used to date to create a model to predict personalized risk for cardiovascular disease.

We have used TWNFI to evaluate personalized renal function using only six variables only and higher accuracy was achieved in comparison to other methods [11]. In this report TWNFI is used to predict risk for cardiovascular disease (specifically hypertension) using 13 variables. A person is considered to have high risk for cardiovascular if his/her has systolic blood pressure ≥ 130 or diastolic blood pressure ≥ 85 mmHg or treatment for hypertension (takes blood pressure medication) [12].

Individual health related data from a New Zealand national nutrition survey (NNS97) individuals has been used to build a personalized model. This data includes information mainly on nutrients intake and nutrition related clinical and anthropometric measures and general demographic information. The variables of interest include age, self-identified ethnicity, heart rate, waist circumference, ratio of sub scapular and triceps skin folds, ratio of total blood cholesterol and HDL and haemoglobin. Dietary intake from a 24 hour recall includes protein, carbohydrates, sugar, total fat, total saturated fat and salt. The diagnostic test for cardiovascular disease was hypertension as defined in the previous paragraph.

Table 1. Example of TWNFI personalized model for two different samples showing different weights for same variables and global weights male and female data

Input Variables	Sample 1 (male)			Sample 2 (female)		
	Values of input	Weights of input variables (TWNFI)	Global weights (male)	Values of input	Weights of input variables (TWNFI)	Global weights (female)
Age, y	62	0.5246	0.96995	71	0.8882	1
Ethnicity European/ Not European?	1	0.6384	0.02035	1	0.737	0.01315
Heart rate, bpm	54	0.772	0.18915	67	0.7617	0.29535
Waist circumference(cm)	113.1	0.6754	0.8325	72.2	0.8732	0.38165
Sub scapular/triceps skin fold	1.76	0.8199	0.869	0.69	0.779	0.35945
Total cholesterol/HDL*	6.9	0.574	0.94245	5.44	1	0.76515
Haemoglobin, g/litre	155	0.999	0.5412	127	0.8751	0.2465
Protein intake g/day	19.27	0.9446	0.08865	11.09	0.7792	0.5216
Carbohydrate intake, g/day	47.04	0.7942	0.33	43.65	0.796	0.6064
Sugar intake g/day	15.96	0.9813	0.00665	4.94	0.784	0.0277
Total fat intake g/day	34.07	0.8145	0.2117	45.66	0.727	0.085
Total saturated fat intake g/day	14.62	0.741	0.2863	21.03	0.8456	0.2652
Salt intake, mg/day	4545	1	0.1056	2335	0.7557	0.25995
Desired Output	1			1		
Predicted output with MLR*		0.8997			0.6956	
Predicted output by WWKNN*		0.4722			0.5672	
Predicted Output by TWNFI*		1.0031			0.89834	

*HDL (High density lipoproteins), MLR (Multiple linear regression), WWKNN (Weighted distance, weighted variables K-nearest neighbors), TWNFI (Transductive neuro-fuzzy inference systems with weighted data normalization).

This approach to predicting risk for cardio-vascular disease is original in that it uses both the individual's clinical measures and their reported intake of important nutrients which can affect a person's risk for disease. A separate model was been created for male and female samples.

To test this personalized model, two different groups of people represented by sample one and two were considered, a personalized model built and the individual weights for all variables compared with global weights. For future surveys of groups of people a personalized model can be created and then used to predict an individual's risk of cardio vascular disease. In the present example, sample one is male and sample two is female; both are at very high risk of cardiovascular disease. Table 1 shows the experimental weightings for these the male and female group samples.

The outcome with TWNFI method for male sample is 1.0031 and outcome for female sample is 0.89834; which means that both male and female samples are at high risk of having the disease. In comparison with multiple linear regression and WWKNN, TWNFI gives higher accuracy plus more knowledge can be derived from TWNFI such as ranking the importance of variables and deriving a set of rules. Therefore, we are able to understand more about each variable and can discover new relationships between variables. In the present example, it was found that for male sample salt intake was a highly important factor for hypertension but in the case of the female sample the ratio of total cholesterol to HDL cholesterol was more important. Interestingly, age was found to be the most important of all variables. But if an individual's weights are compared with their respective global weights there are distinct differences in pattern for each individual. This supports the need for personalized prediction and advice.

For TWNFI fuzzy rules are generated depending on the nearest neighbors. It has been found that the best number of clusters or fuzzy rules is between 6 and 8, and the best number of nearest neighbors for training is around 18. The fuzzy rules are generated on the basis of a Gaussian membership function. The optimization of membership functions informs the search for the overall optimal solution for the parameters spaces. For example for male sample the best set of rules generated:

If Age is Gaussian MF (27.64 60.85)
 Heart rate is Gaussian MF (47.27 52.21)
 Waist circumference is Gaussian MF (78.44 112.46)
 Sub scapular/triceps skin fold is Gaussian MF 0.87 1.67)
 Total cholesterol/HDL is Gaussian MF (3.75 8.31)
 Haemoglobin is Gaussian MF (116.24 154.35)
 Protein intake is Gaussian MF (3.93 22.36)
 Carbohydrates intake is Gaussian MF 24.19 42.76)
 Sugar intake is Gaussian MF (13.27 17.90)
 Total fat intake is Gaussian MF (13.66 36.08)
 Total saturated fat intake is Gaussian MF 38.41 5.21)
 Salt intake is Gaussian MF (3308.36 2162.67)
 Then risk of getting Cardiovascular disease is Gaussian MF (0.17 1.00)

4 Integration of Chronic Disease Ontology and Personalized Modeling

The framework and the software platform presented here bring together ontology knowledge repository and machine learning techniques to facilitate sophisticated adaptive data and information storage, retrieval, modeling, and knowledge discovery. The framework consists of an ontology and data repository module; and a machine learning module. A generic diagram of this framework is shown in figure 1. Both modules evolve through continuous learning from new data. Results from the machine learning procedures can be entered back to the ontology thus enriching its knowledge base and facilitating new discoveries. The framework utilizes ontology based data, as well as new knowledge inferred from the data embedded in the ontology. The software platform allows for the adaptation of an existing knowledge base to new data sources and at the same time entering results from machine learning and reasoning models. There is an interface module between the two modules that is application specific.

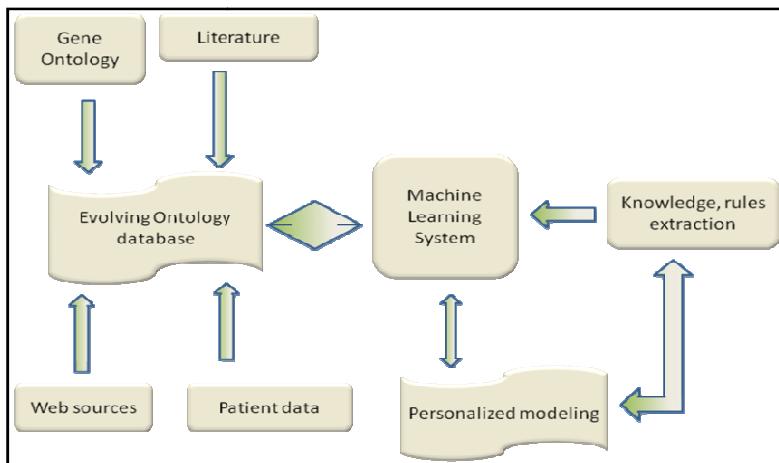


Fig. 1. The ontology-based personalized risk evaluation and advice system framework

The general framework as a software platform is characterized by the following characteristics:

- Chronic disease ontology developed in Protégé ontology development environment;
- Data import module to enter external multimodal data into ontology;
- Data retrieval module to search and retrieve relevant data from an ontology;
- Machine inference module that TWNFI as personalized techniques;
- User-friendly interface modules that can be tailored to predict risk for disease;
- Plus a module for updating the ontology, based on classification and clustering results from the machine inference module.

The system framework described here will be able to combine data from numerous sources to provide individualized person/case reports and recommendations on actions/interventions to help modify the outcome in a desired direction based on previously accumulated information on input variables and outcomes in the database. The framework presented needs to be developed further in terms of:

- 1) an efficient integration of old and new concepts and information in existing ontology
- 2) an automated search for relevant data to a new person's data
- 3) novel, more efficient methods for personalized feature selection and model optimization.
- 4) multiple model creation for a single person and cross model analysis for the discovery of new interactions between variables from different models, e.g. genes related to different types of chronic disease.

Acknowledgements. This study has been supported by Foundation of Research and Technology by TIF scholarship through an affiliate of Pacific Channel Limited and KEDRI, AUT. We would like to thank Ministry of Health for providing Health related data (NNS97).

References

1. Kasabov, N.: Global, local and personalized modeling and profile discovery in Bioinformatics: An integrated approach. *Pattern Recognition Letters* 28(6), 673–685 (2007)
2. Gruber, T.R.: A translation approach to portable ontologies. *Knowledge Acquisition* 5, 199–220 (1993)
3. Fensel, D.: *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, 2nd edn. Springer, Heidelberg (2004)
4. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? *Intelligent Systems and Their Applications* 14, 20–26 (1999)
5. Owens, A.: Semantic Storage: Overview and Assessment. Technical Report IRP Report 2005, Electronics and Computer Science, U of Southampton (2005)
6. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* (May 17) (2001)
7. Anderson, K.M., Odell, P.M., Wilson, P.W.F., Kannel, W.B.: Cardiovascular disease risk profiles. *American Heart J.* 121(1), 293–298 (1991)
8. Jackson, R.: Updated New Zealand cardiovascular disease risk-benefit prediction guide. *British Medical Journal* 320, 709–710 (2000)
9. Milne, R., Gamble, G., Whitlock, G., Jackson, R.: Framingham Heart study risk equation predicts first cardiovascular event rates in New Zealanders at the population level. *The New Zealand Medical Journal* 116(1185) (2003)
10. Bannink, L., Wells, S., Broad, J., Riddell, T., Jackson, R.: Web-based assessment of cardiovascular disease risk in routine primary care practice in New Zealand: the first 18,000 patients (PREDICT CVD-1). *The New Zealand Medical Journal*, 119(1245) (2006)
11. Song, Q., Kasabov, N.: TWNF1 - a transductive neuro-fuzzy inference system with weighted data normalization for personalized modeling. *Neural Networks* 19(10), 1591–1596 (2006)
12. Alberti, K.G.M.M., Zimmet, P., Shaw, J.: Metabolic syndrome-a new world-wide definition. A Consensus Statement from the International Diabetes Federation. *Diabetic Medicine* 23, 469–480 (2006)

Frost Prediction Characteristics and Classification Using Computational Neural Networks

Philip Sallis¹, Mary Jarur², and Marcelo Trujillo²

¹ Geoinformatics Research Centre, School of Computing and Mathematical Sciences,
Auckland University of Technology, New Zealand

² Escuela de Computación e Informática, Facultad de Ingeniería
Universidad Católica del Maule, Chile

Abstract. The effect of frost on the successful growth and quality of crops is well understood by growers as leading potentially to total harvest failure. Studying the frost phenomenon, especially in order to predict its occurrence has been the focus of numerous research projects and investigations. Frost prone areas are of particular concern. Grape growing for wine production is a specific area of viticulture and agricultural research. This paper describes the problem, outlines a wider project that is gathering climate and atmospheric data, together with soil, and plant data in order to determine the inter-dependencies of variable values that both inform enhanced crop management practices and where possible, predict optimal growing conditions. The application of some novel data mining techniques together with the use of computational neural networks as a means to modeling and then predicting frost is the focus of the investigation described here as part of the wider project.

Keywords: Frost Prediction, Data Mining, Neural Networks.

1 Introduction

The effects of climate variation impacts on all forms of agriculture and vegetation. Due to the recent focus by many scientists and growers on the phenomenon known as ‘global warming’, reliable means for predicting climate change has gained relevance and importance. Viticulture is considered as a prime area for study because grapevine crops are among the most sensitive of any to climate or environmental variation.

The use of contemporary methods for mining data and knowledge discovery to classify and quantify the variables that are identified as being critical for the production of robust computational models is seen as both appropriate and necessary for this problem domain. This paper describes early results from work in frost prediction being undertaken by academic researchers at Auckland University of Technology in New Zealand, Universidad Católica del Maule in Chile and Universidad de Talca, Chile.

The Geoinformatics Research Centre at Auckland University of Technology in New Zealand, is working with scientists and industry in New Zealand, Chile, Uruguay, the United States (Sth California) and Japan to gather a large and complex

mixed data set that contains climate and atmospheric data, soil and plant data and human sensory perception data in order to ascertain the dependencies that exist between a large number of variables that effect the growth quality of grapes and production quality of wine. This project known as *Eno-Humanas* (see www.go-e-informatics.org) began in August 2007, so is in the early stages of placing sensors in numerous locations and establishing the infrastructure for data collection and processing within a methodology that seeks to correlate precise and imprecise data to produce scenarios for optimal grape growing conditions and also predicts certain factors that influence growth and production [1],[2],[3],[4]. The effects of climate variation impacts on all forms of agriculture and vegetation. Because of the recent focus by many scientists and growers on the phenomenon known as 'global warming', reliable means for predicting climate change has gained in relevance and importance especially in Viticulture as grapevine crops are among the most sensitive of any to climate or environmental variation.

Many of the previous studies [5], [6] that model and then predict phenomena of frost occurrence and impact consider only the available variables; usually temperature, humidity, wind velocity and precipitation. Some also include atmospheric pressure [7],[8]. It usually depends on the historical data that is held and the types of meteorological stations that are available. With this data it is possible to establish some indicators through various combinations of the variables that intervention should occur. See [9],[10],[11] for descriptions of these studies. In the research described in this paper, some derived values, especially the calculation of *dew point* (the point in time just before frost falls) are used and the *tendency* (such as a falling barometer and cloud cover, which determines solar intensity) are included to enrich the combinatorial analysis of the data.

We are using Computational Neural Network (CNN) methods to assist with the analysis and modeling of this problem. CNN technologies enable us to develop highly non-linear mathematical models with a relatively low level of computational complexity. With this model and data processing we can analyse large data sets still retain the integrity between the dependent variables while we construct scenarios for different result outcomes. The complexity of data dependencies is represented by a network of relationships, which we can 'train' to adapt to new data as it is ingested into the model without losing the intrinsic value of the variable dependencies. By reducing the relationship complexities we can reach a network size that is deeper in 'meaning' than the superficial data set might reflect and thus, we discover the extent to which the dependencies (factors relating to frost prediction in this case) really exist. The identification (or classification) of the variables present in such a model, using the Perceptron Multi-layer (MLP) method, is not in itself a sufficient analytical framework for complex data sets such as for the frost prediction problem. Using Kohonen's Self-Organising Map (SOM) techniques in conjunction with MLP does establish an appropriate level of participation by the variables in the phenomenon being studied. In this respect, we have used *Neucom* [www.kedri.aut.ac.nz] to implement a combinatorial approach to building the final model from the findings we have obtained through other data mining techniques. We propose this as a robust analytical platform for application to the frost prediction problem.

The World Map in Fig. 1 below shows the climate change potential for frost occurrence based on temperature variation related to latitude. This is an example of the

need to collect data from numerous locations and to compare it with other atmospheric and environmental data. Note the similarity between the latitude range 40-50°S as for Chile and New Zealand. The temperature at 10°C is present in both locations but varies in the same latitude spectrum for other longitudinal points around the globe. Most of these are located over the ocean. An interesting comparison is with Sth California, which is 10°C higher (20°C) at Latitude 30°N and Nth California, which is similar to Chile and New Zealand with 10°C at Latitude 40-50°N.

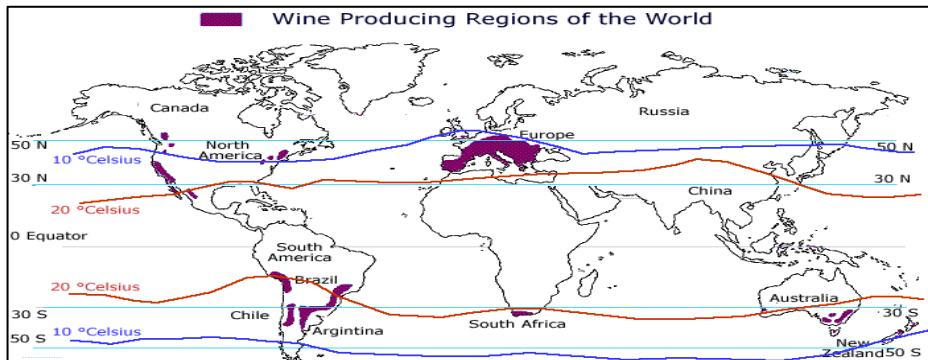


Fig. 1. Wine producing region of the world and climate change. Source: www.thirtyfifty.co.uk/spotlight-climate-change.asp.

