

Colin Fyfe Peter Tino
Darryl Charles Cesar Garcia-Osorio
Hujun Yin (Eds.)

LNCS 6283

Intelligent Data Engineering and Automated Learning – IDEAL 2010

11th International Conference
Paisley, UK, September 2010
Proceedings

 Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Colin Fyfe Peter Tino Darryl Charles
Cesar Garcia-Osorio Hujun Yin (Eds.)

Intelligent Data Engineering and Automated Learning – IDEAL 2010

11th International Conference
Paisley, UK, September 1-3, 2010
Proceedings

Volume Editors

Colin Fyfe
University of the West of Scotland
Paisley, UK
E-mail: colin.fyfe@uws.ac.uk

Peter Tino
University of Birmingham
Birmingham, UK
E-mail: p.tino@cs.bham.ac.uk

Darryl Charles
University of Ulster
Coleraine, UK
E-mail: dk.charles@ulster.ac.uk

Cesar Garcia-Osorio
Universidad de Burgos
Burgos, Spain
E-mail: cgosorio@ubu.es

Hujun Yin
The University of Manchester
Manchester, UK
E-mail: hujun.yin@manchester.ac.uk

Library of Congress Control Number: 2010933530

CR Subject Classification (1998): I.2, H.3, H.4, F.1, H.2.8, J.3

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-642-15380-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15380-8 Springer Berlin Heidelberg New York

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

springer.com

© Springer-Verlag Berlin Heidelberg 2010
Printed in Germany

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

Preface

The IDEAL conference has become a unique, established and broad interdisciplinary forum for experts, researchers and practitioners in many fields to interact with each other and with leading academics and industries in the areas of machine learning, information processing, data mining, knowledge management, bio-informatics, neuro-informatics, bio-inspired models, agents and distributed systems, and hybrid systems.

This volume contains the papers presented at the 11th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2010), which was held September 1–3, 2010 in the University of the West of Scotland, on its Paisley campus, 15 kilometres from the city of Glasgow, Scotland. All submissions were strictly peer-reviewed by the Programme Committee and only the papers judged with sufficient quality and novelty were accepted and included in the proceedings.

The IDEAL conferences continue to evolve and this year's conference was no exception. The conference papers cover a wide variety of topics which can be classified by technique, aim or application. The techniques include evolutionary algorithms, artificial neural networks, association rules, probabilistic modelling, agent modelling, particle swarm optimization and kernel methods. The aims include regression, classification, clustering and generic data mining. The applications include biological information processing, text processing, physical systems control, video analysis and time series analysis.

The attendees at the conference came from throughout Europe, Africa, Asia, North and South America and Australia – making the conference truly international. Perhaps this reflects the fact that the IDEAL conferences have been held in venues throughout Asia and Europe and will be breaking new ground in Brazil in the near future.

We would like to thank all the people who devoted so much time and effort to the successful running of the conference and particularly the members of committees and reviewers, and of course the authors of the papers in this Proceedings. Continued support and collaboration from Springer, especially LNCS editors Alfred Hofmann and Anna Kramer, are also appreciated.

June 2009

Colin Fyfe
Peter Tino
Darryl Charles
Cesar Garcia-Osorio
Hujun Yin

International Advisory Committee

Yaser Abu-Mostafa	CALTECH, USA
Shun-ichi Amari	RIKEN, Japan
Michael Dempster	University of Cambridge, UK
José R. Dorronsoro	Autonomous University of Madrid, Spain
Nick Jennings	University of Southampton, UK
Samuel Kaski	Helsinki University of Technology, Finland
Soo-Young Lee	KAIST, South Korea
Erkki Oja	Helsinki University of Technology, Finland
Latit M. Patnaik	Indian Institute of Science, India
Burkhard Rost	Columbia University, USA
Xin Yao	University of Birmingham, UK

Steering Committee

Nigel Allinson	University of Sheffield, UK
Yiu-ming Cheung	Hong Kong Baptist University, Hong Kong
Emilio Corchado	University of Burgos, Spain
Marc van Hulle	K. U. Leuven, Belgium
John Keane	University of Manchester, UK
Jimmy Lee	Chinese University of Hong Kong, Hong Kong
Malik Magdon-Ismail	Rensselaer Polytechnic Institute, USA
Zheng Rong Yang	University of Exeter, UK
Ning Zhong	Maebashi Institute of Technology, Japan

Programme Committee

Ajith Abraham	Norwegian University of Science and Technology, Norway
Luis Alonso	University of Salamanca, Spain
Davide Anguita	University of Genoa, Italy
Bruno Apolloni	Università degli Studi of Milan, Italy
Wesam Ashour	University of Gaza, Palestine
Javier Bajo	Pontifical University of Salamanca, Spain
Bruno Baroque	University of Burgos, Spain
David Becerra Alonso	University of the West of Scotland, UK
Antonio Bella	Universidad Politécnica de Valencia, Spain
Milos Borenovic	University of Belgrade, Serbia
Lourdes Borrajo	University of Vigo, Spain
Vicente Botti	Polytechnic University of Valencia, Spain
Andrés Bustillo	University of Burgos, Spain
David Camacho	Universidad Autónoma de Madrid, Spain
Jose. Calvo-Rolle	Universidad de la Coruña, Spain
André de Carvalho	University of São Paulo, Brazil
Matthew Casey	University of Surrey, UK

Richard Chbeir	Bourgogne University, France
Seungjin Choi	POSTECH, Korea
Stelvio Cimato	University of Milan, Italy
Juan M. Corchado	University of Salamanca, Spain
Raúl Cruz-Barbosa	Universitat Politècnica de Catalunya, Spain
Leticia Curiel	University of Burgos, Spain
Alfredo Cuzzocrea	University of Calabria, Italy
Ernesto Damiani	University of Milan, Italy
Bernard de Baets	Ghent University, Belgium
María J. del Jesús	University of Jaén, Spain
Ricardo Del Olmo	University of Burgos, Spain
Fernando Díaz	University of Valladolid, Spain
José Dorronsoro	Universidad Autónoma de Madrid, Spain
Gérard Dreyfus	École Supérieure de Physique et de Chimie Industrielles de Paris, France
Jochen Einbeck	University of Durham, UK
Igor Farkas	Comenius University in Bratislava, Slovakia
Florentino Fernández	University of Vigo, Spain
Jan Feyereisl	University of Nottingham, UK
Richard Freeman	Michael Page International, UK
Marcus Gallagher	The University of Queensland, Australia
Matjaz Gams	Jozef Stefan Institute Ljubljana, Slovenia
Esteban García Cuesta	Carlos III University, Spain
Mario A. García-Martínez	Instituto Tecnológico de Orizaba, Mexico
Daniel Glez-Peña	University of Vigo, Spain
David González-Ortega	University of Valladolid, Spain
Petro Gopych	Universal Power Systems USA-Ukraine LLC, Ukraine
Marcin Gorawski	Silesian University of Technology, Poland
Lars Graening	Honda Research Institute Europe GmbH, Germany
Manuel Graña	University of Pais Vasco, Spain
Jerzy Grzymala-Busse	University of Kansas, USA
Ioannis Hatzilygeroudis	University of Patras, Greece
Francisco Herrera	University of Granada, Spain
Álvaro Herrero	University of Burgos, Spain
Michael Herrmann	University of Edinburgh, UK
James Hogan	Queensland University of Technology, Australia
Jaakko Hollmén	Helsinki University of Technology, Finland
Vasant Honavar	Iowa State University, USA
Wei-Chiang S. Hong	Oriental Institute of Technology, Taiwan
David Hoyle	University of Manchester, UK
Jose A. Iglesias	Carlos III University, Spain
Emilio Insfran	Universidad Politècnica de Valencia, Spain
Lakshmi Jain	University of South Australia, Australia
Vicent Julián	Universidad Politècnica de Valencia, Spain
Juha Karhunen	Helsinki University of Technology, Finland
Kyung-Joong Kim	Sejong University, Korea
Mario Köppen	Kyushu Institute of Technology, Japan

Andreas König	University of Kaiserslautern, Germany
Zofia Kruczkiewicz	Wrocław University of Technology, Poland
Pei Ling Lai	Southern Taiwan University, Taiwan
Benaki Lairenjam	JMI University New Delhi, India
Paulo Lisboa	Liverpool John Moores University, UK
Lenka Lhotská	Czech Technical University, Czech Republic
Eva Lorenzo	University of Vigo, Spain
Wenjian Luo	University of Science and Technology of China, China
Roque Marín	University of Murcia, Spain
José F. Martínez	Instituto Nacional de Astrofísica Óptica y Electrónica, Mexico
Giancarlo Mauri	University of Milano Bicocca, Italy
Michael McCreedy	University of the West of Scotland, UK
José R. Méndez	Universidade de Vigo, Spain
José M. Molina	University Carlos III of Madrid, Spain
Carla Möller-Levet	University of Manchester, UK
Yusuke Nojima	Osaka Prefecture University, Japan
Diego Ordóñez	Universidade da Coruña, Spain
Chung-Ming Ou	Kainan University, Taiwan
Seiichi Ozawa	Kobe University, Japan
Vasile Palade	University of Oxford, UK
Stephan Pareigis	Hamburg University of Applied Sciences, Germany
Juan Pavón	University Complutense of Madrid, Spain
Carlos Pereira	University of Coimbra, Portugal
Gloria Phillips-Wren	Loyola College, USA
Victor Rayward-Smith	University of East Anglia, UK
Bernardete Ribeiro	University of Coimbra, Portugal
Fabrice Rossi	Télécom Paris Tech, France
Roberto Ruiz	Pablo de Olavide University, Spain
Sherif Sakr	University of New South Wales, Australia
Yanira Santana	University of Salamanca, Spain
José Santos	Universidade da Coruña, Spain
Javier Sedano	University of Burgos, Spain
Dragan Simic	Novi Sad Fair, Serbia
Michael Small	Hong Kong Polytechnic University, Hong Kong
Ying Tan	Peking University, China
Ke Tang	University of Science and Technology of China, China
Dante I. Tapia	University of Salamanca, Spain
Alicia Troncoso	Pablo de Olavide University, Spain
Eiji Uchino	Yamaguchi University, Japan
Alfredo Vellido	Universidad Politécnica de Cataluña, Spain
Sebastián Ventura Soto	University of Cordoba, Spain
José R. Villar	University of Oviedo, Spain
Michal Wozniak	Wrocław University of Technology, Poland
Du Zhang	California State University, USA
Huiyu Zhou	Queen's University Belfast, UK
Rodolfo Zunino	University of Genoa, Italy

Local Organizing Committee

Colin Fyfe	University of the West of Scotland, UK
Daniels Livingstone	University of the West of Scotland, UK
Qi Wang	University of the West of Scotland, UK
Lidia Wojtowicz	University of the West of Scotland, UK
Xiaochuan Wu	University of the West of Scotland, UK
Jigang Sun	University of the West of Scotland, UK

Table of Contents

Large Scale Instance Selection by Means of a Parallel Algorithm	1
<i>Aida de Haro-García, Juan Antonio Romero del Castillo, and Nicolás García-Pedrajas</i>	
Typed Linear Chain Conditional Random Fields and Their Application to Intrusion Detection	13
<i>Carsten Elfers, Mirko Horstmann, Karsten Sohr, and Otthein Herzog</i>	
Generalized Derivative Based Kernelized Learning Vector Quantization	21
<i>Frank-Michael Schleif, Thomas Villmann, Barbara Hammer, Petra Schneider, and Michael Biehl</i>	
Cost Optimization of a Localized Irrigation System Using Genetic Algorithms	29
<i>Mônica Sakuray Pais, Júlio César Ferreira, Marconi Batista Teixeira, Keiji Yamanaka, and Gilberto Arantes Carrijo</i>	
Dimension Reduction for Regression with Bottleneck Neural Networks	37
<i>Elina Parviainen</i>	
Analysing Satellite Image Time Series by Means of Pattern Mining	45
<i>François Petitjean, Pierre Gançarski, Florent Masseglia, and Germain Forestier</i>	
Sentences Generation by Frequent Parsing Patterns	53
<i>Takashi Yanagisawa, Takao Miura, and Isamu Shioya</i>	
Gallbladder Boundary Segmentation from Ultrasound Images Using Active Contour Model	63
<i>Marcin Ciecholewski</i>	
On the Power of Topological Kernel in Microarray-Based Detection of Cancer	70
<i>Vilen Jumutc and Pawel Zayakin</i>	
An Evolutionary Multi-objective Optimization of Market Structures Using PBIL	78
<i>Xinyang Li and Andreas Krause</i>	
New Application of Graph Mining to Video Analysis	86
<i>Hisashi Koga, Tsuji Tomokazu, Takanori Yokoyama, and Toshinori Watanabe</i>	

Classification by Multiple Reducts-kNN with Confidence	94
<i>Naohiro Ishii, Yuichi Morioka, Hiroaki Kimura, and Yongguang Bao</i>	
Towards Automatic Classification of Wikipedia Content	102
<i>Julian Szymański</i>	
Investigating the Behaviour of Radial Basis Function Networks in Regression and Classification of Geospatial Data	110
<i>Andrea Guidali, Elisabetta Binaghi, Mauro Guglielmin, and Marco Pascale</i>	
A Comparison of Three Voting Methods for Bagging with the MLEM2 Algorithm	118
<i>Clinton Cohagan, Jerzy W. Grzymala-Busse, and Zdzislaw S. Hippe</i>	
Simplified Self-adapting Skip Lists	126
<i>Jonathan J. Pittard and Alan L. Tharp</i>	
Multi-Agent Architecture with Support to Quality of Service and Quality of Control	137
<i>Jose-Luis Poza-Luján, Juan-Luis Posadas-Yagüe, and Jose-Enrique Simó-Ten</i>	
Robust 1-Norm Soft Margin Smooth Support Vector Machine	145
<i>Li-Jen Chien, Yuh-Jye Lee, Zhi-Peng Kao, and Chih-Cheng Chang</i>	
A Generalization of Independence in Naive Bayes Model	153
<i>Yu Fujimoto and Noboru Murata</i>	
Interval Filter: A Locality-Aware Alternative to Bloom Filters for Hardware Membership Queries by Interval Classification	162
<i>Ricardo Quislan, Eladio Gutierrez, Oscar Plata, and Emilio L. Zapata</i>	
Histogram Distance for Similarity Search in Large Time Series Database	170
<i>Yicun Ouyang and Feng Zhang</i>	
The Penalty Avoiding Rational Policy Making Algorithm in Continuous Action Spaces	178
<i>Kazuteru Miyazaki</i>	
Applying Clustering Techniques to Reduce Complexity in Automated Planning Domains	186
<i>Luke Dicken and John Levine</i>	
The M-OLAP Cube Selection Problem: A Hyper-polymorphic Algorithm Approach	194
<i>Jorge Loureiro and Orlando Belo</i>	

Privacy Preserving Technique for Euclidean Distance Based Mining Algorithms Using a Wavelet Related Transform	202
<i>Mohammad Ali Kadampur and Somayajulu D.V.L.N</i>	
Extracting Features from an Electrical Signal of a Non-Intrusive Load Monitoring System	210
<i>Marisa B. Figueiredo, Ana de Almeida, Bernardete Ribeiro, and António Martins</i>	
Annotation and Retrieval of Cell Images	218
<i>Maria F. O'Connor, Arthur Hughes, Chaoxin Zheng, Anthony Davies, Dermot Kelleher, and Khurshid Ahmad</i>	
Adaptive Particle Swarm Optimizer for Feature Selection	226
<i>M.A. Esseghir, Gilles Goncalves, and Yahya Slimani</i>	
A Randomized Sphere Cover Classifier	234
<i>Reda Younsi and Anthony Bagnall</i>	
Directed Figure Codes with Weak Equality	242
<i>Włodzimierz Moczurad</i>	
Surrogate Model for Continuous and Discrete Genetic Optimization Based on RBF Networks	251
<i>Lukáš Bajer and Martin Holeňa</i>	
Relevance of Contextual Information in Compression-Based Text Clustering	259
<i>Ana Granados, Rafael Martínez, David Camacho, and Francisco de Borja Rodríguez</i>	
Simple Deterministically Constructed Recurrent Neural Networks	267
<i>Ali Rodan and Peter Tiňo</i>	
Non-negative Matrix Factorization Implementation Using Graphic Processing Units	275
<i>Noel Lopes and Bernardete Ribeiro</i>	
A Neighborhood-Based Clustering by Means of the Triangle Inequality	284
<i>Marzena Kryszkiewicz and Piotr Lasek</i>	
Selection of Structures with Grid Optimization, in Multiagent Data Warehouse	292
<i>Marcin Gorawski, Sławomir Bańkowski, and Michał Gorawski</i>	
Approximating the Covariance Matrix of GMMs with Low-Rank Perturbations	300
<i>Malik Magdon-Ismail and Jonathan T. Purnell</i>	

Learning Negotiation Policies Using IB3 and Bayesian Networks	308
<i>Gislaine M. Nalepa, Bráulio C. Ávila, Fabrício Enembreck, and Edson E. Scalabrin</i>	
Trajectory Based Behavior Analysis for User Verification	316
<i>Hsing-Kuo Pao, Hong-Yi Lin, Kuan-Ta Chen, and Junaidillah Fadlil</i>	
Discovering Concept Mappings by Similarity Propagation among Substructures	324
<i>Qi H. Pan, Fedja Hadzic, and Tharam S. Dillon</i>	
Clustering and Visualizing SOM Results	334
<i>José Alfredo F. Costa</i>	
A Hybrid Evolutionary Algorithm to Quadratic Three-Dimensional Assignment Problem with Local Search for Many-Core Graphics Processors	344
<i>Piotr Lipinski</i>	
Evolution Strategies for Objective Functions with Locally Correlated Variables	352
<i>Piotr Lipinski</i>	
Neural Data Analysis and Reduction Using Improved Framework of Information-Preserving EMD	360
<i>Zareen Mehboob and Hujun Yin</i>	
Improving the Performance of the Truncated Fourier Series Least Squares (TFSLs) Power System Load Model Using an Artificial Neural Network Paradigm	368
<i>Shonique L. Miller, Gary L. Leiby, and Ali R. Osareh</i>	
An Efficient Approach to Clustering Real-Estate Listings	379
<i>Maciej Grzenda and Deepak Thukral</i>	
Incremental Update of Cyclic Association Rules	387
<i>Eya Ben Ahmed</i>	
Author Index	397

Large Scale Instance Selection by Means of a Parallel Algorithm^{*}

Aida de Haro-García, Juan Antonio Romero del Castillo,
and Nicolás García-Pedrajas

Department of Computing and Numerical Analysis of the University of Córdoba,
Campus de Rabanales, 14071 Córdoba, Spain
{aromero, adeharo, npedrajas}@uco.es

Abstract. Instance selection is becoming more and more relevant due to the huge amount of data that is constantly being produced. However, although current algorithms are useful for fairly large datasets, many scaling problems are found when the number of instances is of hundred of thousands or millions. Most instance selection algorithms are of complexity at least $O(n^2)$, n being the number of instances. When we face huge problems, the scalability becomes an issue, and most of the algorithms are not applicable.

This paper presents a way of removing this difficulty by means of a parallel algorithm that performs several rounds of instance selection on subsets of the original dataset. These rounds are combined using a voting scheme to allow a very good performance in terms of testing error and storage reduction, while the execution time of the process is decreased very significantly. The method is specially efficient when we use instance selection algorithms that are of a high computational cost.

An extensive comparison in 35 datasets of medium and large sizes from the UCI Machine Learning Repository shows the usefulness of our method. Additionally, the method is applied to 6 huge datasets (from three hundred thousands to more than four millions instances) with very good results and fast execution time.

1 Introduction

The overwhelming amount of data that is available nowadays [4] in any field of research poses new problems for data mining and knowledge discovery methods. This huge amount of data makes most of the existing algorithms inapplicable to many real-world problems. Data reduction consists of removing missing, redundant and/or erroneous data to get a tractable amount of information. One of the most common methods for data reduction is instance selection.

Instance selection [6] consists of choosing a subset of the total available data to achieve the original purpose of the data mining application as if the whole

^{*} This work was supported in part by the Project TIN2008-03151 of the Spanish Ministry of Science and Innovation and the Project of Excellence in Research P09-TIC-04623 of the Junta de Andalucía.

data is used. Different variants of instance selection exist. Our aim is focused on instance selection for instance-based learning.

In a previous paper [5] we proposed an algorithm called *democratic* instance selection that was able to achieve a large reduction in the execution time of the instance selection algorithms while keeping their performance. The underlying idea was based upon the following premises:

1. A very promising way of scaling up instance selection algorithms is using smaller subsets. A simple way of doing that is partitioning the dataset into disjoint subsets and applying the instance selection algorithm to each subset separately.
2. The above solution does not perform well, as each subset is only a partial view of the original dataset. In this way, important instances may be removed and superfluous instances may be kept. In the same sense as we talk of “weak learners” in a classifier ensemble construction framework, we can consider an instance selection algorithm applied to a subset of the whole dataset as a “*weak* instance selection algorithm”.
3. Following the philosophy of classifier ensembles we can carry out several rounds of weak instance selection algorithms and combine them using a voting scheme. Therefore, our approach is called *democratic* instance selection, and can be considered a form of extending classifier ensemble philosophy to instance selection.

Democratic instance selection is thus based on repeating several rounds of a fast instance selection process. Each round on its own would not be able to achieve a good performance. However, the combination of several rounds using a voting scheme is able to match the performance of an instance selection algorithm applied to the whole dataset with a large reduction in the time of the algorithm. Thus, in a different setup from the case of ensembles of classifiers, we can consider our method a form of “ensembling” instance selection.

In this paper we show a parallel implementation of this method that is able to achieve a tremendous reduction in the execution time of any instance selection algorithm while keeping its performance. The main advantage of our method is that as the instance selection algorithm is applied only to small subsets the time is reduced very significantly. In fact, as the size of the subset is chosen by the researcher, we can apply the method to any problem regardless of the number of instances involved. As for the case of classifier ensembles, where the base learner is a parameter of the algorithm, in our method the instance selection method is a parameter, and any algorithm can be used.

A further advantage is the reduction in memory storage requirements. The algorithm does not need to have in memory the whole dataset. For huge problems where we have millions of instances it means that we can perform the instance selection when other algorithms would be limited by the amount of available memory.

This paper is organized as follows: Section 2 presents the proposed model for instance selection based on our approach and Section 3 shows its implementation; Section 4 shows the results of the experiments; and Section 5 states the conclusions of our work and future research lines.

2 Parallel Democratic Instance Selection Method

The process consists of dividing the original dataset into several disjoint subsets of approximately the same size that cover all the dataset. Then, the instance selection algorithm is applied to each subset separately. The instances that are selected to be removed by the algorithm receive a vote. Then, a new partition is performed and another round of votes is carried out. After the predefined number of rounds is made, the instances which have received a number of votes above a certain threshold are removed. Each round can be considered to be similar to a classifier in an ensemble, and the combination process by voting is similar to the combination of base learners in bagging or boosting [7].

The most important advantage of our method is the large reduction in execution time. The reported experiments will show a large difference when using standard widely used instance selection algorithms.

An important step in our method is partitioning the training set into a number of disjoint subsets, t_i , which comprise the whole training set, $\bigcup_i t_i = T$. The size of the subsets is fixed by the user. The actual size has no relevant influence over the results provided it is small enough to avoid large execution time. Furthermore, the time spent by the algorithm depends on the size of the largest subset, so it is important that the partition algorithm produces subsets of approximately equal size.

The first thing that we must take into account is the fact that we need several partitions of the dataset, as each round of votes needs a different partition. Otherwise, the votes cast will be the same as most instance selection algorithms are deterministic. The first goal of our partition method is keeping, as much as possible, a certain locality in the partition. But there is an additional, and more subtle point, about the partition that is of the utmost importance for the performance of the method.

Each partition represents a different optimization problem, and so a different error surface for the instance selection algorithm. If the partitions are very different, these error surfaces will also be very different. In such a case, the votes cast by the different rounds are almost randomly distributed, and the obtained performance is poor. Thus, to obtain a good performance the partitions of the different rounds of the algorithm must vary smoothly.

These two previous requirements, partitions that keep certain spatial consistency and that vary smoothly are obtained using the theory of Grand Tour [1]. A vector is rotated in the multidimensional space using Grand Tour and the dataset is projected into this vector. The subsets are obtained from the order induced by this projection.

2.1 Determining the Number of Votes

An important issue in our method is determining the number of votes needed to remove an instance from the training set. Preliminary experiments showed that this number highly depends on the specific dataset. Thus, it is not possible to set a general preestablished value usable in any dataset. On the contrary, we need a way of selecting this value directly from the dataset in run time. Our method to obtain this threshold is based on estimating the best value for the number of votes from the effect on the training set. The election of the number of votes must take into account two different criteria: training error, ϵ_t , and storage, or memory, requirements m . Both values must be minimized as much as possible. Our method of choosing the number of votes needed to remove an instance is based on obtaining the threshold number of votes, v , that minimizes a fitness criterion, $f(v)$, which is a combination of these two values:

$$f(v) = \alpha\epsilon_t(v) + (1 - \alpha)m(v), \quad (1)$$

We perform r rounds of the algorithm and store the number of votes received by each instance. Then, we must obtain the threshold number of votes, v , to remove an instance. This value must be $v \in [1, r]$. We calculate the criterion $f(v)$ (eq. [1](#)) for all the possible threshold values from 1 to r , and assign v to the value which minimizes the criterion. After that, we perform the instance selection removing the instances whose number of votes is above or equal to the obtained threshold v . In this way, the evaluation of each threshold of votes is also democratized.

2.2 Complexity of Our Methodology

We divide the dataset into partitions of disjoint subsets of size s . Thus, the chosen instance selection algorithm is always applied to a subset of fixed size, s . The complexity of this application of the algorithm depends on the base instance selection algorithm we are using, but will always be small, as the size s is always small. Let K be the number of operations needed by the instance selection algorithm to perform its task in a dataset of size s . For a dataset of n instances we must perform this instance selection process once for each subset, that is n/s times, spending a time proportional to $(n/s)K$. The total time needed by the algorithm to perform r rounds will be proportional to $r(n/s)K$, which is linear in the number of instances, as K is a constant value. Furthermore, if we have $r(n/s)$ processors for distributing the selection algorithms all of them can be performed concurrently with a complexity that is constant. Figure [1](#) shows the computational cost, as a function of the number of instances, of a quadratic algorithm and our approach when that algorithm is used with subset sizes of $s = 100, 1000, 2500$ and 5000 instances, $r = 10$ rounds of votes and 256 processors.

If the complexity of the instance selection algorithm is greater, the reduction of the execution will be even better. As the application of the instance selection algorithm to each subset is independent from all the remaining subsets, all the

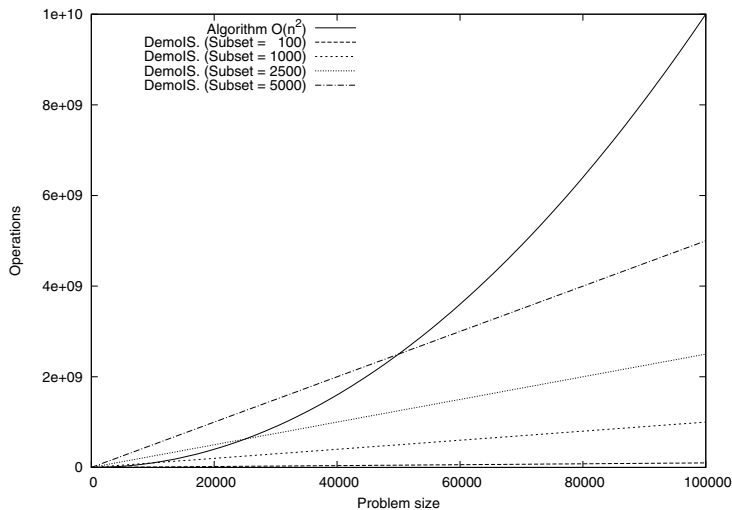


Fig. 1. Computational cost, in logarithmic scale, of our method and a base instance selection algorithm of $O(n^2)$ and 256 processors

subsets can be processed at the same time, even for different rounds of votes. Also, the communication between the nodes of the parallel execution is small.

As we have stated, two additional processes complete the method, the partition of the dataset and the determination of the number of votes. Regarding the determination of the number of votes, as the process is also *democratized* and parallelized, it is also of linear complexity.

The partition described can be implemented with a complexity $O(n \log(n))$, using a quicksort algorithm for sorting the values to make the subsets, or with a complexity $O(n)$ dividing the projection along the vector in equal sized intervals. Both methods achieve the same performance as the obtained partition is very similar, and in our experiments we have used the latter to keep the complexity of the whole procedure linear. However, this partition is specially designed for k -NN classifier. When the method is used with other classifiers, other methods can be used, such as a random partition, which is also of complexity $O(n)$.

3 Parallel Implementation

The parallel implementation is based on a master/slave architecture. The master performs the partition of the dataset and sends the subsets to each slave. Each slave performs the instance selection algorithm using only the instances of its subset and then returns the selected instances to the master. The master stores the votes for each removed instance and perform a new round. The general architecture of the system is shown in Figure 2. As each round is independent of the previous one all of them are performed in parallel. This method has the

advantage that it is still applicable for huge datasets, as only a small part of the dataset must be kept in memory.

The threshold of votes is obtained using the same parallel *democratic* approach. Again, we divide the dataset into disjoint subsets and evaluate the application of each threshold on every subset separately. The value of the goodness of a threshold is the average value of evaluating eq. 1 in each subset.

4 Experimental Results

In order to make a fair comparison between the standard algorithms and our proposal, we have selected a set of 35 problems from the UCI Machine Learning Repository. For estimating the storage reduction and generalization error we used 10-fold cross-validation. The source code, in C and licensed under the GNU General Public License, used for all methods as well as the partitions of the datasets are freely available upon request to the authors.

We have chosen to test our model two of the most successful state-of-the-art algorithms: DROP3 [8], and ICF [2]. However, as overall results of standard DROP3 were better than ICF only the results with the former are reported.

As an alternative to these standard methods, genetic algorithms have been applied to instance selection, considering this task to be a search problem. Cano et al. [3] performed a comprehensive comparison of the performance of different evolutionary algorithms for instance selection and found that evolutionary based methods were able to outperform classical algorithms in both classification accuracy and data reduction. Among the evolutionary algorithms, CHC was able to achieve the best overall performance.

Nevertheless, the major problem addressed when applying genetic algorithms to instance selection is the scaling of the algorithm. As the number of instances grows, the time needed for the genetic algorithm to reach a good solution increases exponentially, making it totally useless for large problems. As we are concerned with this problem, we have used as second instance selection method a genetic algorithm using CHC methodology.

Results for the standard and democratic version are shown in Table 1 for DROP3 and in Table 2 for CHC. These results are illustrated in Figures 3 and 4 respectively. For DROP3 we used a value of $k = 3$. For CHC we used a value of $k = 1$, a population of 100 individuals evolved for 1000 generations with a mutation probability of 10% and a bit mutation probability of 10%. RNN mutation was applied with a probability of 5%. These same values are used each time any of these methods is applied.

For DROP3, in terms of storage reduction our method is able to match the results of the standard method, and even improve them for some datasets. In terms of testing error its performance is slightly worse. In terms of execution time the improvement is very significant. In the most extreme case, shuttle dataset, DEMOIS.drop3 is more than 11,000 times faster than standard DROP3.

For CHC, both in terms of storage reduction and testing error our method is able to match the results of the standard method, and even improve them

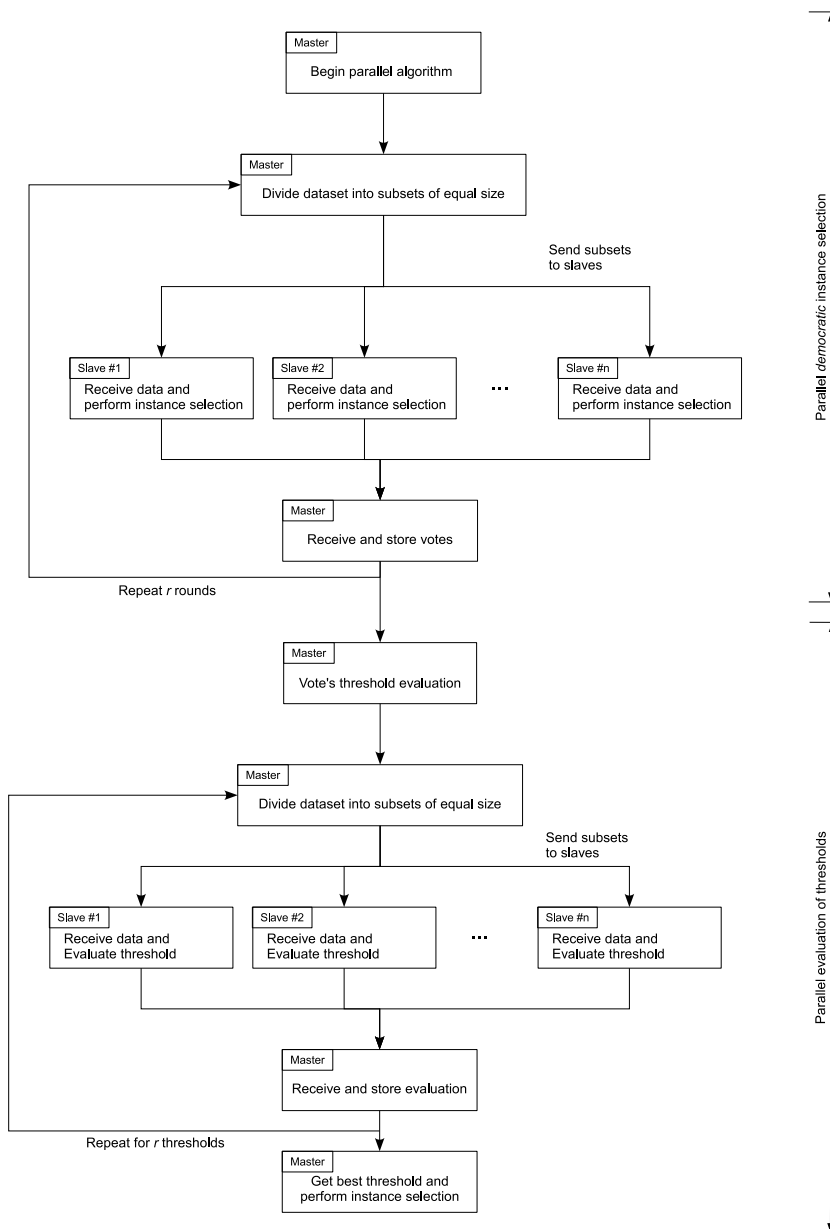


Fig. 2. Parallel implementation of *democratic* instance selection

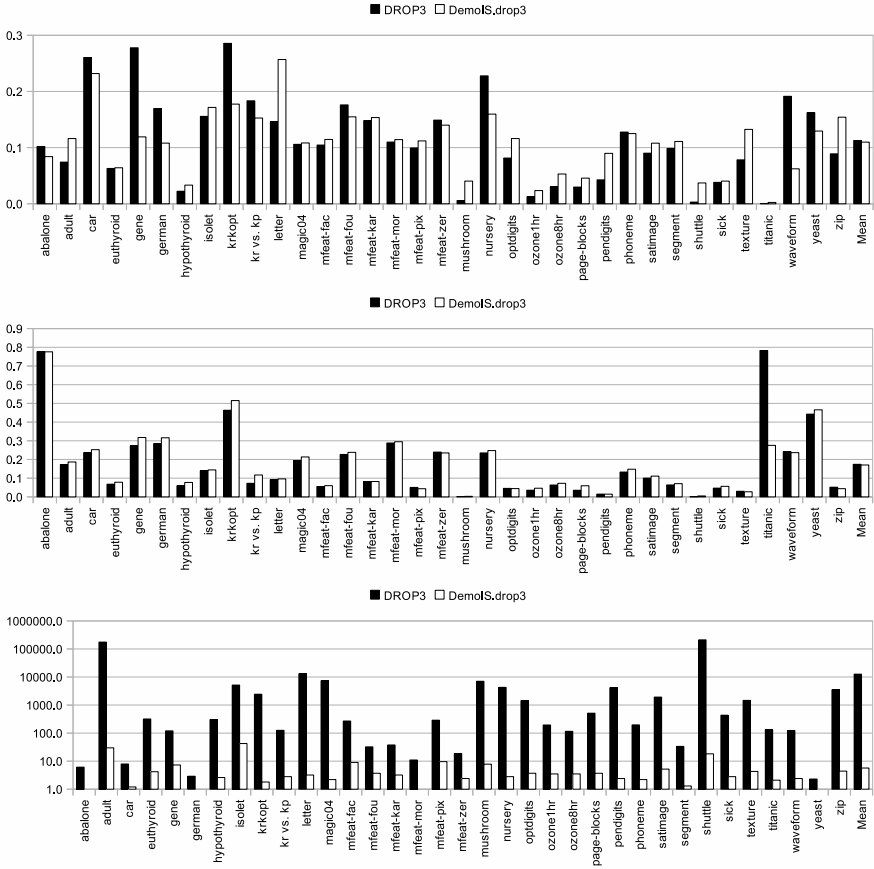


Fig. 3. Standard DROP3 and DEMOIS.drop3 results. Error (top) storage requirements (middle) and time in seconds and logarithmic scale (bottom).