2 Climate Sensor Data for Analysis Using CNN

For this early-stage study, data has been made available from the meteorological service collected for 2007 from the Region of O'Higgins in Chile (Fig. 2), provided by CRIA (*Regional Information Center Agrometeorológica*) of the Ministry of

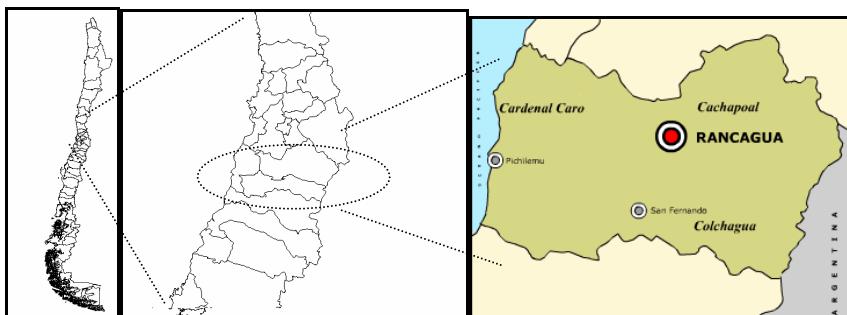


Fig. 2. A map of Chile indicating the O'Higgins Region, which is dominated by vineyards. The city of Santiago, which lies in a basin between high mountains and therefore, has a different climate, is 90kms to the North of this Region.

Agriculture of Chile. Using this data we can graph the weather trends and the following results (Fig. 3 and 4) relating to frost occurrence can be seen. Later in the research, it is intended to use comparative data from the network of weather stations set up by the Geoinformatics Research Group from Auckland University of Technology established in NZ, Chile, Uruguay, Southern California and Japan.

If the temperature is graphed 5 days prior to an event of frost, as shown in Fig. 3, there is a clear downward trend. Considering this, a first approximation to a prediction model of frost will take 48 hours prior to the event, and only until 3 pm.

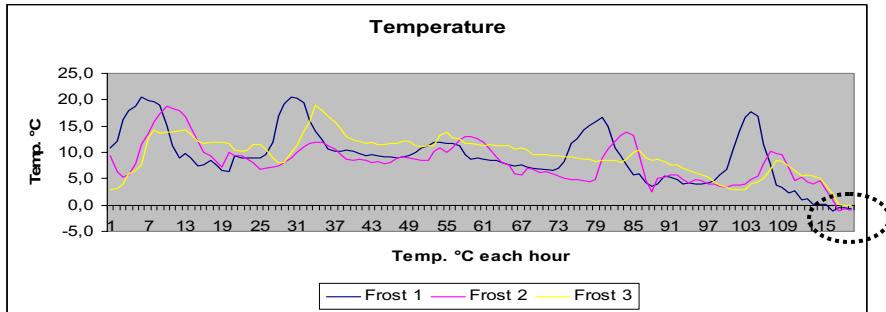


Fig. 3. Temperature measurements for 5 days, including three frost events where it is possible to observe a trend. In the dotted circle, there is a frost event.

Considering the suggestion proposed by Trought et al. [12] some results are shown below (Figure 4) of the last 48 hours till 3 pm prior to an event occurrence of frost. These results are graphed using the temperature and humidity variables only.

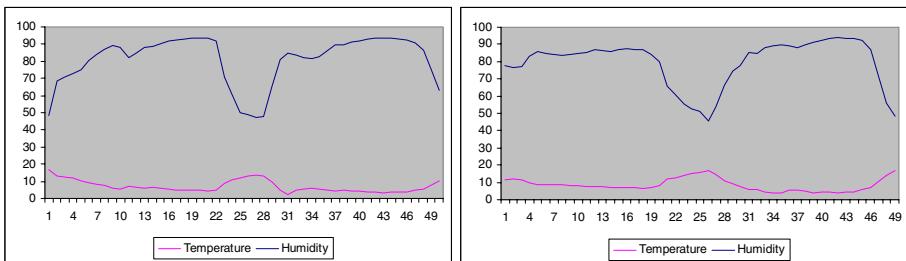


Fig. 4. Temperature and Humidity previous for two cases of frost. There are 48 data points until 3 p.m. in the period.

It can be seen here that it is not sufficient to only consider the variables of humidity and temperature for the prediction process and we propose therefore, to consider the dependency of other variables, such as precipitation, atmospheric pressure and wind velocity.

3 Data Dependency and Clustering

As stated in the paragraph above, to establish the relationship and dependency (correlation) between climatic variables that are preparing to establish a model to predict the frost is a highly complex. This article aims to establish a method to study the dependencies between variable value data (hence correlations) using neural networks, data mining techniques and SOM. This combinatorial approach will establish relations in the variables that will enable an analytical framework and methodology to be developed for work with those meteorological variables most relevant in accordance with a Multi-Layer Perceptron as shown in the generalized work flow model shown in Fig. 5 below.

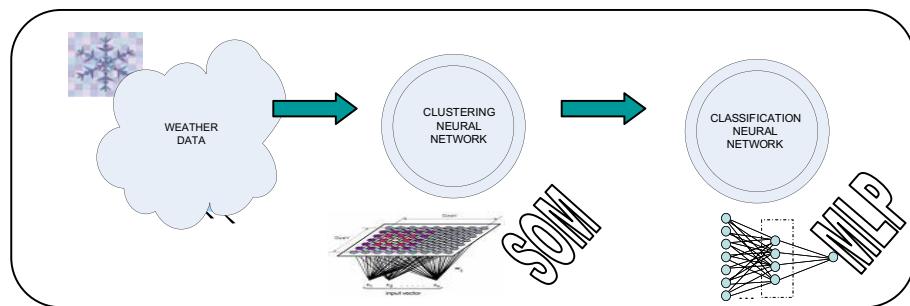


Fig. 5. Concept Proposal for the analysis of Weather Data using Self-Organizing Maps (SOM) in order to develop a Computational Multilayer Perceptron (MLP) model for Frost Prediction

SOMine Discovery [www.eudaptics.com] was used as the analytical software tool for the analysis of the correlation of variables illustrated in the previous model. We considered all the variables that are available to perform the analysis, i.e. temperature (in °C), RH (relative humidity%), wind speed (wind velocity km / hr), wind direction, wind speed (velocity) and dew point (at ° C). In this way the data can be analyzed (and correlated where possible) from different perspectives. The analysis of data is from the months of May and October.

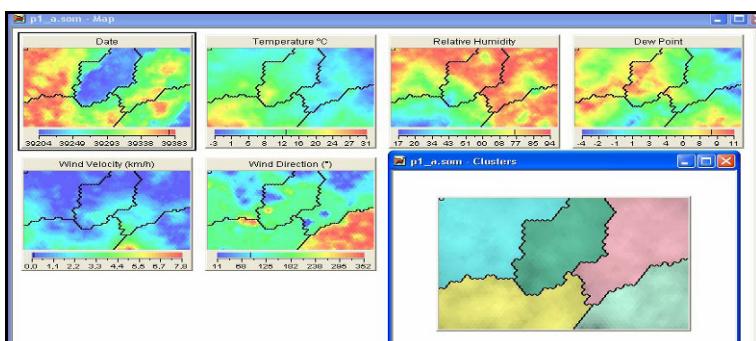


Fig. 6. SOM analysis of date, temperature, relative humidity, dew point, wind velocity and direction

A first step in the analytical approach described, is to consider the data by its date stamp, which includes day and time as a concatenated unique key (Date). This can be seen in Fig. 6.

In this first depiction it is possible to establish the relationship between temperature and dew point within the low temperatures (blue zone); low temperatures (frost events) are associated with very low wind speed. In order to determine a better defined clustering areas of low temperatures, we resorted to eliminating of some variables, such as the date in (Fig. 7a) and wind speed (Fig. 7b).

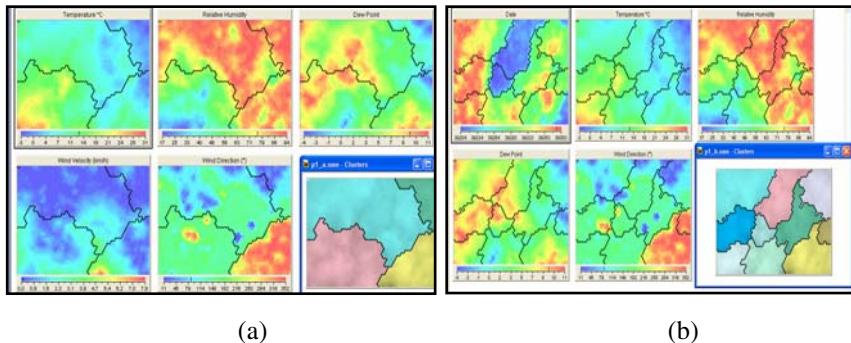


Fig. 7. SOM analysis of (a) temperature, relative humidity, dew point, wind velocity and direction; (b) date, temperature, relative humidity, dew point, wind direction

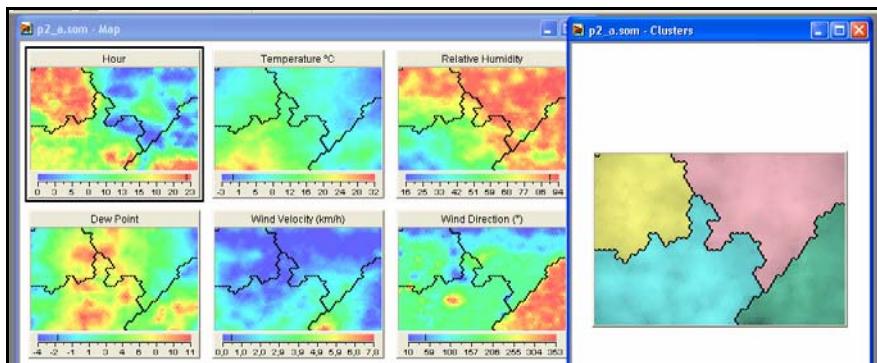


Fig. 8. SOM of hour, temperature, relative humidity, dew point, wind velocity and direction

A second analysis of the data relating to the time constraints (in hours), without date.

From the analysis of Fig. 8 we can see that selected areas of the coldest (blue zone) in the table of temperature, such as low as 1.1 ° C and based on this selection we analysed the hours in which frost occurred. This yields a histogram that is shown in Fig. 9a where schedules are established as a high probability of frost at 3,5 and 7am. The fig. 9b and 9c show histograms with the inclusion of the variables of humidity and wind direction.

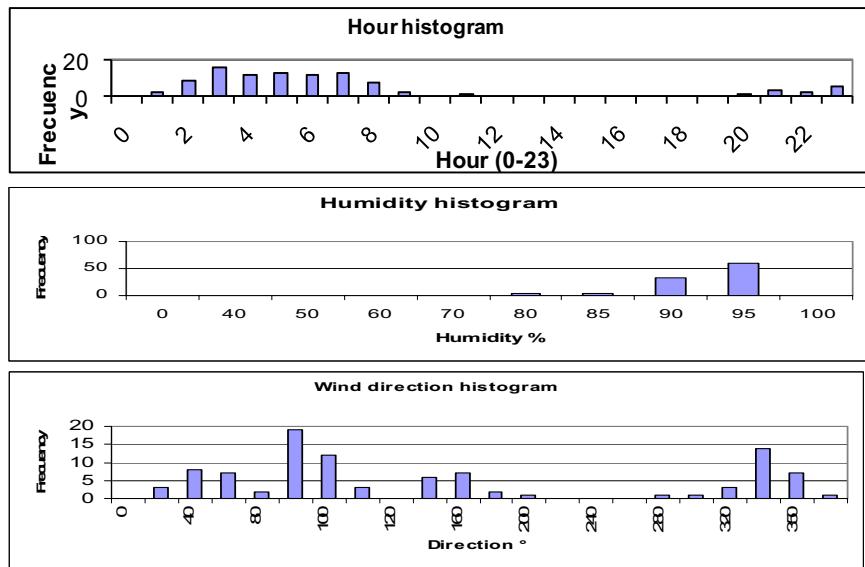


Fig. 9. Histogram analysis for (a) hour, (b) humidity and (c) wind direction for cold temperatures

After eliminating the wind speed factor, it is easier to identify classes (clusters) as is depicted in the SOM (Figure 10) below.

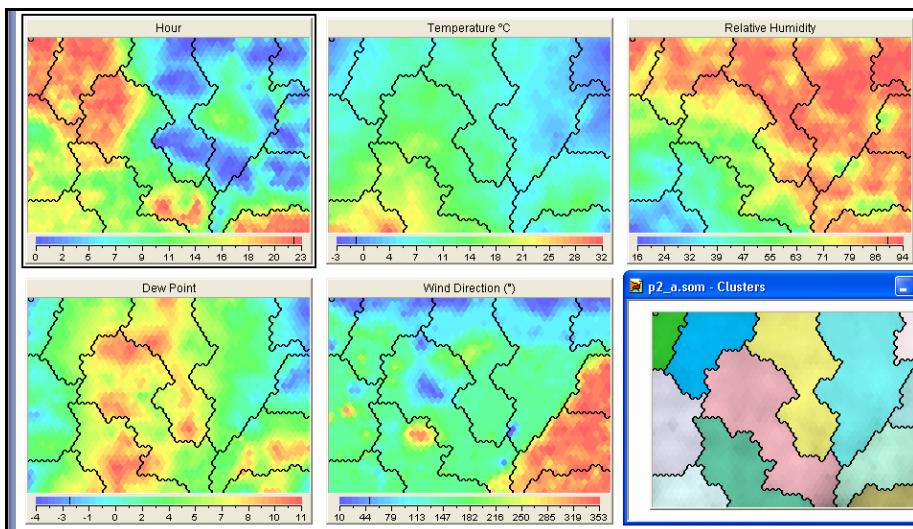


Fig. 10. SOM analysis considering the following variables: hour, temperature, relative humidity, dew point, wind direction

A third approach is to separate the data by time of day regardless of the other date and then to observe its correlation with each of the individual variables. The analysis represented here in Figure 11 relates to the third hour of increased frequency of frost, at the times of 3, 5 and 7 a.m.

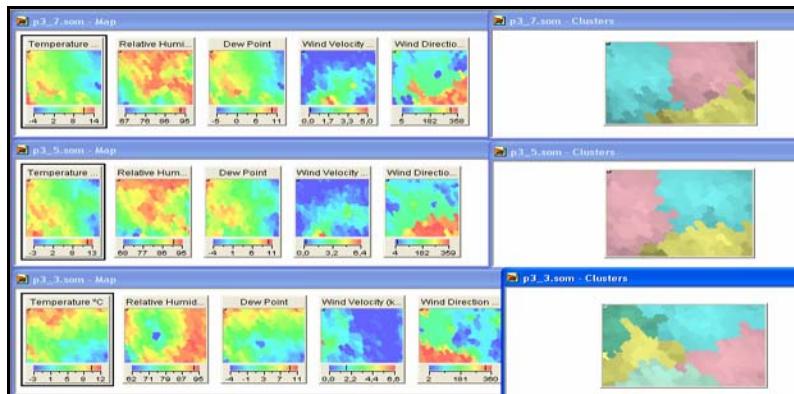


Fig. 11. Preliminary clustering analysis of weather data using SOM of 7, 5 & 3 am

The dependency between the dew point and the temperature variables can be observed in the fig. 6,7,8 and 9 in cold zone (blue zones).

In order to compare the results from our analysis using the SOM output exported to a graphing tool for trend depiction, we processed the data relating to humidity (as an example) using data mining software called *Mineset* [www. XXXX], which produced exactly the same statistical results thus, confirming our assumptions. The graphical depictions from *Mineset* are as shown in Fig. 12 below.

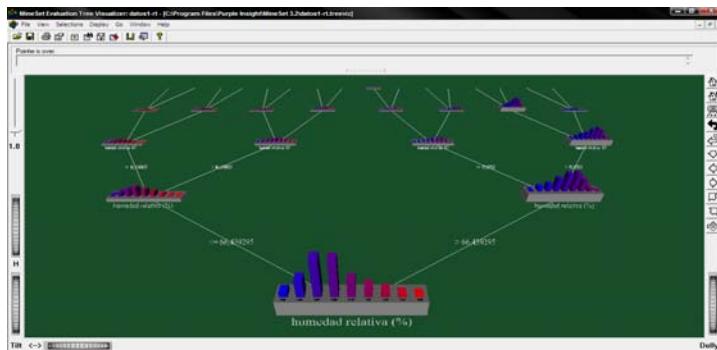


Fig. 12. MineSet [Vero Insight, <http://www.vero-insight.com/>]

The advantage of using the *Mineset* software after conducting the prior SOM analysis, is the better visualization it provides. However, when we are scaling the data in the preliminary analytical stage, *Mineset* can also assist with a visual representation, which makes the task of data preparation and understanding of the dynamics of

the data relationships considerably easier. These visualizations can be as histograms, scattergrams, three-dimensional (splat) diagrams or tree structures. These are all able to be animated in order to dynamically observe the data relationship distributions.