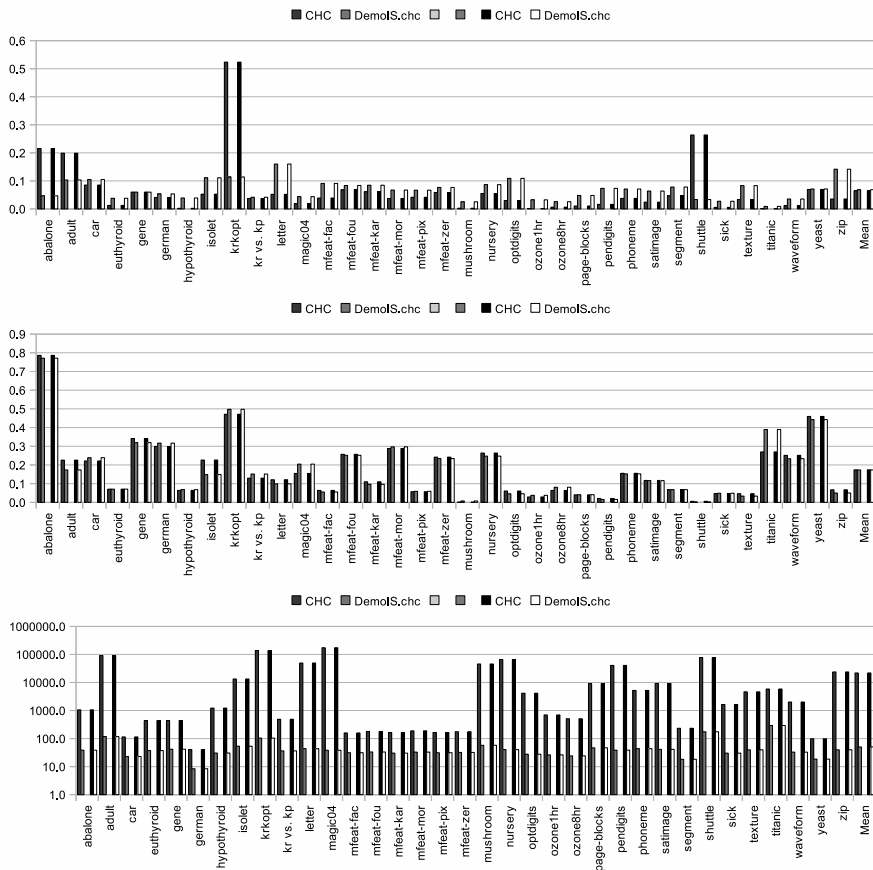


Fig. 4. Standard CHC and DEMOIS.chc results. Error (top) storage requirements (middle) and time in seconds and logarithmic scale (bottom).

Table 1. Summary of results for standard DROP3 and our parallel implementation DEMOIS.drop3

Dataset	DROP3			DEMOIS.drop3		
	Storage	Error	Time (s)	Storage	Error	Time (s)
abalone	0.1020	0.7763	6.1	0.0839	0.7758	0.7
adult	0.0741	0.1715	175976.4	0.1161	0.1867	29.8
car	0.2604	0.2366	7.9	0.2318	0.2523	1.2
euthyroid	0.0629	0.0677	319.1	0.0640	0.0782	4.2
gene	0.2778	0.2729	120.3	0.1191	0.3177	7.3
german	0.1696	0.2850	2.9	0.1080	0.3160	0.9
hypothyroid	0.0223	0.0594	301.9	0.0331	0.0772	2.6
isolet	0.1555	0.1392	5166.5	0.1715	0.1442	42.4
krkopt	0.2856	0.4632	2435.1	0.1775	0.5151	1.8
kr vs. kp	0.1833	0.0724	125.7	0.1526	0.1169	2.8
letter	0.1463	0.0922	13073.6	0.2568	0.0963	3.2
magic04	0.1058	0.1945	7352.8	0.1082	0.2134	2.2
mfeat-fac	0.1045	0.0555	269.8	0.1145	0.0600	9.0
mfeat-fou	0.1760	0.2270	32.3	0.1548	0.2380	3.7
mfeat-kar	0.1481	0.0810	37.8	0.1535	0.0820	3.2
mfeat-mor	0.1098	0.2880	11.0	0.1142	0.2950	1.0
mfeat-pix	0.0994	0.0505	288.6	0.1119	0.0435	9.6
mfeat-zer	0.1489	0.2395	18.5	0.1399	0.2350	2.4
mushroom	0.0058	0.0016	7049.0	0.0404	0.0028	7.8
nursery	0.2277	0.2349	4282.5	0.1596	0.2475	2.8
optdigits	0.0814	0.0454	1444.0	0.1161	0.0441	3.7
ozone1hr	0.0127	0.0356	194.6	0.0234	0.0462	3.5
ozone8hr	0.0303	0.0632	115.7	0.0529	0.0723	3.5
page-blocks	0.0297	0.0353	512.8	0.0456	0.0594	3.7
pendigits	0.0426	0.0147	4143.2	0.0897	0.0145	2.4
phoneme	0.1276	0.1326	196.7	0.1249	0.1478	2.2
satimage	0.0900	0.1003	1925.0	0.1079	0.1110	5.2
segment	0.0985	0.0636	33.6	0.1109	0.0706	1.3
shuttle	0.0030	0.0012	211772.1	0.0371	0.0049	18.2
sick	0.0381	0.0467	435.2	0.0403	0.0562	2.8
texture	0.0782	0.0300	1466.8	0.1324	0.0271	4.3
titanic	0.0004	0.7827	132.2	0.0023	0.2755	2.1
waveform	0.1913	0.2412	124.0	0.0622	0.2366	2.4
yeast	0.1621	0.4426	2.3	0.1294	0.4655	0.4
zip	0.0890	0.0516	3507.9	0.1541	0.0438	4.4
average	0.1126	0.1742	12653.83	0.1097	0.1705	5.68

Table 2. Summary of results for standard CHC algorithm and our parallel implementation DEMOIS.chc

Dataset	DROP3			DEMOIS.drop3		
	Storage	Error	Time (s)	Storage	Error	Time (s)
abalone	0.2152	0.7861	1060.7	0.0468	0.7712	38.6
adult	0.1988	0.2257	91096.0	0.1032	0.1734	117.7
car	0.0852	0.2215	115.0	0.1049	0.2390	22.9
euthyroid	0.0125	0.0706	436.7	0.0380	0.0706	37.3
gene	0.0601	0.3416	439.9	0.0600	0.3196	41.7
german	0.0408	0.2990	40.4	0.0536	0.3170	8.4
hypothyroid	0.0022	0.0639	1221.2	0.0391	0.0679	30.6
isolet	0.0527	0.2263	13351.4	0.1112	0.1485	53.6
krkopt	0.5237	0.4711	137397.7	0.1141	0.4978	104.5
kr vs. kp	0.0372	0.1295	488.9	0.0417	0.1514	36.5
letter	0.0521	0.1210	49443.0	0.1601	0.0992	43.8
magic04	0.0185	0.1546	173970.0	0.0438	0.2045	38.4
mfeat-fac	0.0390	0.0640	159.4	0.0910	0.0555	31.3
mfeat-fou	0.0678	0.2570	179.4	0.0837	0.2515	33.4
mfeat-kar	0.0612	0.1100	163.7	0.0846	0.0965	30.3
mfeat-mor	0.0373	0.2880	189.5	0.0673	0.2970	33.0
mfeat-pix	0.0415	0.0570	165.0	0.0670	0.0590	31.3
mfeat-zer	0.0587	0.2420	176.0	0.0767	0.2340	32.1
mushroom	0.0024	0.0010	45829.6	0.0255	0.0075	57.6
nursery	0.0549	0.2637	66177.1	0.0866	0.2472	40.5
optdigits	0.0302	0.0605	4141.4	0.1091	0.0459	27.7
ozone1hr	0.0016	0.0289	698.4	0.0324	0.0376	26.3
ozone8hr	0.0064	0.0636	511.4	0.0257	0.0806	24.2
page-blocks	0.0104	0.0402	9113.1	0.0482	0.0408	46.8
pendigits	0.0163	0.0207	40666.3	0.0734	0.0155	38.7
phoneme	0.0375	0.1552	5219.1	0.0710	0.1524	43.7
satimage	0.0241	0.1173	9211.2	0.0638	0.1162	41.3
segment	0.0465	0.0680	232.9	0.0782	0.0680	18.3
shuttle	0.2638	0.0055	77089.0	0.0334	0.0019	175.0
sick	0.0056	0.0472	1634.4	0.0274	0.0488	30.3
texture	0.0333	0.0465	4649.1	0.0831	0.0335	39.7
titanic	0.0012	0.2700	5860.7	0.0090	0.3895	294.1
waveform	0.0117	0.2512	2011.7	0.0354	0.2334	32.9
yeast	0.0694	0.4595	98.7	0.0712	0.4426	18.5
zip	0.0355	0.0673	23800.0	0.1417	0.0501	39.9
average	0.0644	0.1741	21915.37	0.0686	0.1733	50.31

Table 3. Summary of results for standard CHC algorithm and our parallel implementation DEMOIS.chc

Dataset	1-NN Error Instances		DEMOIS.drop3			DEMOIS..chc		
			Storage	Error	Time (s)	Storage	Error	Time (s)
census	0.0743	299,285	0.0509	0.0874	869.5	0.0481	0.0769	417.8
covtype	0.3024	581,012	0.1297	0.3666	1343.8	0.1183	0.4143	106.5
kddcup99	0.0006	494,021	0.0206	0.0011	17151.0	0.0212	0.0011	1248.2
kddcup991M	0.0002	1,000,000	0.0165	0.0007	30621.6	0.0182	0.0007	1840.8
kddcup99all	0.0000	4,898,431	0.0144	0.0002	163062.5	0.0149	0.0008	6093.5
poker	0.4975	1,025,103	0.0132	0.5038	2355.5	0.0276	0.5065	77.4

for some datasets. In terms of execution time the improvement is again very significant. In the most extreme case, magic04 dataset, DEMOIS.chc is more than 4,500 times faster than standard CHC.

Finally, we applied our parallel algorithm to several huge problems. We used the 6 problems from the UCI Machine Learning repository that are shown in Table 3. The table shows the potential of our parallel algorithm. Even for a dataset with more than four million instances, kddcup99all, the method is able to obtain a result in the very reduced amount of time of 49 hours. If we take into account that standard CHC needed 25 hours for adult dataset, which has only 48,842 instances, we can get a clear idea of the usefulness of our approach. Furthermore, the parallel version is able to keep the performance of the original algorithm, achieving for this dataset a reduction of 98.56% and a testing error of 0.02%. The results for DROP3 are also very good, performing the instance selection process in less than two hours for kddcup99all.

5 Conclusions and Future Work

In this paper we have presented a new method for scaling up instance selection algorithms that is applicable to any instance selection method without any modification. The method consists of performing several parallel rounds of applying instance selection on disjoint subsets of the original dataset and combining them by means of a voting method. Using two well-known instance selection algorithms, DROP3 and a CHC genetic algorithm, we have shown that our method is able to match the performance of the original algorithms with a considerable reduction in execution time. In terms of reduction of storage requirements and testing error, our approach is even better than the use of the original instance selection algorithm over the whole dataset for some of the problems.

We have also shown that our approach is able to scale up to huge problems with hundreds of thousands of instances. Using six of those huge datasets our method is able to execute rapidly, achieving a significant reduction of storage while keeping the testing error similar to the 1-NN error using the whole dataset. We think that the proposed method might be a breakthrough in instance selection algorithms design, because it allows the development of more complex

methods for instance selection. This is due to the relaxation of the constraints on the complexity of the base method through the possibility of using parallel democratic instance selection.


References

1. Asimov, D.: The Grand Tour: a Tool for Viewing Multidimensional Data. *SIAM Journal on Scientific and Statistical Computing* 6(1), 128–143 (1985)
2. Brighton, H., Mellish, C.: Advances in Instance Selection for Instance-Based Learning Algorithms. *Data Mining and Knowledge Discovery* 6, 153–172 (2002)
3. Cano, J.R., Herrera, F., Lozano, M.: Using Evolutionary Algorithms as Instance Selection for Data Reduction in KDD: An Experimental Study. *IEEE Transactions on Evolutionary Computation* 7(6), 561–575 (2003)
4. Craven, M., DiPasquoa, D., Freitagb, D., McCalluma, A., Mitchella, T., Nigama, K., Slatteyya, S.: Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence* 118(1–2), 69–113 (2000)
5. García-Osorio, C., de Haro-García, A., García-Pedrajas, N.: Democratic Instance Selection: a linear complexity instance selection algorithm based on classifier ensemble concepts. *Artificial Intelligence* (2010)
6. Liu, H., Motada, H., Yu, L.: A selective sampling approach to active feature selection. *Artificial Intelligence* 159(1-2), 49–74 (2004)
7. Schapire, R.E., Freund, Y., Bartlett, P.L., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics* 26(5), 1651–1686 (1998)
8. Wilson, D.R., Martinez, T.R.: Reduction Techniques for Instance-Based Learning Algorithms. *Machine Learning* 38, 257–286 (2000)

Typed Linear Chain Conditional Random Fields and Their Application to Intrusion Detection

Carsten Elfers, Mirko Horstmann, Karsten Sohr, and Otthein Herzog

Center for Computing and Communication Technologies
Am Fallturm 1, 28359 Bremen, Germany
{celfers,mir,sohr,herzog}@tzi.de
<http://www.tzi.de>

Abstract. Intrusion detection in computer networks faces the problem of a large number of both false alarms and unrecognized attacks. To improve the precision of detection, various machine learning techniques have been proposed. However, one critical issue is that the amount of reference data that contains serious intrusions is very sparse. In this paper we present an inference process with linear chain conditional random fields that aims to solve this problem by using domain knowledge about the alerts of different intrusion sensors represented in an ontology .

1 Introduction

Computer networks are subject to constant attacks, both targeted and unsuspected, that exploit the vast amount of existing vulnerabilities in computer systems. Among the measures a network administrator can take against this growing problem are intrusion detection systems (IDS). These systems recognize adversary actions in a network through either a set of rules with signatures that match against the malicious data stream or detection of anomalous behavior in the network traffic. Whereas the former will not recognize yet unknown vulnerability exploits (*zero-day*) due to the lack of respective signatures, the latter has an inherent problem with false positives. Anomalies may also be caused by a shift in the network users' behavior even when their actions are entirely legitimate (see [\[12\]](#)). One strategy is to combine the signature and the anomaly detectors to a hybrid IDS by learning which detection method is reliable for a given situation (e.g. [\[4\]](#)). In this setup detecting false positives is the challenging task to avoid overwhelming the users of an IDS with irrelevant alerts but without missing any relevant ones.

Several well-known machine learning methods have already been applied therefore to the domain of intrusion detection, e.g., Bayesian networks for recognizing attacks based on attack-trees [\[11\]](#) and (hidden colored) Petri nets to infer the actions of the attacker by alerts [\[14\]](#). For the detection of multi-stage intrusions

¹ This work was supported by the German Federal Ministry of Education and Research (BMBF) under the grant 01IS08022A.

in alert sequences especially hidden Markov models have been successfully investigated (e.g., [8,10]). However, these models suffer from an implicit modeling of past alerts with the Markov property because in this domain the threat of an alert may highly depend on the context, e.g., the previously recognized alerts. This problem can be addressed by using Conditional Random Fields (CRF) [7] that can consider several (past) alerts to reason about the current state. It has been shown that CRFs are very promising for detecting intrusions from simulated connection information in the KDD cup '99 intrusion domain² compared to decision trees and naive Bayes [5,6].

However, the high amount of reference data as in the KDD data set is only available in simulated environments and is not available in real network domains. The sparse reference data problem is due to the infrequent occurrence of successfully accomplished critical intrusions (cf. [2]) and the lack of annotation. This leads to the problem that most of the possible alerts are even unknown at the training phase of the alert correlator. One possibility to overcome this problem is described in this paper: Typed Linear Chain Conditional Random Fields.

This method uses type information of feature functions for the inference in linear chain conditional random fields and is motivated by filling the gap of missing reference data by considering semantic similarities. Earlier work has already considered the semantic similarity between states for the inference, e.g., in Markov models [1], in hidden Markov models [3], and in input-output hidden Markov models [9]. The latter is similar to linear chain conditional random fields. The inference can also be regarded as mapping a sequence of input values to a sequence of labels.

This paper is organized as follows: In the next section the intrusion detection domain representation in an ontology and its use for preprocessing the alerts from the different IDSs are described. In Section 3 we overcome the problem of sparse reference data by using the domain knowledge described in Section 2. In Section 4 the type extension to linear conditional random fields is evaluated by some real examples in the intrusion detection domain. At last we come to a conclusion and give an outlook of future research.

2 Preprocessing and Domain Knowledge

Hybrid IDSs that use both signature-based and anomaly-based detectors are a promising strategy to improve the precision of intrusion detection. Our approach therefore involves the correlation of alarms from several detectors that can be added if they are present in a particular network. As a first step, we use a syntactic normalization in the IDMEF³ format, which is done by Prelude Manager⁴, a well-known open source interface. This is followed by a semantic normalization that enables the system to handle each sensor's alarms according to their

² KDD '99 data set: <http://kdd.ics.uci.edu/databases/kddcup99>

³ s. RFC 4765.

⁴ <http://www.prelude-technologies.com/>

meaning and a burst filtering that eliminates duplicates of alarms produced by several sensors or as a result of similar observations.

The semantic normalization is based on an ontology in OWL-DL⁵ representation. This ontology contains several facets of the security domain, including e.g., the topology of the network in question, its computers (assets) and general configuration knowledge. Of particular interest for the recognition of multi-step attacks are definitions of possible observations that can be made by the sensors that are organized in a hierarchy of concepts (see Fig. 1). Among the concepts are some that have been derived from classes introduced by Snort⁶. Individuals that belong to these concepts are possible observations and can be imported from Snort’s rules set by an automatic parser. When analysing multi-step attacks, these observations can be considered as describing adversary actions of an attacker, but from a security expert’s perspective. Furthermore, the hierarchy denotes semantic similarity between nearby concepts and thereby supports the further correlation process.