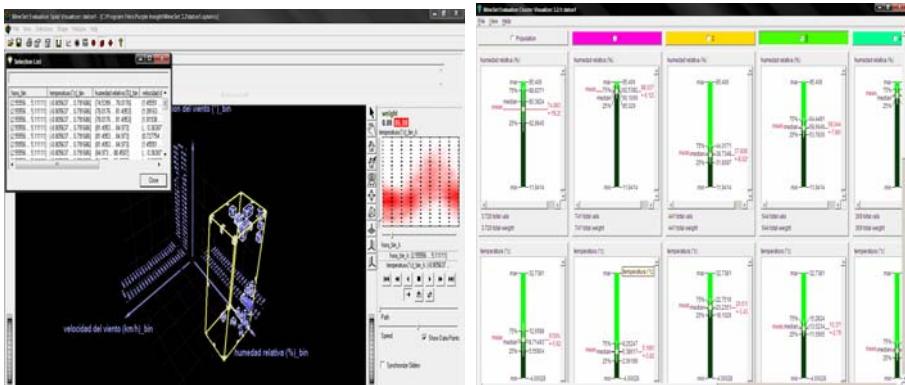


Fig. 13. Histogram, tree and network depiction examples from *Mineset*

4 Results and Conclusions

In summary from this work so far, have seen the following results:

- a correlation between the minimum temperature and dew point so we assume it is necessary to establish their relationship in the hours prior to the dew point event and therefore, we can eliminate one of them to simplify the actual prediction model.
- the wind speed in hours of cold is very low, so the ‘critical time’ prediction model can discard this variable unless a velocity in excess of the threshold level is recorded. An alarm can be set in the software analyzing the logged data, which can in turn inform the neural network and ingest this data into the scenario at the time.
- the wind direction in times of frost comes mainly from the South - East, namely the influence of the mountain range (Los Andes) in the valley of the Region of O’Higgins in Chile. This will be interesting to compare with the influence of the Southern Alps in New Zealand and similar geospatial factors for other locations in the future.
- the use of visualization software tools such as Viscovery and Mineset provides an analytical richness to the data processing task but also potentially could provide a meaningful depiction of the results for use in commercial operations such as in the vineyard.

It should be noted that one disadvantage of this preliminary study is the small amount of data available (only 1 year), so it is necessary to add more data to take more robust conclusions.

This study focused on the relationship of the variables in the hours of greatest cold (frost), so the next step is to filter the data by studying the cases of 24 hours prior to a frost and establish the relationship between the weather variables. This is in order to

generate a prediction model of a robust and reasonable complexity. There are also some indications observed here of the importance of cloud cover at times of extreme cold. Another component of the Eno-Humanas project is using image-processing techniques for this aspect of pattern recognition that we assume will add additional information to the model described here and being further developed as data is collected.

References

1. Sallis, P.J., Shanmuganathan, S., Pavesi, L., Jarur, M.: A system architecture for collaborative environmental modelling research. In: Samari, W.W., McQuay, W. (eds.) The 2008 International Symposium on Collaborative Technologies and Systems (CTS 2008), Irvine, California, pp. 39–47. IEEE, New Jersey (2008)
2. Shanmuganathan, S., Sallis, P.J., Pavesi, L., Jarur, M.: Computational intelligence and geo-informatics in viticulture. In: Al-Dabass, D., Turner, S., Tan, G., Abraham, A. (eds.) Proceedings of the Second Asia International Conference on Modelling & Simulation, Kuala Lumpur, Malaysia, pp. 480–485. IEEE computer society, Los Alamitos (2008) (CD version)
3. Shanmuganathan, S., Ghobakhloou, A., Sallis, P.: Sensors for modeling the effects of climate change on grapevine growth and wine quality. In: Proceeding of the 12th WSEAS International Conference on CIRCUITS (part of the 12th WSEAS CSCC Multi conference), Heraklion, Crete Island, Greece, pp. 315–321 (2008) (invited paper)
4. Sallis, P.J., Shanmuganathan, S., Pavesi, L., Jarur, M.: Kohonen Self-organising maps in the mining data mining of wine taster comments. In: Zanasi, A., Almorza Gomar, D., Ebecken, N.F.F., Brebbia, C.A. (eds.) Data Mining IX, Data Mining, Protection, Detection and other Security Technologies 2008, Cadiz, Spain, May 26–28 (2008); ISBN: 978-1-84564-110-8, ISSN(print): 1746-4463, ISSN(online): 1743-3517; Transactions on Information and Communication Technologies, 40, 125–139, WIT press (2008)
5. Ghielmi, L., Eccel, E.: Descriptive Models and Artificial Neural Networks for Spring Frost Prediction in an Agricultural Mountain Area. Computers and Electronics in Agriculture 54, 101–114 (2006)
6. Prabha, T., Hoogenboom, G.: Evaluation of Weather Research and Forecasting model for two Frost Events. Computers and Electronics in Agriculture 64, 234–247 (2008)
7. Snyder, R.L., de Melo-Abreu, J.P., Matulich, S. (eds.): Frost Protection: Fundamentals, Practice, and Economics. FAO Environmental and Natural Resources Series, Rome, Italy, vol. 1/2, 10 (2005)
8. Ovando, G., Bocco, M., Sayago, S.: Neural Networks for Modeling Frost Prediction. Agric. Téc. 65(1), 65–73 (2005)
9. Temeyer, B.R., et al.: Using Artificial Neural Network to Predict Parameters for Frost Deposition on Iowa Bridgeways. In: Proceedings of the 2003 Mid-Continent Transportation Research Symposium, Iowa (2003)
10. Robinson, C., Mort, N.: A Neural Network Solution to the Problem of Frost Prediction. In: UKACC International Conference on Control. Control 1996, UK, pp. 136–139 (1996)
11. Paras, et al.: A Feature Based Neural Network Model for Weather Forecasting. In: Proceedings of World Academy of Science, Engineering and Technology, vol. 23 (2007)
12. Trought, M.C.T., Howell, G.S., Cherry, N.: Practical Considerations for Reducing Frost Damage in Vineyards, Report to New Zealand Winegrowers (1999)

Personalized Modeling Based Gene Selection for Microarray Data Analysis

Yingjie Hu, Qun Song, and Nikola Kasabov

The Knowledge Engineering and Discovery Research Institute,
Auckland University of Technology, New Zealand
{raphael.hu,qsong,nikola.kasabov}@aut.ac.nz

Abstract. This paper presents a novel gene selection method based on personalized modeling. Identifying a compact set of genes from gene expression data is a critical step in bioinformatics research. Personalized modeling is a recently introduced technique for constructing clinical decision support systems. In this paper we have provided a comparative study using the proposed Personalized Modeling based Gene Selection method (PMGS) on two benchmark microarray datasets (Colon cancer and Central Nervous System cancer data). The experimental results show that our method is able to identify a small number of informative genes which can lead to reproducible and acceptable predictive performance without expensive computational cost. These genes are of importance for specific groups of people for cancer diagnosis and prognosis.

1 Introduction

Identifying a compact set of informative genes from microarray data (gene expression data) is critical in the construction of an efficient clinical decision support system. Microarray technology has been extensively studied for cancer classification and clinical outcomes prediction in response to cancer treatment, owing to its ability of monitoring the expression levels of thousands genes simultaneously. However, the data structure of a typical microarray dataset(gene expression dataset) is usually unbalanced - the number of genes far exceeds the sample size(several thousands *v.s.* several tens). Generally, only a small proportion of genes are able to contribute to the classification, while the rest of genes (known as noise genes) usually confuse the learning algorithms and models. Thus, discovering a set of most informative genes becomes a major task in microarray data analysis research.

In medical area, personalized modeling has significant potential to benefit the patients who need tailored treatment, such as cancer diagnosis and drug response. The main idea of personalized modeling is to create a model for each objective sample, which is able to discover the most important information specifically for this sample. Since personalized modeling focuses on the individual sample rather than simply on the global problem space, it can be more appropriate to build clinical decision support systems for new patients. Previous

work has reported that using personalized modeling can achieve better classification results than the results from global modeling [123].

This study proposes a *Personalized Modeling based Gene Selection method* (PMGS) for microarray data analysis. The organization of the remainder of this paper is as follows: Section 2 briefly reviews gene selection methods and personalized modeling. Section 3 describes the algorithm of the proposed personalized modeling based gene selection method(PMGS). Section 4 presents the experimental results on two benchmark microarray datasets - *Colon cancer and Central Nervous System cancer data*. Section 5 gives the conclusion and future direction.

2 Gene Selection Methods and Personalized Modeling

Many attempts have been made to identify which genes are most important to cancer diagnosis and prognosis task using microarray technology. Generally, most developed gene selection methods can be categorized into two groups, filter and wrapper methods. The measuring criterion in filter gene selection methods is created by measuring the intrinsic characteristic of genes. Many simple gene selection methods come from filter methods in which statistical algorithms and models are used, such as t-test, Fisher's linear discriminate, principal component analysis(PCA). Filter gene selection methods find the informative genes very fast, since the evaluation solely relies on the genes and is independent to any other particular learning algorithm.

Wrapper gene selection methods incorporate a learning model(usually a classification model) convolved with the target to find the minimized generalization error. The learning model evaluating the importance of genes is learnt and optimized during the gene selection process. The performance from wrapper methods is usually superior to that from filter methods, because the result comes from the optimized classification model. A standard wrapper gene selection method can be summarized as follows:

For a given training dataset $D = \{x_{ij}, y_i\} | x \in X, y \in Y, i = 1 \dots n, j = 1 \dots m$, pertaining to a pattern recognition task. x_{ij} and y_i denote the j^{th} gene's value of sample i and the class label of sample i , respectively. The optimized gene selection method including a classifier and a subset of genes are able to maximize the prediction accuracy, *i.e.* obtain the maximum correctness of the mapping from input set X to output set Y . Thus, a typical wrapper gene selection method is formulated in the following way to minimize the expected risk:

$$\Lambda(f_\sigma) = \int \lambda(y, f_\sigma(\sigma \bullet x)) dP(x, y) \quad (1)$$

where Λ is the expected risk, λ is a loss function, f_σ is a family of learning functions (*e.g.* a classifier or regression model), P is an evaluating function on the training data x , and σ denotes a vector indicating whether the gene $i (i = 1 \dots n)$ is selected or not.

Wrapper gene selection methods can generally yield high classification accuracy using a particular classifier with an expensive computational cost. In

wrapper method, the gene selection process is heavily dependent on a search engine, a search area (data), and an evaluation criterion for optimizing the learning model [4].

We have found in our previous microarray data experiments that the global modeling cannot provide precise and sufficient information for a new coming data vector under different circumstances, and also the selected subset of genes are not promising to be the biomarker genes. More importantly, it is difficult to incorporate previous developed models or existed knowledge into global modeling. For a new data sample, the whole (global) problem space usually contains much noise information that prevents the learning algorithm working properly on it, though the information may be useful for the global modeling. The noise information in the global problem space should be excluded to obtain a satisfactory result from the analysis.

As discussed in the introduction section, personalized modeling focuses on the individual sample rather than simply on the global problem space, so that creating a personalized problem space specifically for the new data can be a more appropriate solution to analyze new coming data sample in medical area. Personalized modeling is a relative new method in bioinformatics research, which is less found in literature. A representative work is published in [5].

One main difficulty in gene selection is how to optimize the learning function to evaluate the candidate genes during the training process. Genetic algorithm(GA) is a powerful method that is capable of exploring the combination of features and principally able to converge to the best solution. However, classical GA is often criticized for its huge computational cost and the difficulty of parameter setting.

Compact genetic algorithm(cGA) [6] is a GA based algorithm that drives the evolution towards a better probability distribution. Compared to classical GA, compact GA is able to discover and maintain the relationships between the features through the entire optimization, which creates a much faster discovery of the global optimum.

3 Algorithm of PMGS Method

In this study, we propose a new gene selection method based on personalized modeling for mciroarray data analysis, especially for cancer diagnosis and prognosis. In the proposed PMGS method, we employ wrappers to search candidate gene sets and then use the selected most important genes to profile individual data sample. This gene selection method can incorporate any classifier models for optimizing the learning function during the training process. In this study, we have investigated three classification algorithms, including Fuzzy KNN, Weighted distance K-nearest neighbor (WKNN) [5] and NaiveBayes to make a comparison.

Figure 1 is given to demonstrate the process of the new proposed PMGS method using wrapper method for searching candidate genes.

PMGS method starts with the creation of a personalized problem space (D_{pers}) for an objective data sample. A statistical model(SNR) selects a

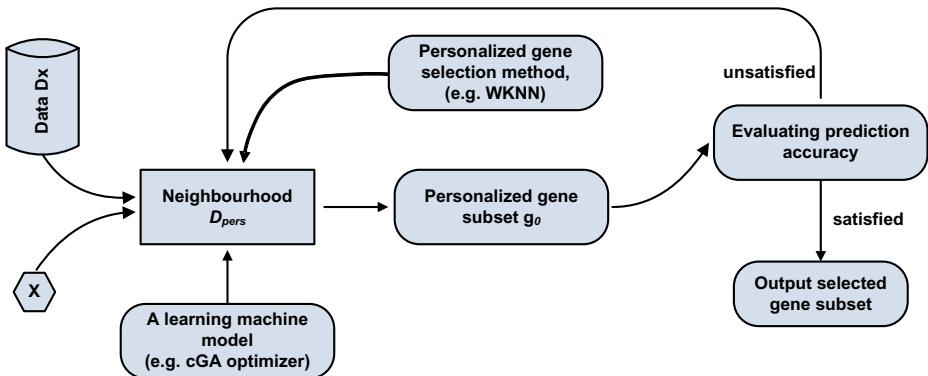


Fig. 1. A diagram of personalized modeling based gene selection method(PMGS)

subset of genes (several hundreds) based on their ranking scores from D_{pers} space. Then, these candidate genes are evaluated by a classifier, which is incorporated into a cGA based optimizer. This training process will be iterated until the stopping criterion is met. The final selected genes from cGA optimizer are then used to construct the classifier for validating the testing data sample. For simplicity, Algorithm 1 outlines the process of PMGS method.

Algorithm 1. Personalized Modeling based Gene Selection

Input: a new data vector x_v and a training dataset $D(m \times n)$:

1. Pre-process the given data vector x_v and the training dataset D (normalization and check missing values);
2. Use a statistical algorithm(e.g. SNR or t-Test) to rank all genes and obtain n top genes g_0 based on their ranking scores;
3. Create a personalized problem space D_{pers} for the new data x_v :

$$D_{pers} = F_{pers}(x_v, D)$$

F_{pers} is a function for computing the personalized problem space(the Euclidean distance based neighborhood).

4. Evaluate the candidate gene set g_0 over the personalized training data D_{pers} via classification models.
5. Use cGA to optimize the training process and select a set of genes g_{sel} that lead to the best classification performance.
6. Validate the selected genes g_{sel} on the new data x_v .

Note: n is suggested from 50 to 200.

4 Experiment

The goal of this study is to design a model to select a compact set of genes that can profile the pattern of objective microarray data.

4.1 Data

Two benchmark cancer gene expression datasets (microarray data) are used in this study:

1. Colon cancer data [7]

(available at <http://microarray.princeton.edu/oncology/>)

This data consists of 62 samples collected from colon-cancer patients; 40 samples are labeled as diseased and 22 are labeled as normal. Only 2,000 genes out of total 6,500 genes are selected into the dataset based on the confidence in the measured expression levels.

2. Central Nervous System (CNS) cancer data [8]

(available at <http://www-genome.wi.mit.edu/mpr/CNS/>)

This benchmark data contains 60 patient samples, in which 39(class 1) are medulloblastoma survivors and 21 are treatment failures(class 0). The learning objective of this gene expression data is to classify the patients who survive after the treatment and those who are succumbed to the central nervous system cancer. Each patient sample is represented by 7,129 probes from 6,817 human genes.

4.2 Experimental Setup

The bias issue in the supervised classification experiments for microarray data analysis has been intensively criticized, because many published impressive results cannot be reproduced by other laboratories [9][10]. To establish a validity of PMGS method, our experiment is accomplished using a leave-one-out cross-validation (LOOCV) approach. Performance is assessed by overall classification accuracy. The gene selection, classifier construction and optimization is based on the training data (the rest of data samples). In each LOOCV iteration, the leave-one-out testing data is only used for validating the selected genes from the training process and measuring the predicted result. In our experiment, the leave-one-out testing data sample is completely independent to the training process, *i.e.* the classlabel of the testing data keeps unknown until the predicted result of testing data is computed. We use this validation schema to maximally decrease the bias error, and ensure the experimental results replicable.

Three classifier models - WKNN, Fuzzy KNN and Naive Bayes are adopted as the evaluating function in algorithm 1- Step 1. The training process is optimized by a cGA evolutionary model and the most informative genes are selected when the optimized solution is obtained.