If knowledge about further sensors is added to the ontology, several observations from one or more sensors can be unified when they are instances of the same concept from the observation ontology. E.g., if an observation according to the ET EXPLOIT MS04-007 Kill-Bill ASN1 exploit attempt rule has been made by the Snort IDS and the Prelude logfile parser LML recognizes a match of the Admin login rule in a log file it observes, they may be normalized to one concept `AttemptedAdminObservation` to which they both belong.

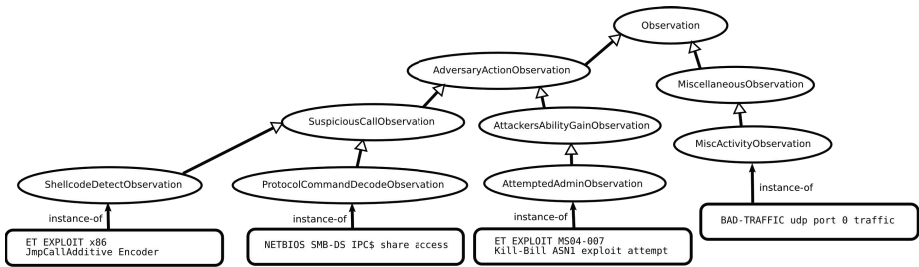


Fig. 1. Excerpt from the observation ontology. Specific observations (as defined by the sensors’ rules) are instances of concepts in a hierarchy.

3 Typed Linear Chain Conditional Random Fields

In this section we briefly introduce conditional random fields and extend them by using a type hierarchy to fill the gap of missing feature functions due to insufficient reference data. For ease of demonstration, this paper assumes that each observation corresponds to one feature function.

⁵ <http://www.w3.org/TR/owl-features/>

⁶ <http://www.snort.org/>

3.1 Prerequisites: Linear Chain Conditional Random Fields

The purpose of linear chain conditional random fields compared to hidden Markov models is to take multiple features (respectively observations) for computing the probability of the labels into account. Thereby they also address the label bias problem from maximum entropy Markov models (cf. [7]). In the following the simplified notation from [13] for linear chain conditional random fields is used with a sequence of labels X and a sequence of observations to be labeled Y with a normalization function Z :

$$p(\mathbf{y}|\mathbf{x}, \lambda) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_j \lambda_j F_j(\mathbf{y}, \mathbf{x})\right) \quad (1)$$

The inference problem is to determine the probability distribution over a vector of labels \mathbf{y} from a vector of observations \mathbf{x} . Conditional random fields are generally not restricted in the dependencies among the nodes, however in linear chain conditional random fields the nodes are only dependent on their predecessor and on the vector of observations. Each feature function F_j has a corresponding weight λ_j that is computed during training.

3.2 Typed Linear Chain Conditional Random Fields

One issue with linear chain conditional random fields is that there is a lack of information for computing the probability of the labels if the features are not known at training time. Our suggestion is to use a type hierarchy of feature functions to find the most similar feature functions that handle the observation. E.g., if no feature function matches a tcp port scan observation, it is dangerous to assume that the tcp port scan observation belongs to a normal system behavior. If there is a feature function matching a udp port scan observation and the type hierarchy expresses a high similarity between udp and tcp port scans, the feature function for udp port scan observation could be assumed to match instead. In our case we can derive the type hierarchy from the semantic normalization (cf. Section 2). The computation of the conditional probability is therefore extended by a parameter for the type hierarchy over feature functions T :

$$p(\mathbf{y}|\mathbf{x}, \lambda, T) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_j \lambda_j F'_j(\mathbf{y}, \mathbf{x})\right) \quad (2)$$

In the case of not having a matching feature function for a given x we propose to instantiate a new feature function F' to match the currently unknown observation, i.e., the feature function is fulfilled (returns 1) iff the given observation arrives.

However, there is the need to determine the corresponding weights for the new feature function. The original weights of the most similar feature functions should be regarded but with a loss to reduce the likelihood that the sequence of observations really belongs to that label. The weights $\lambda_{F'}$ of the new feature function F' are determined by the weights of the most similar feature functions.

The most similar feature functions are given by a similarity measurement. The set of most similar feature functions SF is given by:

$$SF = \{F_s(x, y) | s \in \underset{k}{\operatorname{argmin}} \operatorname{sim}(F_j(x, y), F_k(x, y), T), F_k(x, y) \in B(x, y)\} \quad (3)$$

$B(x, y) \subseteq T$ is the set of bound feature functions, i.e., the feature functions that have a value for the given parameters. The corresponding weights of the new feature function F' based on the most similar feature functions SF is given by:

$$\lambda_{F'} = \frac{1}{|SF|} \sum_s \lambda_s \operatorname{sim}(F_j, F_s, T) \quad (4)$$

As mentioned there is the need for a similarity score between feature functions regarding the type hierarchy, denoted as $\operatorname{sim}(a, b, T)$, $a \in T, b \in T$. There are different possibilities to determine the similarity, e.g., the method of Zhong et al. [15]. This method uses the distance from a to the closest common parent in the type hierarchy denoted as $d(a, ccp, T)$ and the distance from b to the closest common parent $d(b, ccp, T)$ where the distance is defined as:

$$d(a, b, T) = \left| \frac{1}{2^k l(a, T)} - \frac{1}{2^k l(b, T)} \right| \quad (5)$$

$l(n, T)$ is the depth of $n \in T$ from the root node in the corresponding type hierarchy where the depth of the root node is zero ($l(\text{root}, T) = 0$). k is a design parameter to indicate how fast the distance increases depending on the depth in the hierarchy. In this paper $k = 2$ is used as proposed by Zhong. The similarity of two feature functions is given by the distances to the closest common parent by:

$$\operatorname{sim}(a, b, T) = 1 - d(a, ccp, T) - d(b, ccp, T) \in [0; 1] \quad (6)$$

4 Results

The evaluation of typed linear chain conditional random fields is done by two experiments to compare this model to traditional linear chain conditional random fields. In the experiments both models are trained with missing reference data. The first experiment shows how the type knowledge is used to overcome the lack of data. The second experiment is about the dependency of the model to the quality of the type hierarchy. The evaluation data consists of two real intrusions performed with the Metasploit Framework⁷: (1) the *Kill-Bill*⁸ and (2) the *Net-API*⁹ exploit. The gathered sequences of alerts from the Snort detector and the normalized alerts by the preprocessor are presented in table 1 and 2.

⁷ <http://www.metasploit.com/>

⁸ Metasploit: [windows/smb/ms04_007_killbill](#)

⁹ Metasploit: [windows/smb/ms08_067_netapi](#)

Table 1. Alert sequence of the Kill-Bill exploit

Time	Normalized alerts (after preprocessing)	Snort message
1	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
2	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
3	AttemptedAdminObservation	ET EXPLOIT MS04-007 Kill-Bill ASN1 exploit attempt
4	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
5	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic

Table 2. Alert sequence of the Net-API exploit

Time	Normalized alerts (after preprocessing)	Snort rule
1	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
2	ShellcodeDetectObservation	ET EXPLOIT x86 JmpCallAdditive Encoder
3	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
4	ProtocolCommandDecodeObservation	NETBIOS SMB-DS IPC share access
5	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic

Table 3. Similarity of the different feature functions to the feature AttemptedAdminObservation

Alert	Similarity
ProtocolCommandDecodeObservation	0.882812
ShellcodeDetectObservation	0.882812
MiscActivityObservation	0.519531

4.1 Experiment 1

The two kinds of linear chain conditional random fields (typed and untyped) have been trained to detect the `MiscActivityObservation` as normal system behavior and the other observations from the Net-API alert sequence as an attack. Both methods share exactly the same reference data and take the two preceding, the current and the two succeeding alerts for the labeling into account. The linear chain conditional random field has been tested against the typed linear chain conditional random field by performing the untrained Kill-Bill exploit. As expected the typed model detects the Kill-Bill exploit by using the type knowledge for this unknown observation. The typed model does not know the observation `AttemptedAdminObservation` from training but it searches the available feature functions with the highest degree of similarity. These are the alerts `ProtocolCommandDecodeObservation` and `ShellcodeDetectObservation` (cf. Fig. 1 and 3) and it computes the corresponding weights as described in Eqn. 4. Both features refer to an attack and therefore the classification comes to the conclusion that the unknown observation `AttemptedAdminObservation` also refers to an attack. In contrast, the traditional untyped linear chain conditional random field has not detected the Kill-Bill exploit and generated a critical classification in the intrusion detection domain: A false negative. This shows how typed linear chain CRFs enrich traditional linear chain CRFs. In conclusion the typed classification outperforms the traditional classification if a type hierarchy expressing the correct semantic similarities is available.

4.2 Experiment 2

The second experiment shows how a type hierarchy expressing an ambiguous semantic similarity between contradictory feature functions influences the inference process. In this experiment, the `MiscActivityObservation` has been attached to the type hierarchy to have exactly the same similarity than the other similar observations `ProtocolCommandDecodeObservation` and `ShellcodeDetectObservation` (each one having a similarity of 0.882812). The high belief of the model that the `MiscActivityObservation` feature corresponds to normal system behavior and the circumstance that it is as similar as the contradictory feature functions led to the misclassification of the `AttemptedAdminObservation` to a normal system behavior like in linear chain conditional random fields. This behavior results by the contradictory weights associated with the similar observations leading to a nearly uniform probability distribution over the labels. In conclusion the increased inference accuracy of typed linear chain CRFs is highly dependent on a type hierarchy expressing the right semantic similarities. Ambiguous similar feature functions with contradictory semantics lean towards a uniform probability distribution pointing to the appropriate decreased certainty of the results. However, if the type hierarchy expresses the right semantic similarity, the typed model leads to an increased inference accuracy.

5 Conclusion and Future Work

Typed linear chain conditional random fields offer an improved way to handle missing feature functions. The missing feature functions' weights are approximated during runtime by searching semantically similar feature functions out of a type hierarchy. The type hierarchy is extracted out of an ontology and the semantic similarity between the concepts in the ontology (respectively the type hierarchy) are determined by a measurement from Zhong et al. [15]. Fortunately, the training process remains the same as for conditional random fields, only the inference process is adapted. Further, the computational effort of the inference process only increases if missing reference data influences the inference result, all other cases are not affected. First experiments in the domain of intrusion detection have shown that this is a useful extension to linear chain conditional random fields and that with this method variations of already known kinds of intrusions can be detected more reliably. In the future, the evaluation should be extended to a more expressive data set. Currently the benchmark sets of real intrusions are either very limited to the amount/kinds of intrusions or are only available for a low-level analysis. The search for similar features may be improved by suitable search algorithms. Also, the way of similarity measurement might be extended by not only considering a type hierarchy, but also considering different object properties / relations in the ontology, e. g. by considering IP-to-subnet relations or host-to-asset relations. Overall, typed linear chain conditional random fields are a promising step in the direction of using complex domain knowledge to improve reasoning over time with only a few reference data.

References

1. Anderson, C., Domingos, P., Weld, D.: Relational Markov Models and their Application to Adaptive Web Navigation. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002)
2. Anderson, R.: Security Engineering, 2nd edn., p. 664. Wiley Publishing, Chichester (2008)
3. Wagner, T., Elfers, C.: Learning and Prediction based on a Relational Hidden Markov Model. In: International Conference on Agents and Artificial Intelligence (2010)
4. Gu, G., Crdenas, A.A., Lee, W.: Principled Reasoning and Practical Applications of Alert Fusion in Intrusion Detection Systems. In: ASIACCS '08 (2008)
5. Gupta, K.K., Nath, B., Ramamohanarao, K.: Conditional Random Fields for Intrusion Detection. In: 21st International Conference on Advanced Information Networking and Applications Workshops, AINAW'07 (2007)
6. Gupta, K.K., Nath, B., Ramamohanarao, K.: Layered Approach Using Conditional Random Fields for Intrusion Detection. IEEE Transactions on Dependable and Secure Computing (2010)
7. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: 18th International Conf. on Machine Learning (2001)
8. Lee, D., Kim, D., Jung, J.: Multi-Stage Intrusion Detection System Using Hidden Markov Model Algorithm. In: Proceedings of the 2008 International Conference on Information Science and Security (2008)
9. Oblinger, D., Castelli, V., Lau, T., Bergman, L.D.: Similarity-Based Alignment and Generalization. In: Machine Learning: ECML (2005)
10. Ourston, D., Matzner, S., Stump, W., Hopkins, B.: Applications of Hidden Markov Models to Detecting Multi-stage Network Attacks. In: Proceedings of the 36th Hawaii International Conference on System Sciences (2003)
11. Qin, X., Lee, W.: Attack Plan Recognition and Prediction Using Causal Networks. In: Annual Computer Security Applications Conference (2004)
12. Garcia-Teodoro, P., Daz-Verdejo, J., Marci-Fernndez, G., Vzquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers and Security (2009)
13. Wallach, H.M.: Conditional random fields: An introduction. Technical Report MS-CIS-04-21. University of Pennsylvania (2004)
14. Yu, D., Frincke, D.: Improving the quality of alerts and predicting intruder's next goal with Hidden Colored Petri-Net. Computer Networks: The International Journal of Computer and Telecommunications Networking (2007)
15. Zhong, J., Zhu, H., Li, J., Yu, Y.: Conceptual Graph Matching for Semantic Search. In: Proceedings of the 2002 International Conference on Computational Science (2002)

Generalized Derivative Based Kernelized Learning Vector Quantization

Frank-Michael Schleif¹, Thomas Villmann²,
Barbara Hammer¹, Petra Schneider³, and Michael Biehl³

¹ Dept. of Techn., Univ. of Bielefeld,
Universitätsstrasse 21-23, 33615 Bielefeld, Germany
schleif@informatik.uni-leipzig.de, hammer@in.tu-clausthal.de

² Faculty of Math./Natural and CS, Univ. of Appl. Sc. Mittweida,
Technikumplatz 17, 09648 Mittweida, Germany
villmann@hsmw.de

³ Johann Bernoulli Inst. for Math. and CS, Univ. of Groningen,
P.O. Box 407, 9700 AK Groningen, The Netherlands
{m.biehl,p.schneider}@rug.nl

Abstract. We derive a novel derivative based version of kernelized Generalized Learning Vector Quantization (KGLVQ) as an effective, easy to interpret, prototype based and kernelized classifier. It is called D-KGLVQ and we provide generalization error bounds, experimental results on real world data, showing that D-KGLVQ is competitive with KGLVQ and the SVM on UCI data and additionally show that automatic parameter adaptation for the used kernels simplifies the learning.

1 Introduction

Kernelized learning vector quantization (KGLVQ) was proposed in [9] as an extended approach of Generalized Learning Vector Quantization (GLVQ) [10] with the goal to provide non-linear modeling capabilities for learning vector quantizers and to improve the performance in classification tasks. While the approach was quite promising it has been used only rarely due to its calculation complexity. One drawback is the storage of a large kernel matrix and additionally the storage and update of a combinatorial coefficient matrix Ψ . To address this problem multiple authors have proposed alternative strategies to deal with non-linear separable data focusing either on local linear models, the direct integration of alternative diagonal metrics in the cost functions or by full matrix learning strategies [6,2]. Data analysis using kernel methods is a very active field of research (see e.g. [3]) and was very successful in analyzing non-linear problems. The underlying parameters for the kernel are thereby determined in general using cross validation approaches. The obtained models show good generalization behavior but are in general hard to interpret due to the non-linear mapping of the data in a kernel space and the fact that the model parameters are identified based on decision boundary points. Prototype based algorithms provide models which are calculated on typical points of the data space and are hence easily interpretable by experts of the field [11].

In this paper we propose an approach combining the positive aspects of both domains. We extend the GLVQ by a kernelized, differentiable metric called D-KGLVQ which allows the non-linear representation of the data, on the other hand we keep the prototype concept and obtain very compact representation models. Additionally the presented approach allows for an adaptation of kernel parameters during the learning procedure.

In Sec. 2 we present a short introduction into kernels and give the notations used throughout the paper. Subsequently we present the D-KGLVQ algorithm and evaluate it in comparison to standard GLVQ, the kernelized KGLVQ and a state of the art SVM. Finally, we conclude with a discussion in Section 4.

2 Preliminaries

We consider a mapping of a data space X to a potentially high dimensional space $\mathcal{F} : \phi : X \rightarrow \mathcal{F}$. Then, a *kernel function* $\kappa : X \times X \rightarrow \mathbf{R}$. can be characterized by the following necessary and sufficient properties, see [14],

1. κ is either continuous or has a finite domain
2. κ can be computed by decomposition using a certain mapping ϕ

$$\kappa_{\phi}(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle_{\mathcal{F}} \quad (1)$$

From the last equation we have that κ has to be positive semi-definite because of the properties of the inner product. We now assume that the kernel is differentiable with respect to the arguments. Using the linearity in the Hilbert-space \mathcal{F} , dot products of data with the elements of \mathcal{F} generated by X and ϕ can be described as $\mathcal{F}_X = \left\{ \sum_{i=1}^l \alpha_i \kappa(\mathbf{x}_i, \mathbf{y}) : l \in \mathbf{N}, \mathbf{x}_i \in X, \alpha_i \in \mathbf{R} \right\}$. This property is used in [9], adapting the α_i to derive a kernelization of GLVQ.

3 Algorithm

Learning vector quantization was introduced as a generic concept for intuitive prototype-based classification algorithms [8]. Several variants were developed to improve the standard algorithms [7,10,13]. GLVQ is an extension of the standard LVQ providing a cost function [10]. It has the benefit that it can be interpreted as a margin optimization method [5].

All LVQ-algorithms typically constitute distance based approaches. However, as pointed out in [6] more general *similarity measures* can be considered with the remaining restriction of differentiability. Now the idea is to replace such a general similarity measure by inner products which implies the utilization of *kernels*. In this way we obtain a kernel variant of the underlying LVQ algorithms. Focusing on GLVQ extended by a differentiable kernel we obtain the *D-KGLVQ*.

3.1 Standard GLVQ

Let $c_{\mathbf{v}} \in \mathcal{L}$ be the label of input \mathbf{v} , \mathcal{L} a set of labels (classes) with $\#\mathcal{L} = N_{\mathcal{L}}$ and $V \subseteq \mathbf{R}^{D_V}$ be a finite set of inputs \mathbf{v} with $|V| = N$ the number of data points. LVQ

uses a fixed number of prototypes (weight vectors) for each class. Let $\mathbf{W} = \{\mathbf{w}_r\}$ be the set of all prototypes and c_r be the class label of \mathbf{w}_r . Furthermore, let $\mathbf{W}_c = \{\mathbf{w}_r | c_r = c\}$ be the subset of prototypes assigned to class $c \in \mathcal{L}$. Further, let d be an arbitrary (differentiable) distance measure in V . We start with the cost function for GLVQ

$$Cost_{GLVQ} = \sum_{\mathbf{v}} \mu(\mathbf{v}) \quad \mu(\mathbf{v}) = \frac{d_{r_+} - d_{r_-}}{d_{r_+} + d_{r_-}} \quad (2)$$

which has to be minimized by gradient descent. Thereby, d_{r_+} is determined by

$$\mathbf{v} \mapsto \mathbf{s}(\mathbf{v}) = \operatorname{argmin}_{\mathbf{r} \in A} d(\mathbf{v}, \mathbf{w}_r) \quad (3)$$

and $d_{r_+} = d(\mathbf{v}, \mathbf{w}_s)$ with the additional constraint that $c_{\mathbf{v}} = c_r$, i.e. d_{r_+} is the squared distance of the input vector \mathbf{v} to the nearest prototype labeled with $c_{r_+} = c_{\mathbf{v}}$. Analogously, d_{r_-} is defined. Note that $\mu(\mathbf{v})$ is positive if the vector \mathbf{v} is misclassified and negative otherwise.

The learning rule of GLVQ is obtained taking the derivatives of the above cost function. Using $\frac{\partial \mu(\mathbf{v})}{\partial \mathbf{w}_{r_+}} = \xi^+ \frac{\partial d_{r_+}}{\partial \mathbf{w}_{r_+}}$ and $\frac{\partial \mu(\mathbf{v})}{\partial \mathbf{w}_{r_-}} = \xi^- \frac{\partial d_{r_-}}{\partial \mathbf{w}_{r_-}}$ with

$$\xi^+ = \frac{2 \cdot d_{r_-}}{(d_{r_+} + d_{r_-})^2} \quad \xi^- = \frac{-2 \cdot d_{r_+}}{(d_{r_+} + d_{r_-})^2} \quad (4)$$

one obtains for the weight updates [6]:

$$\Delta \mathbf{w}_{r_+} = \epsilon^+ \cdot \xi^+ \cdot \frac{\partial d_{r_+}}{\partial \mathbf{w}_{r_+}} \quad \Delta \mathbf{w}_{r_-} = \epsilon^- \cdot \xi^- \cdot \frac{\partial d_{r_-}}{\partial \mathbf{w}_{r_-}} \quad (5)$$

3.2 Kernelized GLVQ

We now briefly review the main concepts used in Kernelized GLVQ (KGLVQ) as given in [9]. The KGLVQ makes use of the same cost function as GLVQ but with the distance calculations done in the kernel space. Under this setting the prototypes cannot explicitly be expressed as vectors in the feature space due to lack of knowledge about the feature space. Instead in [9] the feature space is modeled as a linear combination of all images $\phi(\mathbf{v})$ of the datapoints $\mathbf{v} \in V$. Thus a prototype vector may be described by some linear combination of the feature space data sample, i.e. $\mathbf{w}_j^F = \sum_{i=1}^N \psi_{j,i} \phi(\mathbf{v}_i)$, where $\psi_k \in \mathbf{R}^{|W| \times N}$ is the combinatorial coefficient vector. The distance in feature space between a sample $\phi(\mathbf{v}_i)$ and the feature space prototype vector \mathbf{w}_k^F can be formulated as:

$$\begin{aligned} d_{i,j}^F &= \|\phi(\mathbf{v}_i) - \mathbf{w}_j^F\|^2 = \|\phi(\mathbf{v}_i) - \sum_{i=1}^N \psi_{j,i} \phi(\mathbf{v}_i)\|^2 \\ &= k(\mathbf{v}_i, \mathbf{v}_j) - 2 \sum_{m=1}^N k(\mathbf{v}_j, \mathbf{v}_m) \cdot \psi_{j,m} + \sum_{s,t}^N k(\mathbf{v}_s, \mathbf{v}_t) \cdot (\psi_{j,s} \psi_{j,t}) \end{aligned}$$

The update rules of GLVQ are modified in [9] accordingly, using the kernelized representation of the distances and prototypes. Subsequently additional derivatives with respect to the ψ parameters are determined in a direct manner. The algorithm performs all calculations in the feature space using the kernel trick and updates the model parameters by means of ψ updates. The final model consists of the pre-calculated kernel matrix and the combinatorial coefficient matrix for the ψ coefficients. The detailed equations are available in [9].

3.3 Inner Product Based GLVQ and Kernel GLVQ

Now we replace the squared distance measure in (2) by a *differentiable* inner product σ defining a norm d_σ . Thus, identifying any subsets by utilization of σ can be done equivalently (in topological sense) by means of the norm d_σ and vice versa. In context of GLVQ this implies that all margin analysis is still valid also for inner product based variants of GLVQ. Further, among all inner products σ those are of particular interest, which are generated by kernels κ_ϕ defined in (1), i.e. $\sigma = \kappa_\phi$. The prototypes are subsequently preimages of its kernel space counterparts. Here, using the *differentiability assumption* for the used kernels provides an alternative easier solution than the one proposed in [9]. Consider the inner product σ based classifier function

$$\mu_\sigma(\mathbf{v}) = \frac{\sigma_{\mathbf{r}_+}^2 - \sigma_{\mathbf{r}_-}^2}{\sigma_{\mathbf{r}_+}^2 + \sigma_{\mathbf{r}_-}^2}$$

which has to be positive if \mathbf{v} is correctly classified, i.e.

$$\mathbf{v} \mapsto \mathbf{s}(\mathbf{v}) = \operatorname{argmax}_{\mathbf{r} \in A} \left[(\sigma(\mathbf{v}, \mathbf{w}_{\mathbf{r}}))^2 \right] \quad (6)$$

and $\sigma_{\mathbf{r}_+}$ as well $\sigma_{\mathbf{r}_-}$ play the same role as $d_{\mathbf{r}_+}$ and $d_{\mathbf{r}_-}$. The cost changes to

$$\operatorname{Cost}_{KGLVQ} = \sum_{\mathbf{v}} \mu_\sigma(\mathbf{v}). \quad (7)$$

we get prototype derivatives as:

$$\frac{\partial \mu_\sigma(\mathbf{v})}{\partial \mathbf{w}_{\mathbf{r}_\pm}} = \xi_\sigma^\pm \frac{\partial \sigma_{\mathbf{r}_\pm}}{\partial \mathbf{w}_{\mathbf{r}_\pm}} \quad \xi_\sigma^+ = \frac{4 \cdot \sigma_{\mathbf{r}_+} \cdot \sigma_{\mathbf{r}_-}^2}{(\sigma_{\mathbf{r}_+}^2 + \sigma_{\mathbf{r}_-}^2)^2} \quad \xi_\sigma^- = -\frac{4 \cdot \sigma_{\mathbf{r}_+}^2 \cdot \sigma_{\mathbf{r}_-}}{(\sigma_{\mathbf{r}_+}^2 + \sigma_{\mathbf{r}_-}^2)^2}$$

The final updates for the gradient ascent are obtained as

$$\Delta \mathbf{w}_{\mathbf{r}_+} = \epsilon^+ \cdot \xi_\sigma^+ \cdot \frac{\partial \sigma_{\mathbf{r}_+}}{\partial \mathbf{w}_{\mathbf{r}_+}} \quad \Delta \mathbf{w}_{\mathbf{r}_-} = \epsilon^- \cdot \xi_\sigma^- \cdot \frac{\partial \sigma_{\mathbf{r}_-}}{\partial \mathbf{w}_{\mathbf{r}_-}} \quad (8)$$

containing the derivatives of the kernel σ . In case of the usual Euclidean inner product $\sigma_\phi(\mathbf{v}, \mathbf{w}_{\mathbf{r}}) = \mathbf{v}^T \cdot \mathbf{w}_{\mathbf{r}}$ with ϕ as identity function, one simply gets $\frac{\partial \sigma_\phi}{\partial \mathbf{w}_{\mathbf{r}}} = \mathbf{v}$. Yet, in case of a kernel based inner product κ_ϕ , the derivative of the inner product can easily be carried out without any explicit knowledge of the underlying function ϕ taking into account the kernel trick property. For example,

if κ_ϕ is the polynomial kernel $\kappa_\phi = \langle \mathbf{v}, \mathbf{w} \rangle^d$ we have $\frac{\partial \kappa_\phi}{\partial \mathbf{w}} = d \cdot \langle \mathbf{v}, \mathbf{w} \rangle^{d-1} \cdot \mathbf{v}$. For the *rbf-kernel*

$$\kappa_\phi(\mathbf{v}, \mathbf{w}, \gamma) = \exp\left(-\frac{(\mathbf{v} - \mathbf{w})^2}{2\gamma^2}\right) \quad (9)$$

one obtains $\frac{\partial \kappa_\phi}{\partial \mathbf{w}} = \frac{1}{\gamma^2} \exp\left(-\frac{(\mathbf{v} - \mathbf{w})^2}{2\gamma^2}\right) (\mathbf{v} - \mathbf{w})$ whereas for the exponential kernel $\kappa_\phi = \exp(\langle \mathbf{v}, \mathbf{w} \rangle)$ this procedure yields $\frac{\partial \kappa_\phi}{\partial \mathbf{w}} = \exp(\langle \mathbf{v}, \mathbf{w} \rangle) \cdot \mathbf{v}$. Further prominent problem specific differentiable kernels are e.g. the Sobolev-Kernel which is well suited for the analysis of functional data or the Tanimoto-kernel in the context of taxonomical data [16], [15]. For further kernel examples we refer to [14].

Generalization error analysis. It has been shown in [5], [12] that generalization bounds for LVQ schemes can be derived based on the notion of the hypothesis margin of the classifier, independent of the input dimensionality of the classifier, rather the margin, i.e. the difference of the distance of points to its closest correct (\mathbf{w}_{r+}) and wrong prototype (\mathbf{w}_{r-}) determine the generalization ability. This fact makes the algorithm particularly suitable for kernelization where the generalization ability is measured in the feature space \mathcal{F} since the nonlinear feature map as well as the kernel are fixed. However, the feature map Φ typically maps to a high (probably infinite) dimensional space such that the generalization ability of classifiers can severely decrease if the generalization ability would depend on the input dimensionality of the classifier. For GLVQ as a large margin approach, a straightforward transfer of the bounds as provided in [5] based on techniques as given in [1] is possible.

Assume a classification into two classes is considered: we refer to the corresponding prototypes by \mathbf{w}_i^S with $S = \pm 1$. Classification takes place by a winner takes all rule, i.e.

$$f : \mathbf{v} \mapsto \operatorname{sgn}\left(\max_{\mathbf{w}_i^+} \{\sigma(\mathbf{v}, \mathbf{w}_i^+)\} - \max_{\mathbf{w}_i^-} \{\sigma(\mathbf{v}, \mathbf{w}_i^-)\}\right) \quad (10)$$

where sgn selects the sign of the term. A trainable D-KGLVQ network corresponds to a function f in this class with N prototypes. We can assume that data \mathbf{v} are limited in size and, thus, also the images $\Phi(\mathbf{v})$ and the possible location of prototype vectors are restricted by a finite size B . We assume that data and their labeling stem from a (partially unknown) probability distribution P . Generalization bounds aim at bound the generalization error $E_P(f) = P(f(\mathbf{v}) \neq c_v)$. An important role will be taken by the margin of a classification. For this purpose, the sign is dropped in (10) leading to the related function M_f . We fix a positive value, the margin, ρ and the associated loss

$$L : \mathbf{R} \rightarrow \mathbf{R}, t \mapsto \begin{cases} 1 & \text{if } t \leq 0 \\ 1 - t/\rho & \text{if } 0 < t \leq \rho \\ 0 & \text{otherwise} \end{cases}$$

Then, a connection of the generalization error and the empirical error on m samples with respect to the loss L can be established with probability $\delta > 0$

$$\hat{E}_m^L(f) = \sum_{i=1}^m L(c_{\mathbf{v}} \cdot M_f(\mathbf{v}))/m \quad (11)$$

simultaneously for all functions f using techniques of [11]:

$$E_P(f) \leq \hat{E}_m^L(f) + \frac{2}{\rho} R_m(M_{\mathcal{F}}) + \sqrt{\frac{\ln(4/\delta)}{2m}}$$

$R_m(M_{\mathcal{F}})$ denotes the so-called Rademacher complexity of the class of functions implemented by D-KLVQ networks with function M_f . The quantity can be upper bounded, using techniques of [12] and structural properties given in [1], by a term

$$\mathcal{O}\left(\frac{N^{2/3}B^3 + \sqrt{\ln(1/\delta)}}{\sqrt{m}}\right)$$

The quantity B depends on the concrete kernel and can be estimated depending on the data distribution. Thus, generalization bounds for D-KGLVQ with arbitrary kernel result which are comparable to generalization bounds for GLVQ.

3.4 Parameter Adaptation for Gaussian Kernels

Kernel width. The width γ of the Gaussian kernel (9) crucially influences the performance of the classifier. Yet, an alternative is to individualize the kernel width $\gamma_{\mathbf{r}}$ for each prototype $\mathbf{w}_{\mathbf{r}}$ and, afterwards treat them as parameters to be learned. As the prototypes itself, this can be done by stochastic gradient ascent on $Cost_{KGLVQ}$ based on $\frac{\partial Cost_{KGLVQ}}{\partial \gamma_{\mathbf{r}}}$. For the localized Gaussian kernel (9):

$$\frac{\partial \mu_{\sigma}(\mathbf{v})}{\partial \gamma_{\mathbf{r}_{\pm}}} = \xi_{\sigma}^{\pm} \cdot \frac{\partial \kappa_{\phi}(\mathbf{v}, \mathbf{w}_{\mathbf{r}_{\pm}}, \gamma_{\mathbf{r}_{\pm}})}{\partial \gamma_{\mathbf{r}_{\pm}}} = \xi_{\sigma}^{\pm} \cdot \frac{\kappa_{\phi}(\mathbf{v}, \mathbf{w}_{\mathbf{r}_{\pm}}, \gamma_{\mathbf{r}_{\pm}})}{\gamma_{\mathbf{r}_{\pm}}^3} \cdot (\mathbf{v} - \mathbf{w}_{\mathbf{r}_{\pm}})^2.$$

Relevance learning. The Gaussian kernel usually takes as ingredients the Euclidean norm of the vector difference, but more special choices like Sobolev-norms for functional data are also possible. Here we look at the scaled Euclidean metric

$$d^{\lambda}(\mathbf{v}, \mathbf{w}) = \sum_i \lambda_i \cdot (v_i - w_i)^2 \quad \sum_i \lambda_i = 1$$

As usual in relevance learning [6], the scaling parameters λ_i can be adapted with respect to the classification task at hand by gradient learning, leading to a gradient ascent but now as $\frac{\partial Cost_{KGLVQ}}{\partial \lambda_i}$. Considering $\frac{\partial \mu_{\sigma}(\mathbf{v})}{\partial \lambda_i}$ we obtain for $\mathbf{w}_{\mathbf{r}_{\pm}}$

$$\frac{\partial \mu_{\sigma}(\mathbf{v})}{\partial \lambda_i} = \xi_{\sigma}^+ \cdot \frac{\partial \kappa_{\phi}(\mathbf{v}, \mathbf{w}_{\mathbf{r}_{\pm}}, \gamma_{\mathbf{r}_{\pm}})}{\partial \lambda_i} = -\xi_{\sigma}^+ \cdot \frac{\kappa_{\phi}(\mathbf{v}, \mathbf{w}_{\mathbf{r}_{\pm}}, \gamma_{\mathbf{r}_{\pm}})}{2\gamma^2} (v_{\mathbf{r}_{\pm}, i} - w_{\mathbf{r}_{\pm}, i})^2$$

We denote this approach as *Kernelized Relevance GLVQ (D-KGRLVQ)* [1].

¹ Extendable to matrix learning [2], giving *Kernelized Matrix GLVQ (D-KGMLVQ)*

4 Experiments

We present a comparison for 5 benchmark datasets 3 derived from the UCI [\[4\]](#) and the other 2 from [\[11\]](#). We analyze the performance of GLVQ, KGLVQ, D-KGLVQ and SVM using Radial Basis Function (RBF) kernels. The σ parameter of the RBF kernel has been optimized using a separate test set, evaluated in a 3-fold cross validation in a range of $\{1e^{-6}, 1e^{-5}, \dots, 1e^6\}$ and fine tuned for D-KGLVQ using the proposed gamma-learning scheme. We observed an improvement of the D-KGLVQ in the generalization of around 1%. For KGLVQ the parameter settings have been found to be very sensitive while for D-KGLVQ and SVM the specific setting of the sigma parameter was quite robust.

GLVQ, D-KGLVQ and KGLVQ have been trained with 1 prototype per class. In comparison to GLVQ the other methods show improved results demonstrated by the mean errors over the datasets see [Table 1](#). We find that the performance of D-KGLVQ and KGLVQ are quite similar, and both are competitive to SVM. The D-KGLVQ allows for metric adaptation like in [\[11\]](#) to e.g. identify rele-

Table 1. Generalization error and model complexity (averaged) for the datasets

	Dim	Dataset size	GLVQ	D-KGLVQ	KGLVQ	SVM
			Error/#PT	Error/#PT	Error/#PT	Error/#SV
Breast Cancer	32	569	26.19/2	08.00/2	07.30/2	02.64/74
Diabetes	8	768	28.26/2	30.00/2	27.00/2	23.32/370
Heart	13	270	25.93/2	17.00/2	18.81/2	15.43/102
Colorectal Cancer	1408	95	23.16/2	16.25/2	17.87/2	11.58/57
Lung Cancer	1408	100	34.00/2	29.00/2	27.50/2	25.00/65
Mean			25.51/2	20.05/2	19.68/2	15.59/134

vant individual features in the original data space. Individual prototypes can be analyzed with respect to their receptive fields. Sets of data points, represented by one prototype in a high-dimensional space can be related back to belong to each other and can be considered to be similar in its characteristics. This is not possible using SVM models because their model parameters are extreme points rather prototypes. Considering the model complexity we find that with only 2 prototypes the LVQ methods perform quite well. Especially for D-KGLVQ a very compact model is obtained whereas for KGLVQ the model contains additionally the kernel matrix and the combinatorial coefficient matrix. For SVM the performance is a bit better than for all other methods but with a large number of support vectors in the model.