4.3 Results

The experiment results from colon and CNS cancer data through the unbiased validation schema are encouraging. Table 1 illustrates the classification accuracy obtained using the selected genes by WKNN, Fuzzy KNN and Naive Bayes classifiers with cGA optimization under the same experiment environment. For

Table 1. The best classification result from Colon and CNS cancer data

Data Set	Classifier	Overall Acc(%)	Class 1/2 (%)	Reported result in the original paper(%)
Colon	WKNN	87.10	92.50 / 77.27	83
	Fuzzy KNN	70.00	61.90 / 74.36	
	Naive Bayes	85.48	90.00 / 77.27	
CNS	WKNN	70.00	61.90 / 74.36	78.33
	Fuzzy KNN	78.33	71.43 / 82.05	
	Naive Bayes	63.33	47.62 / 71.78	

colon data, the 10 most frequently selected genes are: *gene 625, 1423, 802, 897, 245, 267, 1771, 698, 187* and *1235*. For CNS data, the top 10 genes are found most important: *gene 6252, 5812, 1352, 2496, 2474, 2996, 6064, 4576, 844* and *5871*. The result from PMGS method with WKNN classifier over colon data is better than the published benchmark result. For CNS data, PMGS method obtains the same classification accuracy in literature.

For each data sample, the final selected most important genes may be different. The selected frequency of some genes is significantly high, which means they can be recognized highly representative of the data pattern. The experimental result is reproducible, and does not require intensive computational cost (approximate 40mins running on a desktop with a Pentium IV cpu and 2G memory, computation time is dependent on the complexity of classifier used in the training process).

The main goal of developing PMGS method is to discover the personalized information for each sample (can be a patient tissue sample), rather than simply to compare the classification accuracy with published results in literature. For this purpose, PMGS is designed to be able to give a detailed profile for the new testing data sample(a new patient sample), which can contribute to clinical decision support system. Here we give an example to demonstrate how PMGS visualizes the analysis result from a data sample(a patient). For the sample 7 in colon data, PMGS method selects 23 genes and the classifier successfully predicts sample 7 diseased. To help visualize the result, we plot 36 nearest neighbors of sample 7 of colon data in a 3-D space (3 most important genes) (see Fig 2-(a)). It is easy to elucidate that sample 7 is more likely to be in the diseased group, since most of its nearest neighbors belong to diseased group. (*Note:* the value on *x* and *y* axis is normalized to [0, 1]); Fig 2-(b) gives a scenario of the gene expression level improvement for a patient(here is sample 7 of colon cancer data), where *x* axis represents the gene index number and *y* axis represents the gene expression level value.

In order to recover from the disease, the patient should receive a personalized medical treatment tailored for him/her. Fig 2 and table 2 give an example for designing a personalized medical treatment for a colon cancer patient (data sample 7) using PMGS model, Table 2 shows that the patient(sample 7) needs

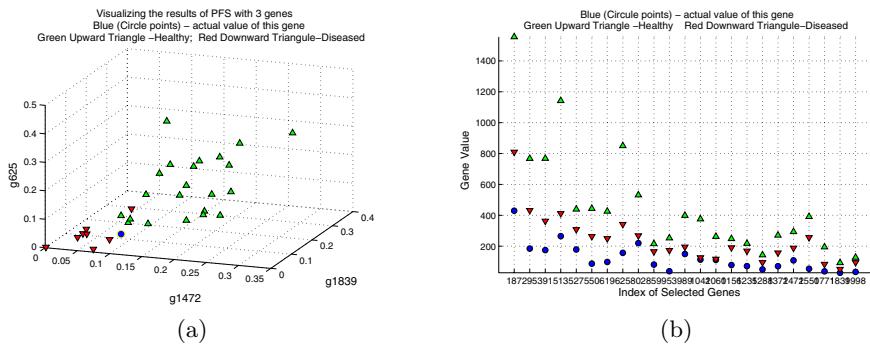


Fig. 2. The profile of sample 7 in Colon cancer data. (a). A 3-D visualization of the neighborhood of data sample 7 using the 3 most important genes(Gene 1472, 1839 and 625); (b). A scenario of the gene expression level improvement for sample 7.

Table 2. A scenario of the potential improvement for a Colon cancer patient (sample 7)

Index of Gene	Gene EST Number	Actual value	Desired average profile	Desired Improvement	Weighted importance
G1472	L41559	108.1480	293.5720	185.4240	0.0613
G1839	X07290	28.2240	92.3052	64.0812	0.0511
G625	X12671	157.0260	850.5171	693.4911	0.0509
G802	X70326	219.7150	531.3680	311.6530	0.0476
G1235	U29175	71.1390	217.7371	146.5981	0.0459
...

to change the expression level of gene 1472 (EST No: L41559) from 108.1480 to 293.5720 (Note: EST is the Expressed Sequence Tag of a gene, which is a unique index that can be retrieved from *GenBank*). Due to the length limitation of the paper, table 2 only lists the improvement scenario of top five genes.

5 Conclusion

The main contribution of the proposed PMGS method is having introduced a new approach of gene selection using personalized modeling technique, especially for complex disease diagnosis. More importantly, our proposed method can discover the information and give a detailed profile specifically for a data sample based on the selected most informative genes. The proposed method has the significant potential for clinical practitioners to design tailored treatment for a patient, *e.g.* the prediction in response to drug usage.

Although the experiment results obtained from PMGS method are promising, more further investigations are required in the future study, *e.g.* it may lead us to have a better understanding to investigate the biological reference of the selected important genes obtained from different gene expression datasets. It will be one of our future direction to discover the genes that are most commonly used for the patients from different types of cancers.

References

1. Nevins, J.R., Huang, E.S., Dressman, H., Pittman, J., Huang, A.T., West, M.: Towards integrated clinico-genomic models for personalized medicine: combining gene expression signatures and clinical factors in breast cancer outcomes prediction. *Human Molecular Genetics* 12(2), R153–R157 (2003)
2. Song, Q., Kasabov, N.: Twnfi - a transductive neuro-fuzzy inference system with weighted data normalization for personalized modeling. *Neural Networks* 19(10), 1591–1596 (2006)
3. Hu, Y., Kasabov, N.: ntology-based framework for personalized diagnosis and prognosis of cancer based on gene expression data. In: ICONIP 2007 14th International Conference on Neural Information Processing, Kitakyushu City, Fukuoka, Japan, vol. 2, pp. 846–855 (2007)
4. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *The Journal of Machine Learning Research* 3, 1157–1182 (2003)
5. Kasabov, N.: Global, local and personalized modelling and pattern discovery in bioinformatics: An integrated approach. *Pattern Recognition Letters* 28, 673–685 (2007)
6. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. *IEEE Trans. Evolutionary Computation* 3(4), 287–297 (1999)
7. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci., USA* 96, 6745–6750 (1999)
8. Pomeroy, S., Tamayo, P., et al.: Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature* 415(6870), 422–436 (2002)
9. Ransohoff, D.F.: Bias as a threat to the validity of cancer molecular-marker research. *Nat. Rev. Cancer* 5(2), 142–149 (2005)
10. Ioannidis, J.P.A.: Microarrays and molecular research: noise discovery? *Lancet* 365, 453–455 (2005)

Integrated Feature and Parameter Optimization for an Evolving Spiking Neural Network

Stefan Schliebs¹, Michaël Defoain-Platel², and Nikola Kasabov¹

¹ Knowledge Engineering and Discovery Research Institute,
Auckland University of Technology, New Zealand
`{sschlieb, nkasabov}@aut.ac.nz`

² Department of Computer Science, University of Auckland
`michael.dp@cs.auckland.ac.nz`

Abstract. This study extends the recently proposed Evolving Spiking Neural Network (ESNN) architecture by combining it with an optimization algorithm, namely the Versatile Quantum-inspired Evolutionary Algorithm (vQEA). Following the wrapper approach, the method is used to identify relevant feature subsets and simultaneously evolve an optimal ESNN parameter setting. Applied to carefully designed benchmark data, containing irrelevant and redundant features of varying information quality, the ESNN-based feature selection procedure lead to excellent classification results and an accurate detection of relevant information in the dataset. Redundant and irrelevant features were rejected successively and in the order of the degree of information they contained.

1 Introduction

In many recent studies attempts have been made to use Spiking Neural Networks (SNN) for solving practical real world problems. It was argued that SNN have at least similar computational power than the traditional Multi-Layer-Perceptron derivates [1]. Substantial progress has been made in areas like speech recognition [2], learning rules [3] and associative memory [4]. In [5] an evolving SNN (ESNN) was introduced and applied to pattern recognition problems, later this work was extended to speaker authentication tasks and even to audio-visual pattern recognition [6]. A similar spiking neural model was analyzed [7], in which a classification problem for taste recognition was addressed. Based on a simple but efficient neural model, these approaches used the ESNN architecture, which was trained by a fast one-pass learning algorithm. Due to its evolving nature the model can be updated whenever new data becomes available, without requiring the re-training of earlier presented data samples. Some promising results could be obtained both on synthetic benchmark and real world datasets.

This study investigates the potential of ESNN when applied to Feature Subset Selection (FSS) problems. Following the wrapper approach the ESNN architecture combined with an evolutionary algorithm. The latter one is used to identify relevant feature subsets and simultaneously evolve an optimal parameter setting for the ESNN, while the ESNN itself operates as a quality measure for a presented feature subset. By optimizing two search spaces in parallel it is expected to evolve an ESNN configuration, specifically generated for the given dataset and a specific feature subset, that maximizes classification accuracy.

2 ESNN Architecture for FSS

The ESNN architecture uses a computationally very simple and efficient neural model, in which early spikes, received by a neuron, are stronger weighted than later ones. The model was inspired by the neural processing of the human eye, which performs a very fast image processing. Experiments have shown that a primate only needs several hundreds of milliseconds to make reliable decisions about images that were presented in a test scenario [8]. Since it is known that neural image recognition involves several succeeding layers of neurons, these experiments suggested that only very few spikes could be involved in the neural chain of image processing. In [9] a mathematical definition of these neurons was attempted and tested on some face recognition tasks, reporting encouraging experimental results. The same model was later used by [6][10] to perform audio-visual face recognition.

Similar to other SNN approaches a specific neural model, a learning method, a network architecture and an encoding from real values into spike trains needs to be defined in the ESNN method. The neural model is given by the dynamics of the post-synaptic potential (PSP) of a neuron i :

$$PSP_i(t) = \begin{cases} 0 & \text{if neuron has fired} \\ \sum_{j|f(j) < t} w_{ji} \times (m_i)^{order(j)} & \text{else} \end{cases} \quad (1)$$

where w_{ji} is the weight of a pre-synaptic neuron j , $f(j)$ the firing time of j , and $m_i \in (0, 1)$ a parameter of the model, namely the modulation factor. Function $order(j)$ represents the rank of the spike emitted by neuron j . For example a rank $order(j) = 0$ would be assigned, if neuron j is the first among all pre-synaptic neurons that emits a spike. In a similar fashion the spikes of all pre-synaptic neurons are ranked and then used in the computation of PSP_i . A neuron i fires a spike when its potential has reached a certain threshold θ . After emitting a spike the potential is reset to $PSP_i = 0$. Each neuron is allowed to emit only a single spike at most. The threshold $\theta = c PSP_{max}$ is set to a fraction $c \in (0, 1)$ of the maximal potential PSP_{max} possible by a neuron.

An evolving neural network architecture using the above model along with a learning algorithm was proposed in [5]. The method 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 to the ones in the repository (according to a specified similarity threshold s), the neuron will be merged with the most similar one. Otherwise the trained neuron is added to the repository as a new output neuron. Because of the incremental evolution of output neurons it is possible to accumulate knowledge as it becomes available. Hence a trained network is able to learn new data without the need of re-training the already learned samples. The procedure is described in detail in Algorithm I.

Encoding of input values seems to be a critical factor in all SNN approaches. Several encoding mechanisms for SNN have been proposed, such as frequency mappings, Poisson processes and rank order encoding. Another approach is the population encoding which distributes a single input value to multiple neurons and hence may cause the excitation and firing of several responding neurons. Our implementation is based on

Algorithm 1. Training an Evolving Spiking Neural Network (ESNN)

Require: $m_l \in (0, 1)$, $s_l \in (0, 1)$, $c_l \in (0, 1)$, $l \in L$

1. initialize neuron repository $R_l = \{\}$
2. **for all** samples $X^{(i)}$ belonging to class l **do**
3. $w_j^{(i)} \leftarrow (m_l)^{order(j)}$, $\forall j | j$ pre-synaptic neuron of i
4. $PSP_{max}^{(i)} \leftarrow \sum_j w_j^{(i)} (m_l)^{order(j)}$
5. $\theta^{(i)} \leftarrow c_l PSP_{max}^{(i)}$
6. **if** $\min(d(w^{(i)}, w^{(n)})) > s_l$, $w^{(n)} \in R_l$ **then**
7. $w^{(n)} \leftarrow \text{merge } w^{(i)} \text{ and } w^{(n)}$
8. $\theta^{(n)} \leftarrow \text{merge } \theta^{(i)} \text{ and } \theta^{(n)}$
9. **else**
10. $R_l \leftarrow R_l \cup \{w^{(i)}\}$
11. **end if**
12. **end for**

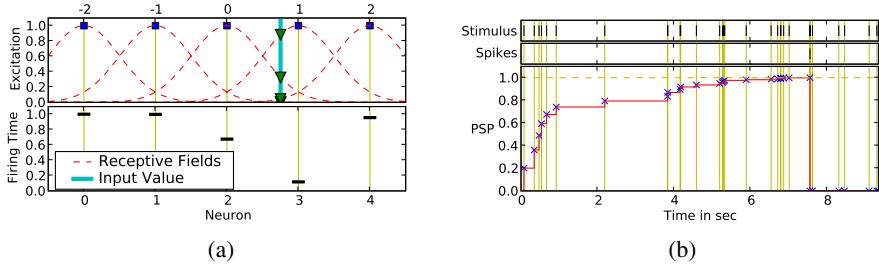


Fig. 1. a) Population encoding based on Gaussian receptive fields. For an input value $v = 0.75$ (thick straight line in top left figure) the intersection points with each Gaussian is computed (triangles), which are in turn translated into spike time delays (lower left figure). b) Evolution of the PSP of the neural model for a given input stimulus. If the potential reaches threshold θ a spike is triggered and the PSP set to 0 for the rest of the simulation, even if the neuron is still receiving incoming spike trains.

arrays of receptive fields as described in [3], cf. Figure 1. Receptive fields allow the encoding of continuous values by using a collection of neurons with overlapping sensitivity profiles. The method is well studied and constitutes biological plausibility. Each input variable is encoded independently by a group of M one dimensional receptive fields. For a variable n an interval $[I_{min}^n, I_{max}^n]$ is defined. The Gaussian receptive field of neuron i is given by its center $\mu_i = I_{min}^n + (2i - 3)/2 * (I_{max}^n - I_{min}^n)/(M - 2)$ and width $\sigma = 1/\beta(I_{max}^n - I_{min}^n)/(M - 2)$, with $1 \leq \beta \leq 2$. Parameter β directly controls the width of each Gaussian receptive field. See Figure 1 for an example encoding of a single variable.

The described ESNN method is summarized in Figure 2a. In this study it was used to address FSS problems following the well known wrapper approach. A wrapper contains a general optimization algorithm interacting with an induction method (classifier). The optimization task consists in a proper identification of an optimal feature subset,

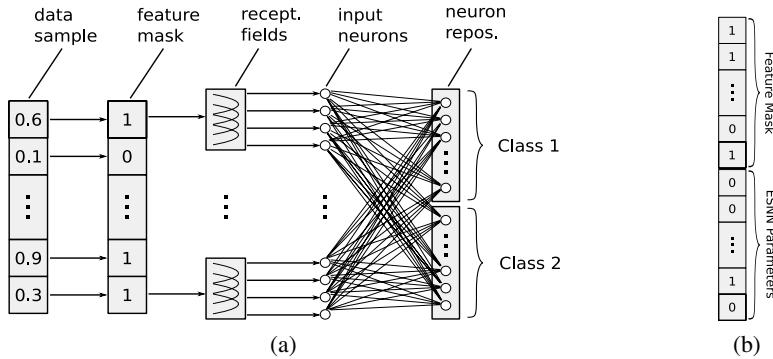


Fig. 2. a) ESNN architecture – A data sample is masked to extract a feature subset, then each variable is translated into trains of spikes. The resulting spike sequence invokes a spiking neural network and a repository of output neurons is successively generated during the training process.
b) Chromosome used in vQEA for simultaneously optimizing feature and parameter space.

which maximizes the classification accuracy determined by the inductor. The ESNN architecture will operate as the induction method during the course of this paper. Due to its interesting properties in terms of solution quality and convergence speed we decided to use the previously proposed Versatile Quantum-inspired Evolutionary Algorithm (vQEA) [11] as the optimization algorithm. The method evolves in parallel a number of independent probability vectors, which interact at certain intervals with each other, forming a multi-model Estimation of Distribution Algorithm (EDA) [12]. It has been shown that this approach performs well on epistatic problems, is very robust to noise, and needs only minimal fine-tuning of its parameters. In fact the standard setting for vQEA is suitable for a large range of different problem sizes and classes. Finally vQEA is a binary optimizer and fits well to the feature selection problem we want to apply it on.