5 Conclusions

In this paper we derived a novel kernelized learning vector quantizer employing differentiable kernel functions. We provided generalization error bounds analogous to that used in [\[5\]](#). We presented experimental results on real world datasets which showed that D-KGLVQ is competitive with both KGLVQ and SVM. In

terms of computational complexity the demands of D-KGLVQ during training are significantly lower with respect to KGLVQ because no large combinatorial coefficient matrix has to be calculated and stored. As pointed out in Section 2 the D-KGLVQ can be easily extended to use problem specific parametrized kernel functions. The parameters of the kernel functions can also be subject of the optimization provided by D-KGLVQ. The solutions generated by D-KGLVQ are in the kernel space but the model parameters keep the prototype concept and are therefore compact and easier to interpret than typical kernel models.

References

1. Bartlett, P., Mendelson, S.: Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research* 3, 463–482 (2003)
2. Biehl, M., Hammer, B., Schleif, F.M., Schneider, P., Villmann, T.: Stationarity of matrix relevance learning vector quantization. *Machine Learning Reports* 3, 1–17 (2009) ISSN:1865-3960, http://www.uni-leipzig.de/~compint/mlr/mlr_01_2009.pdf
3. Bishop, C.: *Pattern Recognition and Machine Learning*. Springer, NY (2006)
4. Blake, C., Merz, C.: *UCI repository of machine learning databases*. Irvine, CA: University of California, Department of Information and Computer Science (1998), <http://www.ics.uci.edu/~mlearn/MLRepository.html>
5. Crammer, K., Gilad-Bachrach, R., A.Navot, A.Tishby: Margin analysis of the LVQ algorithm. In: *Proc. NIPS 2002*, pp. 462–469 (2002)
6. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. *Neural Networks* 15(8-9), 1059–1068 (2002)
7. Hammer, B., Villmann, T.: Mathematical aspects of neural networks. In: Verleysen, M. (ed.) *Proc. of European Symposium on Artificial Neural Networks (ESANN'2003)*, d-side, Brussels, Belgium, pp. 59–72 (2003)
8. Kohonen, T.: *Self-Organizing Maps*. Springer Series in Information Sciences, vol. 30. Springer, Heidelberg (1995) (Second Extended Edition 1997)
9. Qin, A.K., Suganthan, P.N.: A novel kernel prototype-based learning algorithm. In: *Proc. of ICPR'04*, pp. 2621–2624 (2004)
10. Sato, A.S., Yamada, K.: Generalized learning vector quantization. In: Tesauro, G., Touretzky, D., Leen, T. (eds.) *Advances in Neural Information Processing Systems*, vol. 7, pp. 423–429. MIT Press, Cambridge (1995)
11. Schleif, F.M., Villmann, T., Kostrzewa, M., Hammer, B., Gammernan, A.: Cancer informatics by prototype-networks in mass spectrometry. *Artificial Intelligence in Medicine* 45, 215–228 (2009)
12. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in learning vector quantization. *Neural Computation* (to appear)
13. Seo, S., Obermayer, K.: Soft learning vector quantization. *Neural Computation* 15, 1589–1604 (2003)
14. Shawe-Taylor, J., Cristianini, N.: *Kernel Methods for Pattern Analysis and Discovery*. Cambridge University Press, Cambridge (2004)
15. Simmteit, S., Schleif, F.M., Villmann, T., Ellsner, T.: Tanimoto metric in tree-ssom for improved representation of mass spectrometry data with an underlying taxonomic structure. In: *Proc. of ICMLA 2009*, pp. 563–567. IEEE Press, Los Alamitos (2009)
16. Villmann, T., Schleif, F.M.: Functional vector quantization by neural maps. In: *Proceedings of Whispers 2009*. p. CD (2009)

Cost Optimization of a Localized Irrigation System Using Genetic Algorithms

Mônica Sakuray Pais¹, Júlio César Ferreira¹, Marconi Batista Teixeira¹,
Keiji Yamanaka², and Gilberto Arantes Carrijo²

¹ Instituto Federal Goiano, Brazil

monicaspais@gmail.com, jc@ifgoiano.edu.br, marconibt@gmail.com

² Universidade Federal de Uberlândia, Brazil

keiji@ufu.br, gilberto@ufu.br

Abstract. The high cost of localized irrigation system inhibits the expansion of its application, even though it is the most efficient type of irrigation on water usage. Water is a natural, finite and chargeable resource. The population growth and the rising of population's income require the increase of food and biomass production. The guarantee of agricultural production through irrigation with the rational use of water is a necessity and the research and development of methods to optimize the cost of the localized irrigation project can ensure the expansion of its use. This paper presents a genetic algorithm (GA-LCLI) to search a less costly localized irrigation project. The results are compared with those presented by a previous work: there is an improvement in the execution runtime and in the cost of the irrigation systems.

Keywords: evolutionary computation; genetic algorithm; optimization; localized irrigation system.

1 Introduction

The population growth and the rising income in populated countries as China and India demand the augmentation of food and biomass production all around the world. It stimulates the research and development of technologies in agricultural productivity. Irrigation is an artificial application of water to the soil and the safest way to guarantee the agricultural production.

The Brazilian Annual Report on Hydrological Resources Conjuncture [1] overviews the situation of water resources in the country: the total irrigated area is 4.5 million hectares, a growth rate of 50% in ten years. Brazil is the 16th place in the world rank which corresponds to 1% of the total world irrigated area (277 million hectares). It has the lowest relation between irrigated area and irri-gable area, and very low rate on number of irrigated hectares per capita (0.018 ha/capita), the lowest in South America. Even though irrigation has the largest share of water use (47%) while urban water supply represents 26%, industries 17%, animal watering 8% and rural supply 2%.

The large amount of water required for irrigation stimulates the research for its rational use in irrigation without compromising productivity. The localized irrigation has been used as the most efficient way of supplying water and nutrients to plants. Although localized irrigation systems require high initial investment, the cost can be minimized by the initial design and other criteria established by the designer, like flow rate and pipe diameters. The search for minimum cost of irrigation system implies in evaluating all pipe sizing combination for the design of the hydraulic network, a search space of about 8 million possibilities [2]. It implies in using optimization models. Genetic Algorithms (GAs) are a heuristic search technique and it has been demonstrated in other works ([3], [4], [5], [6]) that it can be more effective than traditional optimization algorithms.

Based on the solution presented in [2] and [7], this work implements a genetic algorithm for searching the lowest annual cost of a hydraulic network of a localized irrigation system, which is called GA-LCLI (Genetic Algorithm for a Less Costly Localized Irrigation). We aim to improve the performance of the solution shown in [7] and make it a better tool to assist the irrigation system design process.

2 Localized Irrigation Systems

Localized irrigation systems have a high efficiency but they also have a high initial cost that inhibits the adoption of this system by the majority of producers. To promote the use of localized irrigation systems it is necessary to find techniques to minimize costs.

The design of a hydraulic network basically consists in sizing pipe's diameters and lengths. Pipe's diameters and lengths are chosen in a way that the system constraints are obeyed and costs of investment, operation and maintenance are minimal.

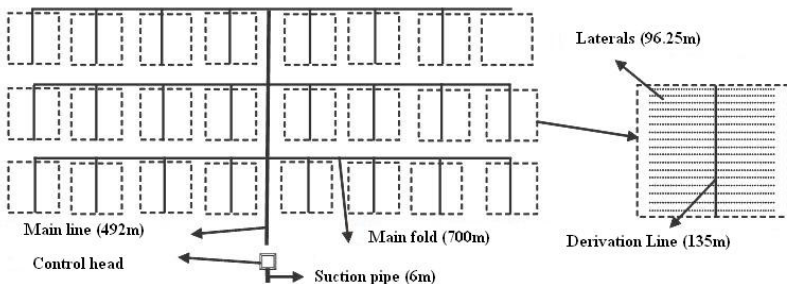


Fig. 1. Irrigation Network Schema with pipe lengths (from [7])

2.1 Related Work on Optimization of Irrigation Systems

Optimization methods can expand the use of irrigation systems because it improves its efficiency and maximizes its profits. Amongst these methods we have linear programming, non-linear programming, dynamic programming, GAs and

others. GAs can help the sizing of hydraulic network pipe's diameters and lengths more efficiently because they evaluate each possible solution against predefined constraints and choose solutions by their feasibility and not only by gradient [3].

Reference [4] presents an application of GAs in water allocation management for irrigation systems. The GAs approach is compared to the quadratic programming approach and results were similar. The advantage of GAs approach is its ability to work with any form of objective function, although the GA showed its sensitiveness to the size of irrigation systems which limits its usage.

A seasonal furrow irrigation model to define the best combination of a weekly irrigation calendar based on economic profit maximization is described in [5]. It shows that GAs approach is better than traditional optimization techniques in establishing a function relating profit, water depth and flow rate.

In reference [6] an irrigated farm model and a GAs optimization method for decision support in irrigation project planning is shown. The project benefits are maximized using GAs, called by the authors as a simple genetic algorithm. Their GA uses a binary representation and a canonical crossover and mutation operators with the following settings: 800 generations, population size of 50, 0.6 crossover rate and 0.02 mutation rate.

2.2 Description of the Irrigation Area Model

We use the same hypothetical area presented by [7] in order to be able to compare results. The area has a length of 1600 meters and has a width of 492 meters. From this is taken an area for traffic and for a riverine forest, given a total of 68.40 hectares for installing the irrigation system.

The hydraulic network of a localized irrigation system consists in: emitters (micro sprinklers or drippers), laterals (to which the micro sprinklers are attached), derivation lines (to which laterals are attached), main fold lines (to which the derivation lines are attached), main lines (to which main fold lines are attached), valves, water meter, filters, control head panel, pump and suction pipe. The number of diameters available for laterals, derivation lines, main fold lines, main lines and suction pipe are, respectively, 2, 4, 4, 1 and 1 diameters.

3 Implementation

The general goal of the localized irrigation system design is to provide irrigation water uniformly and efficiently to a crop, complying with the evapotranspiration needs and to maintain a favorable water balance in the root zone [8]. Our work search for the best pipe sizing for the hydraulic network of a localized irrigation system which gives the lower annual cost per hectare.

The following sections describe how issues related to GA are treated and Table [1] shows a comparison between the best settings for the GA presented by [7] and for the GA-LCLI.

In reference [7], the Matlab Optimization ToolboxTM was used to find the best pipe sizing for the hydraulic network of the localized irrigation system. Our approach is to implement the GA-LCLI coded in Matlab^(R).

Table 1. Comparing GA settings for Marcuzzo's GA [7] and for GA-LCLI

GA parameters	Marcuzzo's GA	GA-LCLI
Predefined Initial Population Size	40	-
Random Initial Population Size	10	50
Population Size	50	50
Maximum Number of Generations	2000	2500
Crossover Probability	0.8	0.8
Mutation Probability	0.01	0.01
Elite size	20	4
Selection Operator	roulette wheel	tournament with 4 players
Crossover Operator	Arithmetic	Arithmetic
Mutation Operator	adaptive feasible	simplified adaptive feasible

3.1 Solution Representation

The best hydraulic network of a localized irrigation system is represented by pipe lengths for each stretch with their respective diameters, and represented by a set of 10 real numbers (same representation used in [7]):

$$[L1 \ L2 \ D1 \ D2 \ D3 \ D4 \ S1 \ S2 \ S3 \ S4]$$

Where L_n = lateral stretches ($n = 1, 2$), D_n = derivation line stretches ($n = 1, 2, 3, 4$), S_n = main fold line stretches ($n = 1, 2, 3, 4$). Each stretch has different pipe diameters.

3.2 Function to Be Optimized

The objective function (from then on it will be called fitness function) is subject to the minimal requirement that the function can map the population into a partially ordered set. In this work, the problem is to find the best sizing of pipe diameter and length for the hydraulic network of a localized irrigation system and the lower annual cost per hectare determines the best solution. Cost calculations consist in summing annual fixed costs (annual irrigation system) and annual variable costs. From [7], the fitness function that represents the total annual cost of the irrigation network per hectare in Brazilian currency¹ (R\$ - Real) is given by:

$$f = \frac{(PPIS \times CRF) + TEEC + TWC}{IA}$$

where

f = fitness function (R\$ $year^{-1} hectare^{-1}$);

$PPIS$ = purchase price of the irrigation system (R\$ $hectare^{-1}$);

$TEEC$ = total electric energy cost (R\$ $year^{-1}$);

TWC = total water cost (R\$ $year^{-1}$);

IA = irrigated area ($hectare$);

¹ The US dollar and Brazilian real exchange rate is 1.7720. June 20, 2010.

CRF = capital recovery factor (*decimal*).

The capital recovery factor is given by:

$$CRF = \frac{J(J+1)^n}{(J+1)^n - 1}$$

where

CRF = capital recovery factor (*decimal*);

J = annual interest rate (*decimal*);

n = useful life of equipment (*year*).

The fitness function evaluation begins by preliminary calculations related to the soil-plant system requirements, and then it performs a sequential calculation on network components: emitters, laterals, derivation lines, main fold, main line, control head, filters, valves, pump and suction pipe.

The hypothetical irrigated area is situated in São Paulo, Brazil, where the rural areas are charged a R\$ 0.1768 per KWh rate. There is a 60% discount between 21h30m and 06h00m (R\$ 0.0707 KWh rate). Since 2001, the Agência Nacional de Águas - ANA (Water National Agency) has implemented actions to charge the use of water resources [9]. The charging works as a water resources management tool and the adopted water rate base is R\$ 0.01 per cubic meter.

3.3 Irrigation System Constraints

Solution feasibility is evaluated against length constraints: lateral stretch lengths between 6 and 96.25 meters, derivation line stretch lengths between 6 and 135 meters, main fold line stretch lengths between 6 and 700 meters.

There are also hydraulic constraints [8]:

1. Maximum flow velocity in pipe stretches: laterals between 1.0 and 1.5 m/s, derivation lines between 1.2 and 2.5 m/s, main fold lines: less than 4.0 m/s.
2. Emitter estimated emission uniformity: more or equal to 90%.
3. Manufacturer's recommendations on emitter pressure range: between 100 and 200 kPa.
4. Maximum variation in pressure drop: laterals with less or equal to 55% of the variation in pressure drop at the operational unit and derivation lines with less or equal to 45% of the variation in pressure drop at the operational unit.

Constraints must be obeyed by solutions. If a solution does not obey any constraint, it is unfeasibly and it is discarded by GA.

3.4 Selection Function

The selection of individuals that represents solutions to produce the successive generations plays an import role in a GA. There are several schemes for the selection processes: roulette wheel, tournament and ranking methods, and elitist models ([10], [11]). This work implements and tests all the previously mentioned selection functions to find the best settings for the GA-LCLI as shown in Table [1].

3.5 Genetic Operators Making Up the Reproduction Function

The two basic types of operator used to produce new solutions based on existing solutions are crossover and mutation. The operators used in this work and in [7] are described next.

Mutation operator. The uniform mutation selects first a fraction of the vector entries of an individual for mutation, where each entry has a probability rate of being mutated. In the second step, the algorithm replaces each selected entry by a random number selected uniformly from the range for that entry.

The adaptive feasible mutation [12] randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. The feasible region is bounded by the constraints and inequality constraints. A step length is chosen along each direction so that linear constraints and bounds are satisfied. The mutation process consists in random generation of a mutation direction vector and an initial step size. Then a mutated individual is generated and in the event this generated mutated individual is located in infeasible region, the step size is adjusted to a smaller value and generates another mutated individual along the chosen mutation direction vector. The previous step is repeated until the generated individual is within the feasible region. In reference [7] the Matlab Optimization ToolboxTM is set to use the adaptative feasible mutation.

The GA-LCLI implements a simplification of the adaptive feasible mutation operator: a uniformly mutated individual is generated. In the event it is not a feasible solution, it is discarded and a new uniformly mutated individual is generated. The process is repeated until a feasible individual is generated or it reaches the maximum number of tries. In this case the process is aborted.

Crossover operator. The arithmetic crossover creates children that are the weighted arithmetic mean of two parents. Children are always feasible with respect to linear constraints and bounds. This operator is used in GA-LCLI and in [7].

3.6 Initial Population

The diversity of the population determines the performance of the GA. In this work, the initial population of 50 individual solutions are randomly generated. If the initial population contains only infeasible solutions, then the random generation process is repeated until a maximum number of times. In [7], 40 initial feasible solutions were provided and 10 were generated randomly.

4 Results

The fitness function for hydraulic analyses in large water distribution network are complex and it affects the running time of a GA: each individual is evaluated by the fitness function and the solution changes slower once it gets closer to the global optimum [3].

We reproduce the [7]'s solution and the same average execution runtime of 32 hours is obtained. After algorithmic choices and implementation details were

made, the GA-LCLI average execution runtime measured for 0% declivity is 1 hour and for 5% declivity is 4 hours, in 3 personal computers, each with a Dual Core processor, 2.2 GHz, 2 GB memory. In reference [7], tests were performed in 18 personal computers configured with Pentium 4 processor, 3 GHz, 1 GB memory to a Dual Core processor, 2.2 GHz, 2 GB memory.

When declivity is not 0%, there is a variation in pressure drop at derivation lines and main line, and the mutation operator might generate unfeasible solutions which cause a computational overhead.

The lowest annual cost obtained by the GA-LCLI is R\$ 1,796.02 per hectare per year and the average is R\$ 1,803.90 when the declivity in the irrigated area is 0%, and the costs of energy and water are respectively R\$ 0.0884 KWh and R\$ 0.01 per cubic meter. In reference [7] for the same declivity, costs of energy and water, the lowest cost was R\$ 1,816.45 per hectare per year.

The solution found by the GA-LCLI that represents the lowest annual cost per hectare obtained is given by:

[48.25 48.00 10.00 67.00 42.00 16.00 104.00 203.00 204.00 189.00]

Which defines a hydraulic network with two lateral stretch lengths of 48.25 and 48.00 meters, four derivation line stretch lengths of 10.00, 67.00, 42.00 and 16.00 meters, four main fold line stretch lengths of 104.00, 203.00, 204.00 and 189.00 meters.

5 Conclusion and Further Research

The localized irrigation works with the maximum efficiency on water usage, a finite, chargeable and natural resource. To stimulate its usage it is necessary to find ways to minimize costs. In this work we presented an implementation of a genetic algorithm (GA-LCLI) for optimization of the pipe sizing definition in a hydraulic network of a localized irrigation system based on the minimization of the annual cost of the system. The results are compared with results presented by [7]: a lower cost and a major improvement on execution runtime are obtained.

Some work remains to be done. The current fitness function only allows one declivity setting value for the irrigated area and it could allow different declivity settings for the irrigated area. A reduction on the initial cost would be gained with a constrain to the pipe size to multiples of 6 (six) or 3 (three) meters, in order to avoid leftovers. Also, experimenting on keeping individual unfeasible solutions and its implications, such as penalty functions, computational overhead, and others, is a work to be done.