Manual fine-tuning the neuronal parameters can quickly become a challenging task [10]. To solve this problem the idea of the simultaneous optimization of the two combinatorial search problems of FSS and learning of parameters for the induction algorithm was proposed [13]. The selection of the fitness function was identified to be a crucial step for the successful application of such an embedded approach. In the early phase of the optimization the parameter configurations are selected randomly. As a result it is very likely that a setting is selected for which the classifier is unable to respond to any input presented, which corresponds to flat areas in the fitness landscape. Hence a configuration that will allow the network to fire (even if not correctly) represents a huge (local) attractor in the search space, which could be difficult to escape in later iterations of the search. In [13] a linear combination of several sub-criteria was used to avoid a too rugged fitness landscape. Nevertheless we can not confirm, that the use of much simpler fitness functions led to any problems in our experiments. Using the classification accuracy on testing samples seemed to work well as it is presented later in this paper. All parameters modulation factor m_l , similarity threshold s_l , PSP fraction c_l , $\forall l \in L$ of ESNN were included in the search space of vQEA. Due to its binary nature vQEA requires the conversion of bit strings into real values. We found that a small number of

Grey-coded bits were sufficient to approximate meaningful parameter configurations of the ESNN method. In Figure 2b the structure of a chromosome as it is used in vQEA is depicted.

3 Experiments

We have applied the vQEA optimised Evolving Spiking Neural Network (ESNN) architecture on the Two-Spiral problem firstly introduced in [24]. It is composed of two-dimensional data forming two intertwined spirals. It requires the learning of a highly non-linear separation of the input space. The data was frequently used as a benchmark for neural networks, including the analysis of the ESNN method itself [10]. Since the data contains only two relevant dimensions we have extended it by adding redundant and random information. The importance of the redundant features was varied: Features range from mere copies of the original two spirals to completely random ones. The information available in a feature decreases when stronger noise is applied. The generation of the dataset is particularly interesting, since it is expected that the ESNN is capable of rejecting features according to their inherent information, *i.e.*, the less information a feature carries, the earlier ESNN should be able to discard the feature during the selection process. We will briefly summarize the data generation below.

Data points belonging to two intertwined Archimedean spirals (also known as the arithmetic spiral) were generated and labelled accordingly. The irrelevant dimensions consist of random values chosen from a uniform distribution, covering the entire input space $[-1, 1]$ of the dataset. The redundant dimensions are represented by copies of the original spiral points $p = (x, y)^T$, which were disturbed by a Gaussian noise using standard deviation $\sigma = |p| * s$, with $|p|$ being the absolute value of vector p and s a parameter controlling the noise strength. The noise increases linearly for points which are more distant from the spiral origin $(0, 0)^T$. A noisy value p'_i is then defined as the outcome of the p_i -centered Gaussian distributed random variable $\mathcal{N}(p_i, \sigma^2)$, using σ as defined above.

Our final dataset contained seven redundant two-dimensional spiral points $(x'_i, y'_i)^T$, for each a different noise strength parameter $s \in \{0.2, 0.3, \dots, 0.8\}$ was used, totalling in 14 redundant features. Additional four random features r_1, \dots, r_4 were included. Together with the two relevant features of the spirals (x and y) the dataset contained 20 features. Figure 3 presents the 400 generated samples of the resulting dataset.

For vQEA we chose a population structure of ten individuals organized in a single group, which is globally synchronized every generation. This setting was reported to be generally superior for a number of different benchmark problems [12]. The learning rate was set to $\theta = \pi/100$ and the algorithm was allowed to evolve over a total number of 400 generations. In order to guarantee statistical relevance 30 independent runs were performed, using a different seed for each of them.

Additional to the feature space, vQEA was used to optimize the parameter space of the ESNN architecture. For each class $l \in L$ three parameters exist: The modulation factor m_l , the similarity threshold s_l , and the proportion factor c_l . Since the data represents a two-class problem, six parameters are involved in the ESNN framework. The binary character of vQEA requires the conversion of bit strings into real values. In the

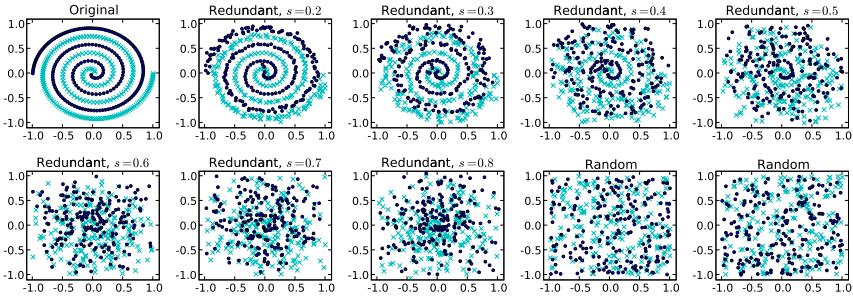


Fig. 3. The different features of the generated synthetic dataset for investigating ESNN in the context of a FSS problem. The colors/symbols represent the class label of a given data point. Each figure shows two features (x - and y -axis), all features are combined to form the complete experimental dataset.

experiments we found four bits per variable enough to offer sufficient flexibility for the parameter space. For the conversion itself a Grey code was used.

In terms of the population encoding we found that especially the number of receptive fields needs careful consideration, since it affects the resolution for distinguishing between different input variables. After some preliminary experiments we decided for 20 receptive fields, the centers uniformly distributed over the interval $[-1, 1]$, and the variance controlling parameter $\beta = 1.5$.

In every generation the 400 samples of the dataset were randomly shuffled and divided into 300 training and 100 testing samples. The chromosome of each individual in the population was translated into the corresponding parameter and feature space, resulting in a fully parameterized ESNN and a feature subset. The ESNN was then trained and tested on the appropriate data subsets. For the computation of the classification error we determined the ratio between correctly classified samples and the total number of testing samples.

3.1 Results

In Figure 4a the evolution of the average best feature subset in every generation is presented. The lighter the color the more often the corresponding feature was selected in a specific run at the given generation. First of all, each of the 30 runs identified the two relevant features very accurately, but particular interesting is the order in which the features have been discarded by the algorithm. The four random features r_1, \dots, r_4 containing no information were almost immediately rejected in less than 20 generations. The redundant features x'_i, y'_i were rejected one after the other, according to the strength of the noise applied: The higher the noise the earlier a feature could be identified as irrelevant. Some runs struggled to reject the features x'_0 and y'_0 perturbed by the smallest noise strength $s = 0.2$. The average number of features selected decreases steadily in later generations, but the trend line in Figure 4b suggests the evolution is not completely finished, yet. On the other hand the classification accuracy has reached a satisfyingly high level in later generations, cf. Figure 4c. The average accuracy reported by each

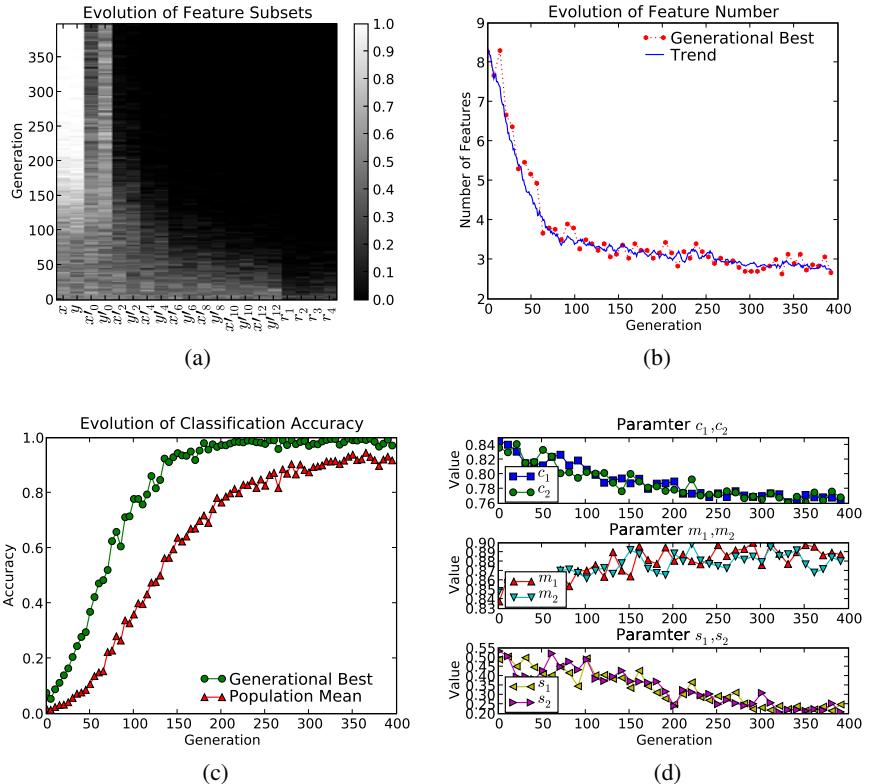


Fig. 4. Results on a synthetic spiral data set averaged over 30 runs using different random seeds for the optimization algorithm. The two relevant features were identified by all of the 30 runs (a). The number of features decreases with increasing generations (b). On the same time the SNN classifier delivers a good estimate of the quality of the presented feature subset (c). Only if most of the noisy features have been discarded an optimal accuracy is reported. Along with the features also the parameters of the SNN model are optimized (d).

individual in the population was constantly above 90%. Parameter optimization using all of the features delivered a very poor average accuracy of $< 10\%$, since the trained network was unable to respond for most of the test samples presented.

Figure 4d presents the evolution of the parameters of the ESNN architecture. Usually the values for modulation, merging and spike threshold are pairwise very close to each other. We take this as an indicator that vQEA indeed controlled these parameters carefully, since different values for these pairs would be meaningless in this well-balanced dataset. All three pairs display a steady trend and evolve constantly towards a certain optimum, not reporting too much variability.

4 Conclusion and Future Work

In this study we have presented an extension for ESNN by accompanying it with an evolutionary algorithm, which simultaneously evolves an optimal feature subset along with an optimal parameter configuration for ESNN. Here we used on already tested and published quantum-inspired evolutionary algorithm [11]. The method was tested on benchmark data for which the global optimum was known *a priori*. The obtained results are promising and encourage further analysis of more realistic scenarios. Especially the meaning and impact of each of the ESNN parameters require a better understanding and should be investigated in detail in future studies. ESNN needs to be compared to similar approaches in order to identify its potential advantages and/or disadvantages on specific problem classes. Finally the use of a real-valued parameter optimization in addition to the binary feature search should be considered.

References

1. Maass, W.: Computing with spiking neurons (1999)
2. Verstraeten, D., Schrauwen, B., Stroobandt, D.: Isolated word recognition using a liquid state machine. In: ESANN, pp. 435–440 (2005)
3. Bohte, S.M., Kok, J.N., Pouthré, J.A.L.: Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48(1-4), 17–37 (2002)
4. Knoblauch, A.: Neural associative memory for brain modeling and information retrieval. Inf. Process. Lett. 95(6), 537–544 (2005)
5. Wysoski, S.G., Benuskova, L., Kasabov, N.: On-line learning with structural adaptation in a network of spiking neurons for visual pattern recognition. In: Kollia, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 61–70. Springer, Heidelberg (2006)
6. Wysoski, S.G.: Evolving Spiking Neural Networks for Adaptive Audiovisual Pattern Recognition. PhD thesis, Auckland University of Technology (August 2008), <http://hdl.handle.net/10292/390>
7. Soltic, S., Wysoski, S., Kasabov, N.: Evolving spiking neural networks for taste recognition. In: IEEE World Congress on Computational Intelligence (WCCI), Hong Kong (2008)
8. VanRullen, R., Thorpe, S.J.: Is it a bird? is it a plan? ultra-rapid visual categorisation of natural and artificial objects. Perception 30, 655–668 (2001)
9. Thorpe, S.J.: How can the human visual system process a natural scene in under 150ms? experiments and neural network models. In: ESANN (1997)
10. Wysoski, S., Benuskova, L., Kasabov, N.: Brain-like evolving spiking neural networks for multimodal information processing. In: ICONIP 2007. LNCS. Springer, Heidelberg (2007)
11. Defoin-Platel, M., Schliebs, S., Kasabov, N.: A versatile quantum-inspired evolutionary algorithm. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 423–430 (2007)
12. Defoin-Platel, M., Schliebs, S., Kasabov, N.: Quantum-inspired evolutionary algorithm: A multimodel eda. IEEE Transactions on Evolutionary Computation (in print, 2009)
13. Valko, M., Marques, N.C., Castelani, M.: Evolutionary feature selection for spiking neural network pattern classifiers. In: Bento, et al. (eds.) Proceedings of 2005 Portuguese Conference on Artificial Intelligence, pp. 24–32. IEEE, Los Alamitos (2005)
14. Lang, K.J., Witbrock, M.J.: Learning to tell two spirals apart. In: Proceedings of the 1988 Connectionist Models Summer School San Mateo, pp. 52–59. Morgan Kauffman, San Francisco (1988)

Personalised Modelling for Multiple Time-Series Data Prediction: A Preliminary Investigation in Asia Pacific Stock Market Indexes Movement

Harya Widiputra, Russel Pears, and Nikola Kasabov

Knowledge Engineering and Discovery Research Institute,
Auckland University of Technology, Auckland, New Zealand
`{wpj6371, russel.pearls, nkasabov}@aut.ac.nz`

Abstract. The behaviour of multiple stock markets can be described within the framework of complex dynamic systems (CDS). Using a global model with the Kalman Filter we are able to extract the dynamic interaction network (DIN) of these markets. The model was shown to successfully capture interactions between stock markets in the long term. In this study we investigate the effectiveness of two different personalised modelling approaches to multiple stock market prediction. Preliminary results from this study show that the personalised modelling approach when applied to the rate of change of the stock market index is better able to capture recurring trends that tend to occur with stock market data.

1 Introduction

Prediction of multiple time-series data is a challenging task. In this paper we study the interactions that occur within several stock markets within the Asia Pacific region. Previous research has shown that a given stock market index is influenced by movement in the indices of certain other markets (Antoniou 2003), (Collins 2003). With the use of techniques such as multiple linear regressions a global model can be built that captures long-term trends to some degree of accuracy. However, such global models are not suitable for short-term predictions, particularly in the case of stock market index data. This task serves as a motivation for our work, and we apply methods adopted in the bioinformatics domain to this problem. Some recent studies in the bioinformatics domain used a personalised modelling approach to predict the behaviour of a patient by looking for similar conditions (by comparing features of the patient) from past patient data (Chan 2006). In this study we associate the patient condition with a current stock market index value and then search for similar conditions based on selected features from historical stock market index data. The similar patterns found from the past are used to predict the movement of multiple stock market indexes.

In our previous work we revealed interactions between stock markets in Asia Pacific which captured the global trend of interdependence between stock markets on a long-term basis. In this study we apply a personalised model to multiple time-series

financial data collected on a daily basis (shorter period). We analysed the performance of personalised modelling by using raw values of stock market indexes in Asia Pacific as well the velocity of change between stock market over a period of time.

The study is focused on markets in the Asia Pacific region and includes the stock indexes of Hong Kong, New Zealand, Australia, Japan, Indonesia, Malaysia, Singapore, Korea, Taiwan, and China. In section 2, we briefly describe global model and personalised models that can be used for multiple time-series prediction. Section 3 presents the results from the application of both global and personalised models. Finally, conclusions and directions for future research are outlined in section 4.

2 Global Model and Personalised Models

A Global model is an implementation of inductive reasoning approach, where the approach is concerned with the creation of a model from all available data, representing the entire problem space. Such models are effective in capturing global trends which can be used to provide generalized solutions over the whole problem space. Transductive inference, introduced by Vapnik (1998) and it used by Kasabov (2007a) is defined in contrast as a method used to estimate the value of a potential model only for a single point of space by utilizing additional information related to that point. This type of inference is suitable for building personalised models.

While the inductive approach is useful when a global solution of the problem is needed in an approximate form, the transductive approach is more appropriate for applications where the focus is not on the model, but rather on a specific case or vector. This is very relevant to the movement of stock market indexes value where the changes of values tend to be specific from time to time. Using a transductive approach fits the common sense principle which states that to solve a given problem one should avoid solving a more general problem as an intermediate step.

2.1 K-Nearest Neighbour (K-NN) and Weighted K-Nearest Neighbour as Transductive Reasoning

A personalised model based on transductive reasoning, is created on the fly for every new input vector and this individual model is based on the closest data samples to the new sample taken from a data set. One of the simplest and most widely used approaches to personalised modelling is the *K*-NN (*K*-nearest neighbour) method, where for every new sample the nearest *K* samples are derived from a data set using a distance measure to define similarity, which is usually based on the Euclidean distance. In the *K*-NN method, the output value y_i for a new vector x_i is calculated as the average of the output values of *k* nearest samples from data set D_i .