References

1. ANA: Brazilian annual report on hydrological resources conjuncture. Technical Report (2009), <http://www.ana.gov.br> (accessed December 10, 2009)
2. Marcuzzo, F.F.N.: Sistema de otimização hidráulica e econômica de rede de irrigação localizada usando algoritmos genéticos. Thesis (Doctoral): São Carlos School of Engineering. University of São Paulo, São Carlos (2008)

3. Walski, T.M., Chase, D.V., Savic, D.A., Grayman, W., Beckwith, S., Koelle, E.: Advanced water distribution modeling and management. Haestad Press, Waterbury (2003)
4. Wardlaw, R., Bhaktikul, K.: Application of a genetic algorithm for water allocation in an irrigation system. *Irrigation and Drainage* 50, 159–170 (2001)
5. Montesinos, P., Camacho, E., Alvarez, S.: Seasonal furrow irrigation model with genetic algorithms (optimec). *Agricultural Water Management* 52(1), 1–16 (2001)
6. Kuo, S.F., Merkley, G.P., Liu, C.W.: Decision support for irrigation project planning using a genetic algorithm. *Agricultural Water Management* 45(3), 243–266 (2000)
7. Marcuzzo, F.F.N., Wendland, E.: Otimização de rede de irrigação de microaspersão usando algoritmos genéticos sob diferentes declividades e tarifação de Água e energia elétrica. *Revista Engenharia na Agricultura. Viçosa-MG* 18(1), 50–62 (2010)
8. ASAE: Standard engineering practices data: Ep 458 - field evaluation of microirrigation systems. American Society of Agricultural Engineers, pp. 792–797. ASAE, St Joseph (1996)
9. ANA: National water resources policy (2009), <http://www.ana.gov.br/ingles/waterPolicy.asp> (accessed December 10, 2009)
10. Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading (1989)
11. Mitchell, M.: An introduction to genetic algorithms. MIT Press, Cambridge (1996)
12. Kumar, R.: System and method for the use of an adaptative mutation operator in genetic algorithms. United States Patent (February 2010) Patent No US 7,660,773 B1

Dimension Reduction for Regression with Bottleneck Neural Networks

Elina Parviainen

BECS, Aalto University School of Science and Technology, Finland

Abstract. Dimension reduction for regression (DRR) deals with the problem of finding for high-dimensional data such low-dimensional representations, which preserve the ability to predict a target variable. We propose doing DRR using a neural network with a low-dimensional "bottleneck" layer. While the network is trained for regression, the bottleneck learns a low-dimensional representation for the data. We compare our method to Covariance Operator Inverse Regression (COIR), which has been reported to perform well compared to many other DRR methods. The bottleneck network compares favorably with COIR: it is applicable to larger data sets, it is less sensitive to tuning parameters and it gives better results on several real data sets.

Keywords: supervised dimension reduction, dimension reduction for regression, neural networks, COIR.

1 Introduction

Dimension reduction tries to find, for a high-dimensional data, a low-dimensional representation which preserves some interesting structure of the data. Supervised variants of dimension reduction methods aim at preserving structure related to predicting a target variable, be it a class label or a regression target. Methods have been developed both in classification and regression contexts.

Dimension reduction for regression seems to evolve as a somewhat isolated branch, sometimes drawing from work on unsupervised and classification-oriented supervised dimension reduction, but receiving little or no attention in works where supervision means supervision by class labels.

Neural networks naturally bridge the gap between classification and regression methods: a multilayer network performing regression turns into a classifier by a simple change in output layer activation function. We have earlier studied bottlenecked classifier networks [1] for dimension reduction. They are trained to form a mapping from inputs to class labels, but on the way the information is forced to flow through a low-dimensional "bottleneck" layer, whose outputs can be used as a low-dimensional representation of the data. In this work, we demonstrate the usefulness of the same idea in regression context.

We compare the bottleneck networks to Covariance Operator Inverse Regression (COIR), [2,3], a nonlinear DRR method based on estimating inverse regression covariance from covariance structures of data and targets. While producing results of same or better quality as COIR, the bottleneck networks provide at

least two benefits. They are fast to train, the time complexity being linear in the number of data points. Methods relying on covariance matrices and eigendecompositions become infeasible when the size of data grows. In [1] the bottleneck networks seemed robust to changes in numbers of hidden units, the only structural parameters needed. COIR, on the other hand, seems fairly sensitive to kernel width parameters.

2 Related Work

Two important groups of DRR methods are exemplified by sliced inverse regression (SIR) [4] and kernel dimension reduction (KDR) [5,6]. Both aim at finding a low-dimensional linear subspace such that projections to that subspace would predict the regression target variable as well as possible.

SIR and other inverse regression based methods rely on the thought that $E[X|Y]$ should lie in the same subspace as $E[Y|X]$, and therefore the former can be used instead of the latter for finding the subspace. This is attractive since target Y is usually lower-dimensional than the data X . Sliced inverse regression divides the range of Y into slices, performs inverse regression on each and combines the results into a weighted covariance matrix, the highest eigenvectors of which give the desired subspace. SIR has given rise to variations and nonlinear extensions [7].

Kernel dimension reduction is more directly based on the requirement that, given the low-dimensional subspace, the data and target should be independent (which means all the information necessary for predicting Y has been captured by the subspace). This is formulated as a minimization problem on conditional covariance operators of Y , in a feature space produced by a suitable kernel. There are also nonlinear extensions of KDR [8,9].

Covariance operator inverse regression [2,3] is a relatively new method which overcomes some of the weaknesses of earlier methods. Slicing in SIR is unreliable for multivariate targets, but is not needed in COIR which uses covariance operators for both data and targets. COIR also avoids the fairly strong assumptions about the distribution of X made by SIR. COIR has a closed form solution, unlike KDR which can suffer from local minima. We will describe COIR in more detail in Sect. 4.

3 Bottleneck Neural Networks for Regression

Bottlenecked regression and classification networks can be thought as supervised versions of autoencoders [10], which have been widely used in dimension reduction. Fig. 1 illustrates these two architectures. The idea is to train the network to perform its main task (reconstruction for autoencoders, classification or regression for supervised networks) but ignore the network output and instead use the results given by a low-dimensional middle layer. When information is forced to flow through such bottleneck, the network will learn a mapping which is able to produce the final results using only a low-dimensional representation of the data.

An autoencoder produces a reconstruction $\tilde{x} \approx x$ when trained with input x and target x . Outputs of the middle layer form the low-dimensional embedding for points x . The network is trained to minimize mean square reconstruction error. The bottleneck network produces an estimate $\tilde{y} \approx y$ when trained with input x and target y (which can be multidimensional). Again, the low-dimensional embedding for points x is read from the outputs of the bottleneck layer. The network minimizes the mean square error between the true target and the network output.

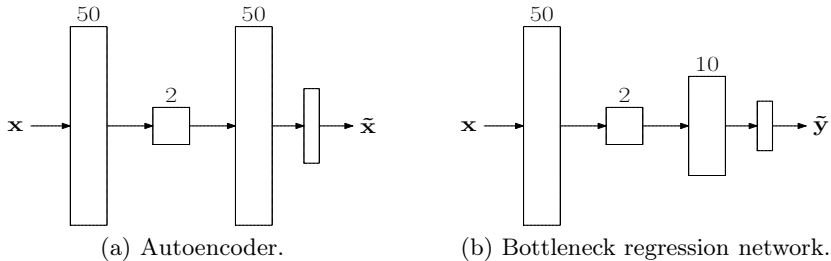


Fig. 1. Illustrations of an autoencoder and a bottleneck regression network. In both networks, the 2D bottleneck layer and the output layer have linear units, and the two remaining hidden layers (in this example, with 50 and 10 hidden units) use sigmoid activations. The autoencoder is trained to reproduce its input x , whereas the bottleneck regression network learns a mapping from x to y .

Feedforward neural networks are almost always trained using gradient descent algorithms, computing the gradients by backpropagating the training error through network to yield gradients w.r.t. the network weights. Gradient descent methods are prone to get stuck in local optima if the objective function is complex. This was the case in our initial experiments as well, and therefore we train the network using simulated annealing [11]. The algorithm starts from a random initial configuration. New solution candidates are generated, and a candidate is accepted as the new configuration with a probability that depends on the temperature. From initial temperature of 1 the system is gradually cooled, always multiplying the old temperature by 0.9. New solutions to try are generated by randomly perturbing all coordinates with addition of normally distributed random variable (with $\sigma = 0.01$). Simulated annealing is better at avoiding local minima than a gradient-based deterministic algorithm, but does not guarantee global optimality with finite running times. We therefore perform 20 runs with different initial configurations and keep the best results.

4 Covariance Operator Inverse Regression

In this section, we briefly introduce the covariance operator inverse regression. This presentation is thought to be extensive enough for implementing the algorithm, but for details and derivations the reader is referred to [2] or [3], from which the description below is abstracted.

Let $\phi_x(\mathbf{u}) = [r_{\sigma_x}(\mathbf{u}, \mathbf{x}_0), r_{\sigma_x}(\mathbf{u}, \mathbf{x}_1), \dots, r_{\sigma_x}(\mathbf{u}, \mathbf{x}_n)]$ denote the representation of point \mathbf{u} (which can be either train or test point) by its similarities to the training data points \mathbf{x}_j , where the similarities are computed using a Gaussian kernel with variance τ

$$r_\tau(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|^2}{2\tau^2}\right),$$

and a similar vector is formed for targets \mathbf{y} , $\phi_y(\mathbf{u}) = [r_{\sigma_y}(\mathbf{u}, \mathbf{y}_0), r_{\sigma_y}(\mathbf{u}, \mathbf{y}_1), \dots, r_{\sigma_y}(\mathbf{u}, \mathbf{y}_n)]$. Using $\phi_x(\mathbf{u})$ we form a feature space representation for the full training data and the labels

$$\Phi_x = [\phi_x(\mathbf{x}_0), \phi_x(\mathbf{x}_1), \dots, \phi_x(\mathbf{x}_n)] , \quad \Phi_y = [\phi_y(\mathbf{y}_0), \phi_y(\mathbf{y}_1), \dots, \phi_y(\mathbf{y}_n)] ,$$

which are used to compute covariance matrices

$$\mathbf{K}_x = \Phi_x^T \Phi_x , \quad \mathbf{K}_y = \Phi_y^T \Phi_y .$$

These are used to compute sample estimate for inverse regression covariance, whose eigenvectors (d highest vectors for a d -dimensional representation) β can be solved from the generalized eigenvalue problem

$$\frac{1}{n} \mathbf{K}_y (\mathbf{K}_y + n\epsilon \mathbf{I}_n)^{-1} \mathbf{K}_x^2 \beta = \lambda \mathbf{K}_x \beta .$$

The leading eigenvectors capture the directions of maximum variance, in the same way as the leading principal components do for a sample covariance matrix. From β and the covariance matrix Φ_x we find the central subspace basis vectors \mathbf{b} and the projections \mathbf{c} of the training points to them

$$\mathbf{b} = \Phi_x \beta , \quad \mathbf{c} = \Phi_x \mathbf{b} .$$

The projections \mathbf{c} are the embedding coordinates for the training points.

When we have the basis vectors available, computing coordinates for the test points means simply computing a feature space representation and projecting it to the basis vectors,

$$\Phi_x^* = [\phi_x(\mathbf{x}_0^*), \phi_x(\mathbf{x}_1^*), \dots, \phi_x(\mathbf{x}_{n^*}^*)] , \quad \mathbf{c}^* = \Phi_x^* \mathbf{b} .$$

5 Experiments and Results

We try both COIR and the bottleneck regression (BR) network on five data sets and compare the results. Details of the data sets are given in Table [II](#), and below we briefly describe the prediction tasks and experiments related to each data.

To make sure that in two-target cases both targets contribute equally to the network error, all target variables are scaled to $[0,1]$. In addition, the data is centered at 0, as assumed in the derivation of COIR formulas in [\[2\]](#).

Two of the data sets are face images, one (Head Pose data) where the task is predicting the horizontal and vertical pose of the head, and one (Yale data)

Table 1. Characteristics of the data sets used in the experiments. “UCI” as the source refers to the UCI Machine Learning Repository [12] <http://archive.ics.uci.edu/ml/> and “ExtYaleB”, the Extended Yale Face Database B, is available from <http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html>. The table lists the dimensions as used in the experiments; in some cases the original data was preprocessed (smaller images, dimensions with missing data dropped etc).

name	targets	data dim	#train	#test	data source	ref.
Head Pose	2	64x64	349	349	isomap.stanford.edu/datasets.html	
Yale small	2	48x64	2419	1728	ExtYaleB	[13]
Parkinson	2	16	4165	1710	UCI	[14]
Crime	1	99	1396	598	UCI	[15]
Concrete	1	8	500	50	UCI	[16]

where the target variable is direction (azimuth and elevation) of illumination. Three other data sets are smaller dimensional. The covariates are different measurements or statistics, and the prediction tasks are two disease symptom scores (Parkinson data), number of violent crimes per population (Crime data) and compressive strength (Concrete data).

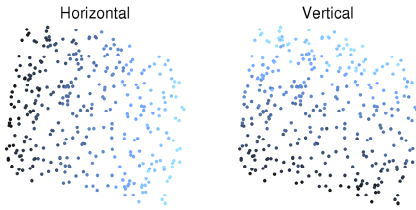
On these data sets, we build two-dimensional visualizations of the data, which are shown in Figs. 2. Table 2, discussed below, will show numerical comparisons.

Parameters needed for the bottleneck network are numbers of hidden units in layers. Size of the bottleneck layer is determined by desired output dimension, 2 in this case. In the first layer, which performs feature extraction on the data, we use 50 units, and the the third, classifier, layer, 10 units are used.

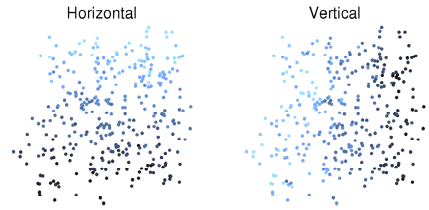
Gaussian kernels are used with COIR. The parameters needed are σ_x and σ_y , the kernel widths for data and target spaces, respectively. In initial experiments the results seemed to be fairly sensitive to the kernel widths. Therefore, several values are tried, and the best result, measured by the NN-regression, is used in the comparisons. The values tried are $0.5 + k$, $k = 0 \dots 14$ for the data space, a large range to allow for different scalings of the data, and $0.1 + 0.1k$, $k = 0 \dots 9$ in the target space, as the targets are known to be scaled.

We measure the quality of results by linear regression (a linear model is fit to low dimensional embedding of the training data, and used to predict the targets for the test data) and nearest neighbor regression (target value for a test data point is taken to be the target at the nearest training data point in the low-dimensional space). Results (RMS errors) are shown in Table 2. In nonlinear (nearest neighbor) prediction, the performance of bottleneck regression is consistently higher. On three of the five data sets, BR gives better linear predictions as well.

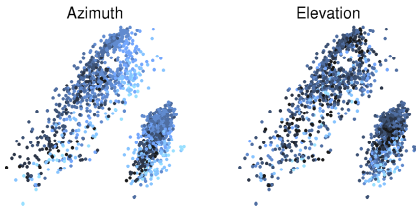
Good performance of bottleneck regression comes as no surprise, given the successful history of autoencoders in dimension reduction. We find it slightly surprising, however, that nonlinear BR results exceed those of COIR on all data sets, as performance of most dimension reduction methods we know of depends on the data at least to some extent. One possible explanation is the parameter



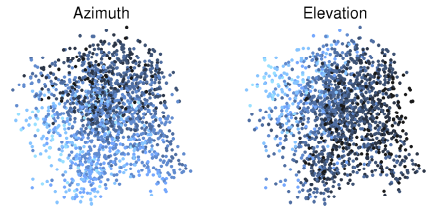
(a) COIR, Head Pose data.



(b) BR, Head Pose data.



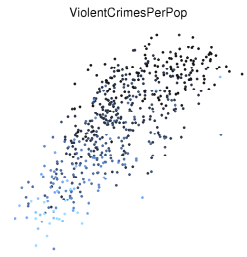
(c) COIR, Yale data.



(d) BR, Yale data.



(e) COIR, Crime data.



(f) BR, Crime data.

Fig. 2. Example visualizations of the test data on some of the data sets. The points are colored according to their true target values.

Table 2. RMS errors for test data. If comparing results to those cited in other works, please remember these errors are for target values which are scaled to $[0, 1]$.

		Head Pose	Parkinson	Crime	Concrete	Yale
lin.regr.	COIR	0.040	0.271	0.255	0.231	1.111
	BR	0.084	0.286	0.145	0.129	0.197
NN-regr.	COIR	0.017	0.163	0.018	0.150	0.119
	BR	0.010	0.141	0.004	0.000	0.013

sensitivity of COIR. We tried to ensure good choices by trying several variance values, but still cannot be sure if COIR was actually using optimal parameters.

6 Conclusions

We introduced a neural network approach to dimension reduction for regression. The bottleneck regression network has a low-dimensional ‘‘bottleneck’’ layer, which learns to form a low-dimensional embedding for the data when the network is trained to perform regression. We compared BR networks to COIR, a recent nonlinear method for finding low-dimensional representations which preserve the ability to predict a target variable. We tried both methods on five real-life data sets, measuring the performance by linear and nearest neighbor regression on the method outputs.

In our experiments bottleneck regression gave consistently better results than COIR.

COIR needs $O(n^2)$ for storing covariance matrices and $O(n^3)$ time for matrix inversions. It was also found sensitive to variance parameters of Gaussian kernels used. We see these factors as the biggest obstacles for successful use of COIR in real applications.

Bottleneck regression does not suffer from these problems, the training time scaling linearly in the number of data points. The biggest drawback of BR networks is that the error function has several local optima. Good local optima can be found using stochastic training methods, but such methods can be rather slow. This is true especially for high-dimensional data sets which lead to networks with large number of parameters. For small data sets, this may cancel out the benefits gotten from linearity in training.