Another method that can be used to increase the performance is known as the weighted *K*-NN (WKNN), where the output y_i is calculated based not only on the output values (e.g. class label) y_j of the *K*-NN samples, but also on a weight w_j , that depends on the distance of them to x_i

$$y_i = \frac{\sum_{j=1}^{N_i} w_j y_j}{\sum_{j=1}^{N_i} w_j}, \quad (1)$$

where y_j is the output value for sample x_j from D_i and w_j are their weights calculated based on the distance from the new input vector

$$w_j = [\max(d) - (d_j - \min(d))] / \max(d). \quad (2)$$

The vector $\mathbf{d} = [d_1, d_2, \dots, d_{N_i}]$ is defined as the distances between the new input vector x_i and the N_i nearest neighbours (x_j, y_j) for $j = 1$ to N_i

$$d_j = \sqrt{\sum_{l=1}^V (x_{i,l} - x_{j,l})^2}, \quad (3)$$

where V is the number of the input variables defining the dimensionality of the problem space; $x_{i,l}$ and $x_{j,l}$ are the values of variable x_l in vectors x_i and x_j , respectively. The parameters $\max(\mathbf{d})$ and $\min(\mathbf{d})$ are the maximum and minimum values in \mathbf{d} respectively. The weights w_j have the values between $\min(\mathbf{d})/\max(\mathbf{d})$ and 1; the sample with the minimum distance to the new input vector has the weight value of 1, and it has the value of $\min(\mathbf{d})/\max(\mathbf{d})$ in case of maximum distance.

3 Experiments and Results

Our dataset consist of daily data of 720 days from June 2005 to June 2007. We normalize the data to lie in the range [-1,1], and we further pre-process the data by removing the linear trend. Figure 1 illustrates the trajectories produced from the dataset.

3.1 Transductive Reasoning with K-NN and WKNN

The first implementation of personalised modelling that we use is the basic K-Nearest Neighbour (K-NN) to daily data. We apply K-NN to data which has been normalized and de-trended so that the values lie between -1 and 1. Using K-NN we try to predict the value of all stock market indexes at time step $t+1$ given the values from previous time steps 1, 2, ..., t. Results of the application of K-NN are shown in Figure 2.

As we can see from the graphic and calculated RMSE (0.079), K-NN actually shows good predictive performance. However, a deeper analysis reveals that the nearest neighbours found are actually points which are close to current vector in terms of time. Based on this we can assume that the basic K-NN tends to be highly localised in its pattern extraction and does not take into consideration relevant patterns in the more distant past. Past trends are also important when deviations in behaviour occur that cannot be explained by recent events alone. This means that we cannot place absolute confidence in the basic form of the K-NN in this domain of stock market prediction.

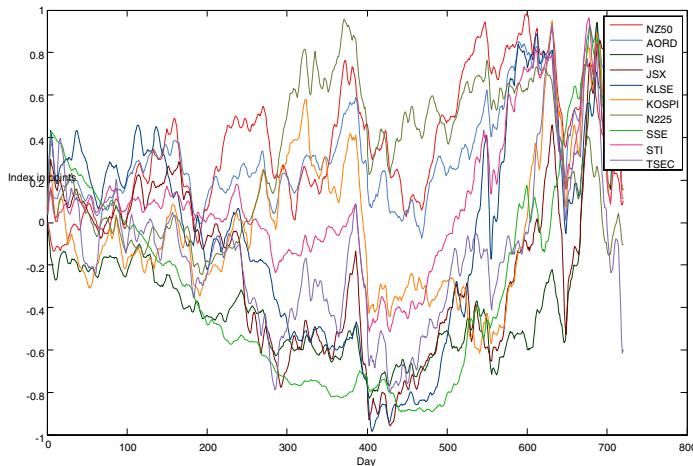


Fig. 1. Normalized and trend-removed daily stock market trajectories

This motivated us to use the rate of change in the indeed rather, then the value of the index itself. The rate of change has more potential to extract patterns that are outside the immediate time locality. Suppose that we have a state S_i in the system which matches most closely to states S_j , S_k and S_l from the immediate past. However if the rate of change of the current state S_i is very different from S_j , S_k and S_l then the next state from S_i will likely be quite different from each of S_i , S_j or S_k . Having obtained a prediction of the rate of change (hereinafter referred to as velocity) the actual stock index value can be obtained by a combination of the previous value (which is known at time $t-1$) and the predicted velocity at time t .

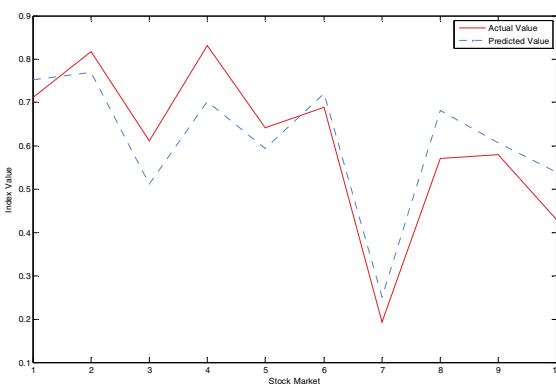
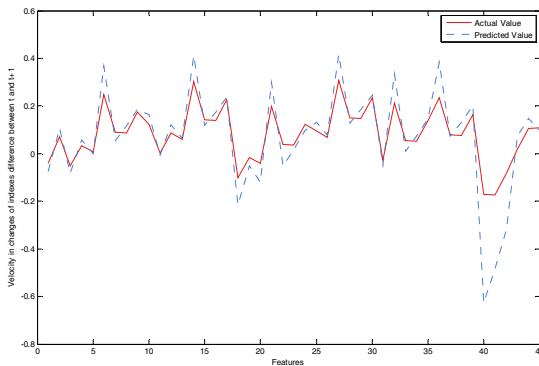


Fig. 2. Prediction of $t+1$ values of 10 stock market indexes with K-NN; t represents time point 701; $K=10$

Table 1. 10 nearest neighbours with the KNN approach based on normalized index value

Time Point	700	699	698	693	697	694	696	663	660	662
Distance	0.128	0.307	0.320	0.373	0.378	0.378	0.514	0.559	0.566	0.573

We use 45 features which consist of differences of index values between pairs of stock markets. The results of the velocity prediction process are shown in Figure 3. We can see from the results that the K -NN (RMSE=0.1057) performs quite well in predicting the velocity, and we could see that nearest neighbours found are not limited to those points that are close to the current point. Some of the nearest neighbours are points from older data points. Based on this we can assume that patterns of stock movement tend to recur over a period of time. However, it should be noted that the RMSE value is much higher than prediction with the raw index value. This motivated us to investigate the utility of the WKNN approach. We hypothesise that in stock market scenario some neighbours have much more importance than others in the prediction process and the WKNN approach is well suited to weigh the relative importance of each neighbour over the others.

**Fig. 3.** Predictions based on velocity and KNN**Table 2.** 10 nearest neighbours based on the basic KNN approach applied on velocity

Time Point	93	654	398	697	700	658	696	291	692	690
Distance	0.128	0.307	0.320	0.373	0.378	0.378	0.514	0.559	0.566	0.573

Note: Feature 1 is the velocity of changes in index difference between NZ50 and AORD, feature 2 is the velocity of changes in index difference between NZ50 and HSX, and so on and so forth.

Results of the WKNN approach can be seen in Figure 4. We can see from Figure 4 and from the RMSE (RMSE=0.0367) value that WKNN performs much better when compared to the basic K -NN approach, thus supporting our hypothesis. Thus we can

see that personalised modelling can be used not only for prediction of multiple time-series financial data but also could to reveal new knowledge, such as information that interactions between stock markets viewed from the velocity perspective tend to recur over time.

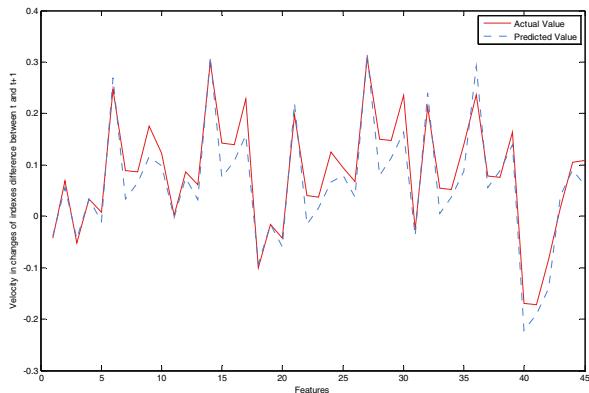


Fig. 4. Predictions based on velocity and WKNN

Table 3. 10 nearest neighbours based on the WKNN approach

Time Point	93	654	398	697	700	658	696	291	692	690
Distance	0.240	0.244	0.266	0.311	0.327	0.354	0.377	0.423	0.428	0.438

4 Conclusion and Further Research

Preliminary results from this study using K -NN and WKNN show that personalised models predicts stock index values in the shorter term with a reasonable degree of accuracy.

As future work we would like to investigate the effectiveness of the WWKNN approach (Kasabov 2007a) since we believe that interactions between different stock markets have differing degrees of influence, so that more weight should be given to those markets that have a higher degree of interaction with each other. The challenge here is to identify a method that will rank the degree of influence between pairs from the stock markets. We would also like to extend the model to be able to capture global, local and personal trends into one single cohesive model for financial time-series data, and therefore improve prediction. The global model will be used to capture the trend in the whole problem space, the local model will extract local regression or specific behaviour from different sub-spaces of the whole problem space, and finally the personalised model will be used to cope with changes with current data or state of the system. Integrating the three models will provide a better framework for pattern recognition, adaptation and prediction for financial time-series data.

Acknowledgements

We would like to thank Dr Antoaneta Sergueeva from Centre for Empirical Finance, Brunel University, and Brunel Business School for her contributions in our previous work of building a global model to extract dynamic interaction between stock markets in Asia Pacific.

References

1. Antoniou, A., Pescetto, G., Violaris, A.: Modelling international price relationships and interdependencies between the stock index and stock index future markets of three EU countries: A Multivariate Analysis. *Journal of Business Finance and Accounting* 30, 645–667 (2003)
2. Caporale, G.M., Sergueeva, A., Wu, H.: A mixed-game agent-based model for simulating financial contagion. In: *Proceedings of the 2008 Congress on Evolutionary Computation*, pp. 3420–3425. IEEE Press, Los Alamitos (2008)
3. Chan, Z., Kasabov, N., Collins, L.: A two-stage methodology for gene regulatory network extraction from time-course gene expression data. *Expert System with Applications* 30, 59–63 (2006)
4. Chiang, T.C., Doong, S.: Empirical analysis of stock returns and volatility: Evidence from seven Asian stock markets based on TAR-GARCH model. *Review of Quantitative Finance and Accounting* 17, 301–318 (2001)
5. Collins, D., Biekpe, N.: Contagion and interdependence in African stock markets. *The South African Journal of Economics* 71(1), 181–194 (2003)
6. D'haeseleer, P., Liang, S., Somogyi, R.: Gene expression data analysis and modelling. In: *Proceedings of the Pacific Symposium on Biocomputing, Hawaii* (1999)
7. Goldfeld, S., Quandt, R.: A Markov model for switching regressions. *Journal of Econometrics* 1(1), 3–16 (1973)
8. Kasabov, N.: Evolving connectionist systems: Methods and applications in bioinformatics, Brain Study and Intelligent Machines. Springer, Heidelberg (2002)
9. Kasabov, N., Chan, Z., Jain, V., Sidorov, I., Dimitrov, D.: Gene regulatory network discovery from time-series gene expression data – A computational intelligence approach. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) *ICONIP 2004. LNCS*, vol. 3316, pp. 1344–1353. Springer, Heidelberg (2004)
10. Kasabov, N.: Global, local and personalised modelling and pattern discovery in bioinformatics: An integrated approach. *Pattern Recognition Letters* 28, 673–685 (2007a)
11. Kasabov, N.: *Evolving Connectionist Systems: The Knowledge Engineering Approach*. Springer, Heidelberg (2007b)
12. Masih, A., Masih, R.: Dynamic modelling of stock market interdependencies: An empirical investigation of Australia and the Asian NICs, *Working Papers*, 98–18, 1323–9244, University of Western Australia (1998)
13. Welch, G., Bishop, G.: An Introduction to the Kalman Filter, *Computer Science Working Papers TR95-041*, University of North Carolina at Chapel Hill (2006)
14. Sergueeva, A., Kalganova, T., Khan, T.: An intelligent system for risk classification of stock investment projects. *Journal of Applied Systems Studies* 4(2), 236–261 (2003)
15. Sergueeva, A., Khan, T.: Knowledge representation in risk analysis. *Business and Management Working Papers*. Brunel University, pp. 1–21 (March 2004)

16. Sergueeva, A., Wu, H.: Computational intelligence in financial contagion analysis. In: Seventh International Conference on Complex Systems, Boston, Massachusetts (2007); Inter-Journal on Complex Systems 2229, 1–12 (2008)
17. Song, Q., Kasabov, N.: Dynamic evolving neuro-fuzzy inference system (DENFIS): Online learning and application for time-series prediction. IEEE Transactions of Fuzzy Systems 10, 144–154 (2002)
18. Vapnik, V.N.: Statistical Learning Theory. Wiley Inter-Science, Chichester (1998)

Dynamic Neural Fuzzy Inference System

Yuan-Chun Hwang and Qun Song

Auckland University of Technology, Knowledge Engineering and Discovery Research
Institute, Auckland, New Zealand
`{peter.hwang, qun.song}@aut.ac.nz`

Abstract. This paper proposes an extension to the original offline version of DENFIS. The new algorithm, DyNFIS, replaces original triangular membership function with Gaussian membership function and use back-propagation to further optimizes the model. Fuzzy rules are created for each clustering centre based on the clustering outcome of evolving clustering method. For each test data, the output of DyNFIS is calculated through fuzzy inference system based on m -most activated fuzzy rules and these rules are updated based on back-propagation to minimize the error. DyNFIS shows improvement on multiple benchmark data and satisfactory result in NN3 forecast competition.

1 Introduction

Dynamic Evolving Neural Fuzzy Inference System, DENFIS[1], is an online local model was proposed by Qun et al. in 2002. The algorithm creates a Takagi-Sugeno (TS) fuzzy inference system (FIS)[2, 3] where the fuzzy rules are created based on the clustering outcome of the input vectors. The system evolves by updating its clusters and rules as new input vectors enter the system. The concept of developing a fuzzy inference system based on data clustering is one of the reasons why DENFIS performs well as an online algorithm since this method captures the different sub-groups of data points that represent similar characteristics and develop a rule for each sub-group.

Online algorithms [4, 5] process the input data piece by piece and usually aim for fast processing speed and minimal memory usage since it is designed to process streams of data. Because of this, the complexity of the algorithm is often minimised the input data discarded once it's processed. Offline algorithm[6-8] on the other hand has the benefit of being able to process the entire dataset as a whole. This often allows for better prediction accuracy since there are more data for the algorithm to learn from. DENFIS was originally designed as an online algorithm and because of this; the FIS and learning method were simplified to meet the requirements.

DENFIS offline was proposed together with the online version of DENFIS, which sacrifices the dynamic evolving aspect of the DENFIS algorithm and replaces it with more sophisticated learning algorithm aims to provide higher accuracy. It has shown improvement in prediction accuracy; however, more optimisation can be applied to further improve its accuracy.

In the original DENFIS algorithm, the antecedents are created based on the cluster centres and the consequences are created based on the samples near the cluster

centres. No further supervised learning is applied once the fuzzy rules are created until more input vectors enter the system. Triangular membership function was used to reduce the computational complexity.

This paper proposed an improved offline version of DENFIS that extends the original algorithm with more in-depth learning. In the proposed algorithm, DyNFIS, Gaussian MF is used in place of the triangular MF. Since Gaussian MF extends infinitely, it provides better cover of the problem space, the gaps that may be introduced by triangular MF due to its finite range is eliminated. DyNFIS also allows the antecedents and consequences to be optimised using back-propagation to minimise the error in active rules. With the above improvements, DyNFIS is expected to be more accurate due to back-propagation and generalised due to its Gaussian MF and therefore more suitable for real world applications that are not time critical.

2 Algorithm Description

The DyNFIS offline learning process is outlined as follows:

1. Cluster the data to find n cluster centres using Evolving Clustering Method (ECM[1])
2. Create one fuzzy rule for each cluster .
3. Antecedent is created based on each cluster's centre position.
4. Consequence is calculated by deriving a linear function with the samples that belong to the clusters.
5. For each testing sample, derive the output from m rules close it and adjust the fuzzy membership function and its consequence using back-propagation to minimise the error.
6. Repeat step three for multiple epochs or until desire accuracy reached.

Consider the data is composed of N data pairs with P input variables and one output variable $\{(x_{i1}, x_{i2}, \dots, x_{ij}), y_i\}$, $i = \{1, 2, \dots, N\}$, $j = \{1, 2, \dots, P\}$ M fuzzy rules are defined initially through the clustering procedure, the l th rule has the form of:

R_l: If x_1 is about F_{l1} and x_2 is about $F_{l2} \dots x_p$ is about F_{lp} and then

$$n_l = \beta + x_1\beta_1 + x_2\beta_2 + \dots + x_p\beta_p \quad (1)$$

F_{lj} are the fuzzy sets defined by the following Gaussian type membership function (MF):

$$\text{Gaussian MF} = \alpha \exp\left[-\frac{(x - m)^2}{2\sigma^2}\right] \quad (2)$$

Using the modified centre average defuzzification procedure, the output value of the system can be calculated for an input vector $x_i = [x_1, x_2 \dots x_p]$ as follows:

$$f(x_i) = \frac{\sum_{l=1}^M n_l \prod_{j=1}^P \alpha_{lj} \exp\left(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2}\right)}{\sum_{l=1}^M \prod_{j=1}^P \alpha_{lj} \exp\left(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2}\right)} \quad (3)$$

Suppose the DyNFIS is given a training input-output data pair $[x_i, t_i]$. The system minimizes the following objective function:

$$E = \frac{1}{2} [f(x_i) - t_i]^2 \quad (4)$$

The back-propagation (steepest descent) algorithm is used to obtain the formulas (5)-(11) for the optimisation of the parameters n_l , α_{lj} , m_{lj} , σ_{lj} and β_l

$$\begin{aligned} m_{lj}(k+1) &= m_{lj}(k) - \frac{\eta_m \Phi(x_i)[f^{(k)}(x_i) - t_i]}{\sigma_{lj}^2(k)} \\ &\times [n_l(k) - f^{(k)}(x_i)][x_{ij} - m_{lj}(k)] \end{aligned} \quad (5)$$

$$n_l(k+1) = n_l(k) - \eta_n \phi(x_i)[f^{(k)}(x_i) - t_i] \quad (6)$$

$$\begin{aligned} \alpha_{lj}(k+1) &= \alpha_{lj}(k) - \frac{\eta_\alpha \phi(x_i)[f^{(k)}(x_i) - t_i]}{\alpha_{lj}(k)} \\ &\times [n_l(k) - f^{(k)}(x_i)] \end{aligned} \quad (7)$$

$$\begin{aligned} \sigma_{li}(k+1) &= \sigma(k) - \frac{\eta_\sigma \phi(x_i)[f^{(k)}(x_i) - t_i]}{\sigma_{lj}^2} \\ &\times [n_l(k) - f^{(k)}(x_i)][x_{ij} - m_{lj}(k)]^2 \end{aligned} \quad (8)$$

$$\phi(x_i) = \frac{\prod_{j=1}^p \alpha_{lj} \exp(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2})}{\sum_{l=1}^M \prod_{j=1}^p \alpha_{lj} \exp(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2})} \quad (9)$$

$$\beta_{l0}(k+1) = \beta_l(k) - \eta_\beta \phi(x_i)[f^{(k)}(x_i) - t_i] \quad (10)$$

$$\beta_{lj}(k+1) = \beta_l(k) - \eta_\beta \phi(x_i)[f^{(k)}(x_i) - t_i] x_{ij} \quad (11)$$

Where η_m , η_n , η_α , η_σ and η_β are learning rate for updating the parameters: n_l , α_{lj} , m_{lj} , σ_{lj} and β_l respectively.

In DyNFIS algorithm, the following indexes are used:

- training data points: $i=1,2,\dots,N$;
- input variables: $j=1,2,\dots,P$;
- fuzzy rules: $l=1,2,\dots,M$;
- training iterations: $k=1,2,\dots$

3 Case Study – Benchmark

We have applied DyNFIS on Mackey-Glass dataset, which has been widely used as a benchmark problem in the area of neural networks, fuzzy systems and hybrid systems for time series prediction. The dataset was created with the following differential equation:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \quad (12)$$

The integer time points for the above equation were obtained using the *fourth-order Runge-Kutta* method. Here we assume that the time step is 0.1, $x(0) = 1.2$; $\tau = 17$; and $x(t) = 0$ for $t < 0$. [1]

The goal is to predict the values $x(t+6)$ from input vectors $[x(t-18), x(t-12), x(t-6), x(t)]$ for any value of the time t .

The following experiment was conducted with 1000 data points, from $t = 118$ to 1117. The first 500 data points were taken as the training data and the remaining 500 as the testing data. A model is created using the training data and then tested on the test data. The test results are shown in table 1 along with other published results for comparison.

Table 1. Prediction accuracy comparison of several offline algorithms on t+6 Mackay Glass dataset [1]

Methods	Neurons or Rules	Epochs	Training NDEI	Testing NDEI
MLP-BP	60	50	0.083	0.090
MLP-BP	60	500	0.021	0.022
ANFIS	81	50	0.032	0.033
ANFIS	81	200	0.028	0.029
DENFIS I (TS)	883	2	0.023	0.019
DENFIS II (MLP)	58	100	0.017	0.016
DyNFIS (TS)	55	100	0.017	0.016

As shown in table 1, DyNFIS performs better than DENFIS I offline and is on par with DENFIS II offline (MLP). The improvements are expected due to additional learning and the use of Gaussian membership function.

The performance similarity between DyNFIS and DENFIS II (MLP) is likely to be caused by the additional learning performed at different level. DENFIS II (MLP)'s prediction is optimized by improving the consequence of each rules without optimizing the membership functions and DyNFIS optimizes the fuzzy membership function instead and does less optimization of each rule's accuracy.

The prediction error on the T+6 Mackay Glass dataset is very low across top performing algorithms and has become difficult to show significant differences. A more difficult benchmark test on the performance of the DyNFIS algorithm is carried out on the T+85 Mackay Glass dataset which is a much more difficult.

For predicting the T+85 output value, the 3000 data points, from $t = 201$ to 3200, were extracted as the training data and 500 data points, from $t = 5001$ to 5500, were extracted as the testing data.

Table 2. Prediction accuracy comparison of several offline algorithms on t+85 Mackay Glass dataset [1]

Methods	Neurons or Rules	Epochs	Training NDEI	Testing NDEI
MLP-BP	60	50	0.083	0.090
ANFIS	81	50	0.032	0.033
ANFIS	81	500	0.024	0.025
DENFIS I (TS)	116	2	0.068	0.068
DENFIS I (TS)	883	2	0.023	0.019
DENFIS II (MLP)	58	60	0.020	0.020
DyNFIS (TS)	91	500	0.017	0.018

As shown in table 2, DyNFIS has no difficulty in solving the more difficult problem, T+85 Mackay Glass dataset.

4 Case Study – Real World

The proposed algorithm was entered in the NN3 Neural Network Forecasting competition[9] under the original name “DENFIS” to forecast 11 time series. It achieved 10th place using one set of parameters for all time-series problem. Better performance can be expected should the parameters be optimised for each time-series’ training data.

5 Conclusion and Direction for Further Research

This paper presents an improved version of DENFIS offline algorithm, DyNFIS, which enhances its learning by adapting additional learning to the fuzzy inference system as well as using Gaussian membership function. The algorithm uses M highly activated fuzzy rules to dynamically compose an inference system for calculating the output for a given input vector. The proposed system demonstrates superiority when compared with global models including MLP, ANFIS and the original DENFIS offline version on benchmark data. In the real life problems, it also demonstrated high performance in the NN3 competition with minimal parameter tuning.

DyNFIS builds an initial fuzzy inference system based on the result of the clustering process. These fuzzy memberships are then optimised using back-propagation to minimise the error in each rule.

In the recall process, DENFIS would give satisfactory results if the recall examples are near the cluster centres.

Future research directions include: automated parameter optimisation, alternate fuzzy rule types and application of DyNFIS model for classification problems.

References

1. Kasabov, N.K., Qun, S.: DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems* 10(2), 144–154 (2002)
2. Wang, L.-X.: *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, p. 352. Prentice-Hall, Englewood Cliffs (1994)
3. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* 15(1), 116–132 (1985)
4. Kasabov, N.: Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning. *IEEE Trans. SMC - part B Cybernetics* 31(6), 902–918 (2001)
5. Domeniconi, C., Gunopulos, D.: Incremental support vector machine construction. In: *ICDM 2001, Proceedings IEEE International Conference on Data Mining* (2001)
6. Jang, J.-S.R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics* 23, 665–684 (1993)
7. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural Networks* 2(5), 359–366 (1989)
8. Vapnik, V.N.: *Statistical Learning Theory*. John Wiley and Sons, New York (1998)
9. Crone, S.F.: NN3 Neural Network Forecasting competition (2006) (18/10/2006), <http://www.neural-forecasting-competition.com/NN3/results.htm> (cited 06/11/2008)

Author Index

- Abe, Shigeo I-937
Adachi, Masaharu I-14
Adam, Stavros II-308
Ahmad, Zainal II-469
Al-Ani, Ahmed I-103
Al-Jumaily, Adel I-103
Álvarez, I. II-402, II-410, II-418
Amemiya, Yoshihito II-851
Ando, Ruo I-555
Ang, Zhi Ping I-428
Aoki, Kazuhiro I-469
Arakawa, Masao I-995
Asahina, Akiyo I-361
Asai, Tetsuya II-851
Asthana, Akshay II-655
- Babinec, Štefan I-200, II-284
Ban, Tao I-563, I-571
Bando, Takashi I-679
Bandyopadhyay, Sanghamitra I-605, II-543
Barambones, Oscar I-1053
Barczak, Andre I-1095
Barczak, Andre L.C. I-945
Barreto, Guilherme A. I-1180
Barros, Allan Kardec I-169
Basford, Kaye I-529
Beňušková, Ľubica I-111, I-285, I-671
Becerra, Jose Antonio I-1013
Beigy, Hamid I-829
Bellas, Francisco I-1013, I-1037
Benso, Victor Alberto Parcianello II-859
Berenz, Vincent II-244
Bhattacharyya, Malay I-605
Bian, Peng I-251
Blas, Nuria Gómez II-123
Böcker, Sebastian I-513
Boel, René II-615
Boroomand, Arefeh I-883
Brito, D.S. I-169
Brown, Andrew II-1049
Brusic, Vladimir I-529
Buteneers, Pieter I-56
- Caamaño, Pilar I-1013
Cacciola, Matteo II-720
Carvalho, André C.P.L.F. de I-461, I-486, II-252
Casey, Matthew I-40
Čerňanský, Michal I-285, I-671
Chan, Ching Hau II-808
Chan, Hsiao-Lung I-194
Chan, Jonathan Hoyin II-260
Chandra, B. II-27, II-37, II-867
Chandra Sekhar, C. II-394
Chao, Pei-Kuang I-194
Chau, Sheau Wei II-808
Chaves, Adriana da Costa F. II-386
Chen, Jialiang I-436
Chen, Peng-Chuan I-194
Cheng, Wei-Chen II-3, II-378
Cho, Sung-Bae I-1121
Choi, Byung-Uk I-747
Choi, Sun-Wook II-1072
Chong, Tow Chong I-428
Chung, Jen-Feng II-1038
Chung, Wan Kyun I-64
Cichocki, Andrzej I-177, I-224, I-318, I-400, I-453, I-469, II-979, II-1021
Coelho, André L.V. I-1180, II-276
Conilione, Paul II-792
Corrêa, Marcelo F. II-461
Costea, Ruxandra L. II-885
Crutcher, Keith A. II-70
Cysneiros, Francisco José A. II-19
- d'Anjou, Alicia I-1045
Dadgostar, Farhad I-587
Dauwels, Justin I-177, I-224, I-318, I-400, I-469
Defoin-Platel, Michaël I-1229
Dehuri, Satchidananda I-1121
Delgado, Soledad II-437
de Lope, Javier I-1029
de Mingo, Luis Fernando II-893
Dhir, Chandra Shekhar I-613, II-80
Díaz, Miguel Angel II-123
Doi, Kunio I-504

- Domingues, Marco A.O. II-19
 Doya, Kenji I-22
 Duarte, João I-723, II-97
 Duch, Włodzisław II-70, II-88
 Duro, Richard J. I-1013, I-1037
- Echegoyen, Zelmar I-1021, I-1045
 Eggers, G. II-575
 Eggert, Julian I-275, II-567, II-960
 Eto, Masashi I-579
- Farkaš, Igor II-292
 Fernández, Luis II-437
 Fiasché, Maurizio II-720
 Figueiredo, Karla II-461
 Franz, Liz I-111
 Fujii, Masami I-369
 Fujimura, Hirotada II-835
 Fukaya, Naoki I-679
 Fukushima, Yasuhiro I-72, I-423
 Funase, Arao II-1021
 Funaya, Hiroyuki I-731, I-929
 Furber, Steve II-1049, II-1057
 Furuhashi, Takeshi II-1013
 Fyfe, Colin II-935
- Gabrys, Bogdan I-1172, II-607
 Gao, Tao II-695, II-736
 Gao, Wen-chun II-695, II-736
 Garcez, Artur d' Avila II-335
 Garro, Beatriz A. II-800
 Gedeon, Tom II-236, II-655
 Gerdelan, Antony P. I-1079
 Geweniger, Tina II-61
 Goecke, Roland II-655
 Gonçalves, Nicolau II-559
 Gordon, Sean W. I-962
 Górriz, J.M. II-402, II-410, II-418
 Grábner, Martin I-893
 Graña, Manuel I-1021, I-1045, I-1053
 Gross, Horst-Michael I-805,
 I-813, II-567
 Gu, Xiaodong II-535, II-768
 Guan, Cuntai I-1137
 Guizilini, Vitor Campanholo I-1110
 Guo, Shanqing I-563, I-571
 Gutiérrez, Abraham II-437, II-893
- Hada, Takahiro II-203
 Hagiwara, Katsuyuki II-187
- Hammer, Barbara II-61
 Hara, Kazuyuki II-195
 Harada, Hiroshi I-851
 Harris, Chris I-408
 Harun, Mohd Hafez Hilmi II-445
- Hasegawa, Mikio I-851
 Hasegawa, Osamu I-32
 Hasegawa, Tomonari I-837
 Hasenfuss, Alexander II-61
 Hashimoto, Genta II-583
 Hashimoto, Sho I-145
 Hatakeyama, Takashi I-352
 Hatori, Yasuhiro I-301
 Hattori, Motonobu I-344
 Hayakawa, Yoshihiro I-875
 Hayashi, Akira I-663, I-715
 Hayashi, Hatsuo I-161
 Hayashida, Kenji I-970
 Hazeyama, Hiroaki I-539, I-547
 Hellbach, Sven II-567
 Herbarth, Olf II-816
 Hikawa, Hiroomi II-835
 Hirabayashi, Miki I-267
 Hirai, Yuzo II-353
 Hirayama, Jun-ichiro I-361, II-951
 Hiro, Takafumi II-583
 Hisada, Masayuki I-1163
 Hitomi, Kentarou I-679
 Hlalele, Nthabiseng II-485
 Ho, Kevin II-324, II-919
 Hon, Hock Woon II-808
 Honda, Yutaro I-32
 Honma, Shun'ichi I-621
 Hori, Michihiro II-843
 Hoshino, Yuuki I-621
 Hotta, Kazuhiro II-344, II-361, II-671
 Hu, Yingjie I-1221
 Hudson, Nigel II-979
 Hung, Chihli II-139
 Hwang, Yuan-Chun I-1245
- Ichiyama, Shohei II-583
 Ide, Yoshinori I-416
 Igarashi, Jun II-1065
 Ikeda, Kazushi I-679, I-731, I-929
 Inada, Tomohisa I-494
 Inohira, Eiichi I-240
 Inoue, Daisuke I-579
 Ishii, Kazuo I-771, I-779
 Ishii, Shin I-361, I-469, II-951

- Ishikawa, Masumi I-763
 Ishikawa, Satoru I-293
 Ishikawa, Seiji I-504
 Ishizaki, Shun I-344
 Iswandy, Kuncup II-477
 Itai, Yoshinori I-504
 Ito, Masahiro I-986
 Ito, Norihiko II-663
 Itoh, Hideaki I-327
 Iwashashi, Naoto I-953
 Iwasa, Kaname II-300
 Iwata, Akira II-300, II-859
 Iwata, Kazunori I-663
- Jarur, Mary I-1211
 Jeong, Jaeseung II-979
 Jeong, Seungdo I-747
 Jin, Xin II-1057
 Jin, Yaochu I-48, I-216
 Jones, Ben I-216
 Jung, Tzzy-Ping II-1038
 Juszczyszyn, Krzysztof II-607
- Kadlec, Petr I-1172
 Kadobayashi, Youki I-539, I-547
 Kaji, Daisuke II-903
 Kalra, P.K. II-27, II-37, II-867
 Kamimura, Ryotaro II-163, II-171,
 II-268, II-943
 Kamioka, Takumi I-22
 Kanda, Atsushi I-771
 Kane, G.J. II-575
 Kaneko, Shuzo II-776
 Karras, Dimitrios Alexios I-595, I-1063,
 II-112, II-308
 Kasabov, Nikola I-3, I-962, I-1163,
 I-1196, I-1204, I-1221, I-1229, I-1237
 Kaseda, Chosei I-995
 Kashimori, Yoshiki I-444
 Kaski, Samuel I-629
 Kato, Shigeru I-638
 Kato, Shuzo I-851
 Kato, Takatoshi I-352, I-621
 Katsumata, Shiori I-88
 Katsuragawa, Shigehiko I-504
 Kawachi, Ryo I-970
 Kawai, Yuji I-844
 Kawano, Tomohiko I-755
 Kawato, Mitsuo I-336
 Kawawaki, Dai I-336
- Kawewong, Aram I-32
 Kazemitarab, Seyed Jalal I-829
 Kazienko, Przemysław II-607
 Khan, Mukaram II-1057
 Khoo, Suisin II-236
 Khushaba, Rami N. I-103
 Kie, Patrick Low Tiong II-509, II-927
 Kikuchi, Shinichi I-344
 Kim, Hyoungseop I-504
 King, Irwin II-53
 Kirstein, Stephan I-805, I-813
 Kitajima, Tatsuo I-310
 Kitano, Katsunori I-978
 Klette, Reinhard I-1002, II-527
 Ko, Li-Wei II-1038
 Kobayakawa, Shunsuke II-679
 Kobayashi, Kunikazu II-147
 Koga, Takanori II-583, II-1065
 Kojima, Hiroshi I-88
 Komazawa, Daisuke I-731
 König, Andreas II-477, II-827
 Kordos, Miroslaw II-453
 Körner, Edgar I-805, I-813, II-567
 Koshimizu, Hiroshi I-1146
 Kuba, Hinata II-361
 Kubota, Shigeru I-310
 Kugler, Mauricio II-300, II-859
 Kuiper, Martin II-615
 Kuremoto, Takashi II-147
 Kuroda, Shigeru I-72, I-423
 Kuroda, Taro I-851
 Kurokawa, Hiroaki II-776
 Kuroyanagi, Susumu II-300, II-859
 Kvičera, Václav I-893
- Laaksonen, Jorma II-703
 Lai, Pei Ling II-935
 Lai, Weng Kin II-599
 Lan, Shaohua I-571
 Latchoumane, Charles II-979
 Lau, Phooi Yee II-745
 Lauwereyns, Johan I-416
 Lee, Chong Ho II-1072
 Lee, Hong II-639
 Lee, Nung Kion I-478
 Lee, Sanghoon I-64
 Lee, Sang Hyoung I-64
 Lee, Shih-Tseng I-194
 Lee, Soo Young I-613, II-80
 Lester, David II-1049

- Leung, Chi-sing II-324, II-919
 Leung, Chi Sing II-316
 Li, Fajie I-1002
 Li, Jianming I-428
 Li, Ming Hui I-428
 Li, Xi I-521
 Li, Yongming II-493
 Lian, Xiao-Chen II-647
 Liao, Ling-Zhi II-994
 Libralon, Giampaolo L. I-486
 Lim, Gi Hyun I-747
 Lim, Kian Guan I-428
 Lim, Mei Kuan II-599
 Lin, Chin-Teng II-1038
 Lin, Chunhao II-426
 Lin, Lanxin II-426
 Lin, Ming-An I-194
 Lin, Tzu-Chao II-687
 Lin, Yang-Cheng I-647
 Liou, Cheng-Yuan II-3, II-378
 Liu, Mu-kun II-687
 Liu, Qingshan II-1003
 Liu, Qingzhong I-723
 Liu, Zheng-guang II-695, II-736
 Liu, Zhifeng II-527
 Loo, Chu Kiong I-1103
 Lopez-Guede, Jose Manuel I-1053
 Lopez-Peña, Fernando I-1037
 López, Luis Fernando de Mingo II-123
 López, M. II-402, II-410, II-418
 Lorena, Ana C. I-486
 Lu, Bao-Liang II-647
 Lu, Shao-Wei II-1038
 Lucena, Fausto I-169
 Luo, Ming I-129
- Ma, Jun I-707, II-623
 Ma, Liying II-711
 Ma, Yu II-768
 Maia, José Everardo B. I-1180
 Makula, Matej I-671
 Mandic, Danilo P. I-453
 Maravall, Darío I-1029
 Marinov, Cornelius A. II-885
 Mariyama, Toshisada I-327
 Markošová, Mária I-111
 Marmulla, R. II-575
 Martínez, Víctor II-893
 Marwala, Tshilidzi II-485, II-517,
 II-728, II-752
- Masisi, Lesedi Melton II-517
 Matsuda, Michiyuki I-469
 Matsuda, Nobuo II-703
 Matsui, Hideyuki II-155
 Matsumoto, Kazuya I-1196
 Matsumoto, Masashi I-688
 Matsushima, Fumiya I-352
 Matsuyama, Yasuo I-352, I-621, II-369
 Matsuzaki, Masunori II-583
 Matykiewicz, Paweł II-70
 Maul, Tomás II-599
 Maurice, Monique I-224, I-400, II-979
 Mayr, Christian I-137
 Meechai, Asawin II-260
 Megali, Giuseppe II-720
 Mendis, B. Sumudu U. II-655
 Menhaj, Mohammad B. I-883
 Messom, Chris I-587, I-1095
 Mishra, Bijan Bihari I-1121
 Mistry, Jaisheel II-752
 Miyamoto, Daisuke I-539, I-547
 Miyamoto, Goh I-851
 Miyazawa, Yasuhiro I-384
 Miyoshi, Seiji II-195
 Mogi, Ken I-377
 Monk, Travis I-408
 Morabito, Francesco C. II-720
 Moriguchi, Yoshitaka II-344
 Morikawa, Kazuya I-937
 Morishige, Ken-ichi I-336
 Morita, Masahiko I-384
 Morita, Mikio I-763
 Morita, Satoru II-784
 Mouri, Motoaki II-1021
 Mudroch, Martin I-893
 Mulder, Wim De II-615
 Murata, Yoshitoshi I-851
 Murray, Alan I-95
 Musial, Katarzyna II-607
 Musirin, Ismail II-445
- Nagamatu, Masahiro I-787
 Nagata, Kenji I-696
 Nakajima, Hidekazu I-259
 Nakajima, Koji I-875
 Nakajima, Yukiko I-986
 Nakamura, Takeshi I-469
 Nakao, Koji I-579
 Nakayama, Hirotaka I-995
 Nascimento, Mariá C.V. I-461

- Nattkemper, Tim W. I-513, II-911
 Nelwamondo, Fulufhelo II-485, II-517
 Nelwamondo, Fulufhelo V. II-752
 Neoh, Tze Ming I-1103
 Neto, Manoel J.R. II-45
 Neves, João Carvalho das I-723
 Neves, Joao Carvalho II-97
 Ng, Ee Lee II-599
 Nikkilä, Janne II-559
 Nina, Zhou II-551
 Nishida, Youichi I-352
 Nishida, Yuya I-787
 Nishigami, Toru I-208
 Nishiguchi, Junya I-995
 Nishikawa, Hitoshi I-821
 Nishikawa, Ikuko I-986
 Nishino, Eisuke I-579
 Nishioka, Ryota I-962
 Nitta, Katsumi II-220
 Nomoto, Atsuo II-671
 Nomura, Yoshihiko I-929
 Nonaka, Yukihiro I-161
 Noor, Rabiatul Adawiah Mat II-469
 Nugaev, Ildar II-131
 Obayashi, Masanao II-147
 Ohashi, Hirodata I-267
 Ohkouchi, Kazuya I-579
 Ohnishi, Noboru I-169
 Okada, Masato I-655, II-195
 Okamoto Jr., Jun I-1110
 Oliveira, Pedro G. de II-276
 Omori, Takashi I-293
 Ong, Chuan Poh I-1103
 Onoda, Takashi II-663
 Osana, Yuko II-203, II-212
 Osman, Hassab Elgawi II-104, II-760
 Osman, Yasir Salih II-808
 Othman, Muhammad Murtadha II-445
 Ozawa, Seiichi I-821, I-1163, I-1196
 Pang, Shaoning I-962, I-1163, I-1196
 Partzsch, Johannes I-137
 Patel, Leena N. I-95
 Patel, Pretesh B. II-728
 Pavlou, Athanasios I-40
 Pears, Russel I-1237
 Pechač, Pavel I-893
 Peng, Jianhua I-129
 Pestian, John P. II-70
- Phon-Amnuaisuk, Somnuk I-153, I-232
 Pilichowski, Maciej II-88
 Plana, Luis II-1049
 Pontin, David R. I-909
 Pospíchal, Jiří I-200, II-284
 Prieto, Abraham I-1037
 Prom-on, Santitham II-260
 Prudêncio, Ricardo B.C. II-45
 Puntonet, C.G. II-402, II-410, II-418
 Quek, Chai I-1137
 Ralescu, Anca L. I-453
 Ramírez, J. II-402, II-410, II-418
 Ramanathan, Kiruthika I-428
 Rast, Alexander II-1057
 Raszkowsky, J. II-575
 Rebhan, Sven II-960
 Ren, Lu II-335
 Reyes, Napoleon H. I-1071, I-1079
 Ribeiro, Bernardete I-723, II-97
 Ribeiro, Marcelo N. II-45
 Richter, Matthias II-816
 Roeder, Stefan W. II-816
 Rosen, Alan I-794
 Rosen, David B. I-794
 Rossi, André L.D. II-252
 Roy, Asim I-821
 Rush, Elaine I-1204
 Rutkowski, Tomasz M. I-453
 Saeki, Katsutoshi II-877
 Saethang, Thammakorn II-260
 Saha, Sriparna II-543
 Saito, Toshimichi I-837, I-844, I-1146
 Sakai, Ko I-301
 Sakakibara, Kazutoshi I-986
 Sakamoto, Hirotaka I-986
 Sakamoto, Kazuhiro I-259
 Sakumura, Yuichi I-469
 Salas-Gonzalez, D. II-402, II-410, II-418
 Salazar, Diogo R.S. II-11
 Sallis, Philip I-917, I-1211
 Samsudin, Mohamad Faizal bin II-631
 Samura, Toshikazu I-344
 Sarrafzadeh, Abdolhossein I-587
 Sato, Hideaki II-703
 Sato, Masa-aki I-336
 Sato, Masanori I-771
 Sato, Masayoshi I-293
 Sato, Naoyuki I-186

- Sato, Yasuomi D. I-867
 Sawada, Koji I-352
 Schüffny, Rene I-137
 Scherbart, Alexandra I-513, II-911
 Schleif, Frank-Michael II-61
 Schliebs, Stefan I-1229, II-615
 Schramm, Lisa I-48
 Schrauwen, Benjamin I-56
 Segovia, F. II-402, II-410, II-418
 Sekine, Yoshifumi II-877
 Sekino, Masashi II-220
 Sendhoff, Bernhard I-48, I-216
 Shanmuganathan, Subana I-917
 Sharif, Waqas II-960
 Shen, Minfen I-436
 Shi, Luping I-428
 Shiba, Naoto I-494
 Shibata, Katsunari I-755, II-631, II-970
 Shibata, Tomohiro I-679, II-1029
 Shiblee, Md. II-27, II-37
 Shiino, Masatoshi I-867
 Shimizu, Atsushi II-212
 Shimizu, Ryo II-877
 Shin, Heesang I-1071
 Shouno, Osamu II-228
 Silva, Catarina I-723
 Šíma, Jiří II-179
 Soares, Carlos II-252
 Solé-Casals, Jordi I-224, II-979
 Solgi, Mojtaba I-80
 Soltic, Snjezana I-1129
 Song, Qun I-1204, I-1221, I-1245
 Sonoh, Satoshi II-1065
 Sossa, Humberto II-800
 Souza, Renata M.C.R. de II-11, II-19
 Stroobandt, Dirk I-56
 Sudo, Tamami I-377
 Suemitsu, Atsuo I-384
 Suetake, Noriaki II-583
 Suetsugu, Ken I-494
 Suh, Il Hong I-64, I-747
 Sum, John II-324, II-919
 Sum, Pui Fai II-316
 Sun, Lisha II-426
 Sun, Tieli II-501
 Sung, Andrew H. I-723
 Susnjak, Teo I-945
 Suzuki, Kenji II-244
 Suzuki, Michiyasu I-369
 Suzuki, Takeshi I-259
 Tabata, Yusuke I-14
 Tagawa, Yoshihiko I-494
 Tajima, Fumiaki II-703
 Takahashi, Haruhisa II-344, II-361, II-671
 Takahashi, Hiromu II-1013
 Takata, Mika II-369
 Takazawa, Kazuhiro I-444
 Takeda, Taichi I-851
 Takeuchi, Johane II-228
 Takeuchi, Jun'ichi I-579
 Takumi, Ichi II-1021
 Tamada, Hirotaka I-715
 Tamei, Tomoya II-1029
 Tanaka, Toshihisa I-453
 Tanaka, Toshiyuki II-155
 Tanaka, Yuji I-867
 Taniguchi, Tadahiro I-953
 Tanino, Tetsuzo I-970
 Tanscheit, Ricardo II-386
 Tatsumi, Keiji I-970
 Tay, Yong Haur II-745
 Then, Siu Jing II-808
 Theng, Lau Bee II-509, II-591, II-927
 Timm, Wiebke I-513
 Ting, K.H. I-436
 Togashi, Yuki I-293
 Tohru, Yagi II-1021
 Toledo, Franklina M.B. de I-461
 Tomita, Jyun-ichiro II-353
 Torikai, Hiroyuki I-145, I-208
 Tou, Jing Yi II-745
 Tovar, Gessyca Maria II-851
 Tran, Ha Nguyen I-851
 Tripathi, B.K. II-867
 Trujillo, Marcelo I-1211
 Tsai, I-Ling II-1038
 Tsuboyama, Manabu I-32
 Tsuda, Ichiro I-72, I-423
 Tsujino, Hiroshi II-228
 Tsukada, Minoru I-72, I-416, I-423
 Tsukada, Yuki I-469
 Tung, Sau Wai I-1137
 Uchibe, Eiji I-22
 Uchino, Eiji II-583
 Ueda, Michihito II-843
 Uezu, Tatsuya II-195

- Uota, Shiori I-240
 Utsunomiya, Hiroki II-970
- Vasilyev, Vladimir II-131
 Vázquez, Roberto A. II-800
 Vellasco, Marley II-386, II-461
 Vellasco, Pedro II-461
 Verma, Anju I-1204
 Verma, Brijesh II-639
 Versaci, Mario II-720
 Verstraeten, David I-56
 Vialatte, François-Benoit I-177, I-224, I-318, I-400, I-469, II-979
 Vieira, Armando II-97, I-723
 Vieira, V.M.M. II-575
 Vigário, Ricardo II-559
 Vijaya Rama Raju, A. II-394
 Villaverde, Ivan I-1021, I-1045
 Villmann, Thomas II-61
 Vrahatis, Michael N. II-308
- Wagatsuma, Hiroaki I-119, I-859, I-1087
 Wakita, Toshihiro I-293
 Wang, Chen-Cheng I-647
 Wang, Dianhui I-478, I-521, II-792
 Wang, Fuxiang I-737
 Wang, Hongjiang II-316
 Wang, Jun II-1003
 Wang, Linlu II-535
 Wang, Lipo II-551
 Wang, Shiyu I-1188
 Wang, Xuebing I-779
 Wang, Yuanyuan II-535, II-768
 Watanabe, Kazuho I-655
 Watanabe, Sumio I-688, I-696, II-903
 Watts, Michael J. I-901, I-909
 Weber, Theophane I-177, I-318
 Weng, Juyang I-80
 Wersing, Heiko I-805, I-813
 Widiputra, Harya I-1237
 Willert, Volker I-275
 Wilson, Peter II-1049
 Wimalaratna, Sunil II-979
 Wong, Kok Wai I-638
 Wong, Wei Kin I-1103
 Worner, Sue P. I-901, I-909
 Wu, Tony I-194
 Wu, Yongjun I-129
- Xiu, Jie I-1188
 Xiu, Yan I-1188
 Xu, Quliang I-563
- Yamada, Kazuhiro I-978
 Yamagata, Masaya I-579
 Yamaguchi, Yoko I-186, I-859
 Yamaguti, Yutaka I-72, I-423
 Yamakawa, Takeshi I-369, I-962
 Yamakawa, Toshitaka I-369
 Yamasaki, Hironobu II-663
 Yamauchi, Koichiro I-293, I-1154
 Yamazaki, Keisuke I-629
 Yang, Fengqin II-501
 Yang, Fu-Shu II-1038
 Yang, Haiqin II-53
 Yano, Masafumi I-259
 Yao, Xin I-216
 Yeh, Chien-Ting II-687
 Yeh, Chung-Hsing I-647
 Yi, Chuho I-747
 Yokoi, Hirokazu I-240, II-679
 Yonekawa, Masato II-776
 Yong, Yen San II-808
 Yoshikawa, Tomohiro II-1013
 Yoshino, Tetsuma I-621
 Yoshioka, Katsunari I-579
 Yoshioka, Taku I-336
 Yu, Yan I-571
 Yuan, Zhijian II-987
 Yun, Yeboon I-995
- Zakos, John II-639
 Zender, Paul M. II-70
 Zeng, Xiaoping II-493
 Zhang, Changhai II-501
 Zhang, Chen I-275
 Zhang, Jun I-737, II-695, II-736
 Zhang, Kau I-1163
 Zhang, Liming I-251, II-535
 Zhang, Ping I-529
 Zhang, Qi I-392
 Zhao, Zhonghua I-563
 Zhu, Dingyun II-655
 Zorkadis, Vasilios C. I-595
 Zulueta, Ekaitz I-1053
 Zuppich, Alan I-1129