References

1. Parviainen, E.: Deep bottleneck classifiers in supervised dimension reduction. In: Proc. of ICANN 2010 (to appear, 2010)
2. Kim, M., Pavlovic, V.: Dimensionality reduction using covariance operator inverse regression. In: Proc. of CVPR (Computer Vision and Pattern Recognition) (2008)
3. Kim, M., Pavlovic, V.: Covariance operator based dimensionality reduction with extension to semi-supervised learning. In: Proc. of AISTATS. JMLR: W&CP, vol. 5, pp. 280–287 (2009)

4. Li, K.C.: Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association* 86(414), 316–327 (1991)
5. Fukumizu, K., Bach, F.R., Jordan, M.I.: Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research* 5, 73–99 (2004)
6. Fukumizu, K., Bach, F.R., Jordan, M.I.: Kernel dimension reduction in regression. *The Annals of Statistics* 37(1), 1871–1905 (2009)
7. Li, K.C.: High dimensional data analysis via the SIR/PHD approach. Lecture notes in progress (April 2000), <http://www.stat.ucla.edu/~kcli/>
8. Nilsson, J., Sha, F., Jordan, M.I.: Regression on manifolds using kernel dimension reduction. In: *Proc. of ICML* (2007)
9. Moon, K., Pavlović, V.: Regression using Gaussian Process manifold kernel dimensionality reduction. In: *Proc. of MLSP (Machine Learning for Signal Processing)*, pp. 14–19 (2008)
10. DeMers, D., Cottrell, G.: Non-linear dimensionality reduction. In: *Proc. of NIPS*, pp. 580–587 (1993)
11. Schneider, J.J., Kirkpatrick, S.: *Stochastic optimization*. Springer, Heidelberg (2006)
12. Asuncion, A., Newman, D.: *UCI machine learning repository* (2007)
13. Georghiadis, A.S., Belhumeur, P.N., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(6), 643–660 (2001)
14. Tsanas, A., Little, M.A., McSharry, P.E., Ramig, L.O.: Accurate telemonitoring of Parkinson s disease progression by non-invasive speech tests. *IEEE Transactions on Biomedical Engineering* (2009)
15. Redmond, M.A., Baveja, A.: A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research* 141, 660–678 (2002)
16. Yeh, I.C.: Modeling of strength of high performance concrete using artificial neural networks. *Cement and Concrete Research* 28(12), 1797–1808 (1998)

Analysing Satellite Image Time Series by Means of Pattern Mining^{*}

François Petitjean¹, Pierre Gançarski¹,
Florent Masseglia², and Germain Forestier¹

¹ LSIIT (UMR 7005 CNRS/UdS), Bd Sébastien Brant, 67412 Illkirch, France

² INRIA Sophia Antipolis, 2004 route des lucioles, 06902 Sophia Antipolis, France
fpetitjean@unistra.fr

Abstract. Change detection in satellite image time series is an important domain with various applications in land study. Most previous works proposed to perform this detection by studying two images and analysing their differences. However, those methods do not exploit the whole set of images that is available today and they do not propose a description of the detected changes. We propose a sequential pattern mining approach for these image time series with two important features. First, our proposal allows for the analysis of all the images in the series and each image can be considered from multiple points of view. Second, our technique is specifically designed towards image time series where the changes are not the most frequent patterns that can be discovered. Our experiments show the relevance of our approach and the significance of our patterns.

1 Introduction

As remote sensing has witnessed an important technological progress with high definition images, another progress is taking shape with satellites (*e.g.* *Venüs*, *Sentinel-2*) able to acquire image time series at high frequency (two, three images a week and even more). These Satellite Image Time Series (SITS) are an important source of information for scene (*i.e.* geographic area) analysis. A possible but naive usage of these images would consist in selecting two images from the series and study their differences and the evolutions they reveal. However, changes in a scene might spread over a long time period (urbanization, for instance, lasts for several years and building sites do not have the same start time and end time) or they might cycle (such as crop rotation). Consequently, the number of possible combinations is intractable and cannot be reduced to the analysis of two images. We propose to analyse a scene with satellite images on important time periods (our approach will be tested over 35 images and a period of 20 years).

Our approach combines an adequate transform of satellite images and a targeted sequential pattern mining algorithm [12]. This family of algorithms is

^{*} This work was supported by the CNES (French space agency) and by Thales Alenia Space.

typical of knowledge discovery and allows to discover regular or frequent patterns from a set of records. Here a record will be the values of one pixel (*i.e.* its evolution in time). Let us consider, for instance, a set of 24 satellite images of Dubaï, over a period of 2 years (1 image each month). An expected frequent pattern that would be discovered from such a dataset would probably be “15% of all pixels are typical of a desert, then they have the characteristics of a building site and then the characteristics of buildings”. In other words if there exists a large enough set of pixels with the same “behaviour” (*i.e.* these pixels have the same evolution), then this behaviour must be discovered. Let us mention that the pixels’ position is not a criteria here (our goal is not to extract pixels because of a shape). Our goal is to extract significant schemas in the evolution of a set of pixels.

Mining sequential patterns from satellite images makes sense, since pixels having the same evolution will be characterized by the same pattern. Once discovered, these specific schemas of evolution will be given to experts for validation. Examples of such schemas can be found in urbanization (like in Dubaï for instance) or in road creation (where the schema would contain “vegetation” followed by “bare soil” followed by “road”).

This paper is organized as follows. In Sect. 2 we give an overview of existing works in SITS analysis. Section 3 gives the main definitions of sequential patterns and Sect. 4 describes the preprocessing of SITS for the discovery of such patterns. In Sect. 5 we propose a sequential pattern mining technique devoted to SITS and our results are described in Sect. 6. Eventually, we conclude this paper in Sect. 7.

2 Related Works: SITS Analysis

Change detection in a scene allows the analysis, through observations, of land phenomenon with a broad range of applications such as the study of land-cover or even the mapping of damages following a natural disaster. These changes may be of different types, origins and durations.

In the literature, we find three main families of change detection methods. Bi-temporal analysis, *i.e.*, the study of transitions, can locate and study abrupt changes occurring between two observations. Bi-temporal methods include image differencing [3], image ratioing [4] or change vector analysis (CVA) [5]. A second family of mixed methods, mainly statistical, applies to two or more images. They include methods such as post-classification comparison [6], linear data transformation (PCA and MAF) [7], image regression or interpolation [8] and frequency analysis (*e.g.*, Fourier, wavelet) [9]. Eventually, we find methods designed towards image time series and based on radiometric trajectory analysis [10].

Whatever the type of methods used in order to analyse satellite image time series, there is a gap between the amount of data representing these time series, and the ability of algorithms to analyse them. First, these algorithms are often dedicated to the study of a change in a scene from bi-temporal representation. Second, even if they can map change areas they are not able to characterize

them. As for multi-date methods, their results are usually easy to interpret and do not characterize the change.

Meanwhile, frequent sequential pattern mining [12] is intending to extract patterns of evolution in a series of symbols. These methods allow to identify sets of sequences that had the same underlying evolution. Furthermore, they are able to characterize this evolution, by extracting the pattern shared by this set of sequences.

Extracting frequent sequences from SITS was introduced in [11]. The authors study the advantages of such sequences in two applications: weather and agronomics. However, their proposal allows discovering sequences on series of images where the pixels can have only one value.

Our proposal, as explained in Sect. 4, applies to images where the pixels take values on tuples, each value corresponding to a separate band. This characteristics, along with the large number of images, will have important consequences on the patterns, their relevance and the complexity of their discovery.

3 Mining Frequent Sequential Patterns

Sequential patterns are extracted from large sets of records. These records contain sequences of values that belong to a specific set of symbols, as stated by definition 1 (inspired by the definitions of [1]).

Definition 1. Let $\mathcal{I} = \{i_1, i_2, \dots, i_m\}$, be a set of m values (or items). Let $I = \{t_1, t_2, \dots, t_n\}$, be a subset of \mathcal{I} . I is called an itemset. A sequence s is a non-empty list of itemsets noted $\langle s_1, s_2, \dots, s_n \rangle$ where s_j is an itemset. A data sequence is a sequence in the dataset being analysed.

Definition 2 shows the conditions for the inclusion of two sequences. In other words, s_1 is included in s_2 if each itemset of s_1 is included in an itemset of s_2 with the same order. This definition is illustrated by Example 1.

Definition 2. Let $s_1 = \langle a_1, a_2, \dots, a_n \rangle$ and $s_2 = \langle b_1, b_2, \dots, b_m \rangle$ be two sequences. s_1 is included in s_2 ($s_1 \prec s_2$) if and only if $\exists i_1 < i_2 < \dots < i_n$ integers, such that $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \dots, a_n \subseteq b_{i_n}$.

Example 1. The sequence $s1 = \langle (3) (4\ 5) (8) \rangle$ is included in the sequence $s2 = \langle (7) (3\ 8) (9) (4\ 5\ 6)(8) \rangle$ (i.e., $s1 \prec s2$) since $(3) \subseteq (3\ 8)$, $(4\ 5) \subseteq (4\ 5\ 6)$ and $(8) \subseteq (8)$. Meanwhile, the sequence $s3 = \langle (3\ 8\ 9) (4\ 5) \rangle$ is not included in $s2$ since $(3\ 8\ 9)$ is not included in an itemset of $s2$.

In this paper, the main characteristic for sequential pattern extraction will be their frequency. This notion is based on the number of occurrences of a pattern, compared to the total number of sequences, as stated by definition 3. Eventually, for simplicity in the results, only the longest patterns are kept (C.f. definition 4).

Definition 3. A data sequence s_d supports a sequence s (or participates in the support of s) if $s \prec s_d$. Let D be a set of data sequences. The support of s in D

is the fraction of data sequences in D that support s : $\text{support}(s) = |\{s_d \in D / s \prec s_d\}| / |D|$. Let minSupp be the minimum support value, given by the end-user. A sequence having support higher than minSupp is frequent.

Definition 4. Let F^D be the set of frequent sequential patterns in D . In a set of sequences, a sequence s is maximal if s is not contained in any other sequence. Let L^D be the set of maximal sequences of F^D . L^D is the set of maximal frequent sequential patterns in D .

4 Data Preprocessing (SITS)

We want to analyse images from the Kalideos database (the scenes are located in the south-west of France). We have extracted a series of 35 images as illustrated by Fig. 1. These images cover a period of 20 years.

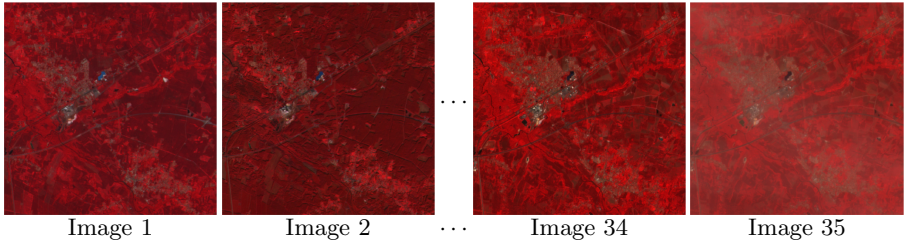


Fig. 1. Extract of the Satellite Image Time Serie of KALIDEOS used. © CNES 2010 – Distribution Spot Image.

Since these images were acquired by different sensors, the comparison of radiometric levels of a pixel (x, y) from one image to another calls for corrections. The value of each pixel has to be adjusted. First, we need to make sure that pixel (x, y) in a series cover the very same geographic localization in every image. Then, some corrections are performed by the CNES in order to reduce the impact of atmospheric changes from one picture to another (since two pictures can be separated by several months).

Once the corrections performed, we are provided with 35 images where each pixel takes values on three bands: Near Infra-Red (NIR), Red (R) and Green (G). To these bands, we add a fourth one, corresponding to the Normalized Difference Vegetation Index (NDVI) calculated as follows for a pixel p :
$$\text{NDVI}(p) = \frac{\text{NIR}(p) - \text{R}(p)}{\text{NIR}(p) + \text{R}(p)}$$

Then, each sequence is built as the series of tuples (NIR,R,G,NDVI) for each pixel (x, y) in the image series.

Eventually, a discretization step is necessary on the bands' values for a sequential pattern extraction. Actually, this step will lower the total number of items during the mining step. Therefore, on each band, we have applied a K-MEANS algorithm [12] in order to obtain 20 clusters of values. For readability,

the cluster numbers have been reordered according to their centroids values. We are thus provided, for each pixel, with a sequence of discrete values as follows:

$$(NIR_1, R_6, G_3, NDVI_{16}) \rightarrow \dots \rightarrow (NIR_{12}, R_3, G_{14}, NDVI_{19}) \quad (1)$$

where $(NIR_1, R_6, G_3, NDVI_{16})$ means that the value of that pixel in the first image is in the first slice of near infra-red, in the 6th slice of red, in the third slice of green and in the 19th slice of $NDVI$.

5 Extracting Sequential Patterns from SITS

The preprocessing steps described in Sect. 4 provide us with a series of images where each pixel is described on a tuple of values. Let us consider the series of 3 images merely reduced to 4 pixels ($p1$ to $p4$) illustrated by Fig. 2. Each pixel in this figure is described on 3 values (corresponding to bands $B1$ to $B3$). With a minimum support of 100 %, there is no frequent pattern in these images (no “behaviour” corresponding to the whole set of pixels). With a minimum support of 50 %, however, we find two frequent behaviours:

1. $\langle (B1, white; B2, white) (B1, grey; B2, red) \rangle$. This behaviour matches the sequences of values of pixels $p2$ on images 1 and 2 (or 3) and $p3$ on images 1 (or 2) and 3.
2. $\langle (B1, white; B2, white) (B1, white; B2, white) \rangle$ (corresponding to $p1$ and $p3$ on images 1 and 2).

Let us note that, in the illustration above, patterns may be frequent even despite a lag in the images that support them.

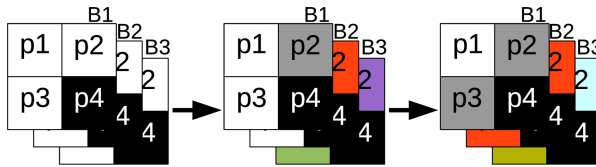


Fig. 2. A series of 3 images, with 4 pixels described on 3 bands

Our goal is to extract sequential patterns, as described above. However, given the characteristics of our data, we find a large number of items (pixel values for one band) with a high support (say, more than 80 %). This has important consequences on the discovery process. First, this will lead to numerous patterns which contain several occurrences of only one frequent item. Such patterns reveal non-evolutions such as $\langle (B1, white; B2, white) (B1, white; B2, white) \rangle$ in our previous illustration and are not really informative. In our images, patterns with high support always correspond to geographic areas that did not change (these areas are majority).

To solve that issue, a naive approach would consist in lowering the minimum support in order to obtain patterns that correspond to changes (since the areas of changes are minority). Actually, specialists on this topic are interested in patterns that correspond to changes. Therefore, we need to extract patterns having lower support (say, between 1 % and 10 %).

However, let us consider $v_i b_j$ the i^{th} value on the j^{th} band. If the support of $v_i b_j$ is larger than 80 % then it is larger than any support below 80 %. Therefore, our extraction process will have to handle every frequent value during the discovery of frequent patterns. Unfortunately, these frequent values will flood the process with an intractable number of candidate and frequent patterns with two important consequences. First, the results are difficult, or even impossible, to obtain (because of the combination curse associated with frequent pattern extraction). Second, if the results can be obtained, they will be difficult to read and very few will be relevant because they will contain a lot of non-evolution patterns.

Therefore, we propose a frequent pattern extraction algorithm that is based on [2] with two important adjustments for SITS:

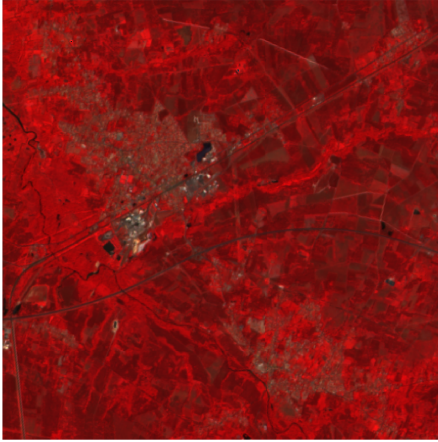
1. During the first step (discovery of frequent items) we only keep the items having threshold between a minimum and a maximum value. To that end, we have added a new support value to the process, which corresponds to a maximum support. Any item having support below the minimum or above the maximum value will be discarded.
2. During the remaining steps, we discard candidates that contain two successive identical values for a band. For instance, the candidate $\langle (B1, white) (B2, white) \rangle$ is authorized, but not the candidate $\langle (B1, white) (B1, white) \rangle$.

6 Experiments

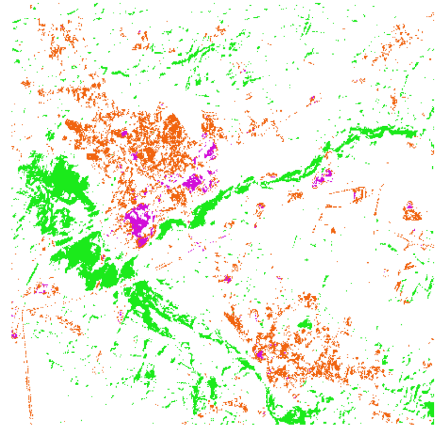
Our images have a definition of 202,500 pixels (450x450). Once preprocessed, (as described in Sect. 4) each pixel takes values on 4 bands and our data contain a total of 28 millions values in the series. By applying the method described in Sect. 5 to the SITS illustrated in Fig. 1, we obtained patterns corresponding to thresholds between 5 % and 50 %. In this section, we report and describe three significant patterns selected from this result.

As we have illustrated in Fig. 2 a frequent pattern is extracted if it corresponds to the behaviour of a given number of pixels. When the pattern is found, we can retrieve the pixels whose series of values contain the pattern. These pixels may than be visualised (highlighted) as illustrated by Fig. 3(b). In this figure, each colour corresponds to a pattern selected from our SITS. Here is the geographic explanation of these patterns:

1. Pattern $\langle (IR, 1) (NDVI, 20) \rangle$ is represented by the green dots in Fig. 3(b). It corresponds to swamps (wetlands) in the SITS. During winter, swamps are almost covered with water, resulting in a low infra-red level (slice



(a) June 3 2006: image selected from the SITS



(b) Illustration of three selected patterns (one color per pattern)

Fig. 3. A sample of our results

- 1) since water does not reflect light a lot. During summer, these swamps are not covered with water any more and light is reflected by the vegetation. Due to its high chlorophyll concentration (due to high irrigation), vegetation in summer has a very high level in NDVI (slice 20).
2. The orange dots represent pattern $\langle (R, 17) (R, 18; NDVI, 3) \rangle$. It corresponds to urban areas that get denser (the number of residences has grew). Actually, urban areas (residences) have a high response in the red band. The level at the beginning of the pattern (slice 17) is highly likely to be the sign of a urban area. The following level (slice 18) shows a urban densification (slices 17 and 18 are separated by a radiometric increase of nearly 25 %), confirmed by a low level of NDVI (corresponding to almost no vegetation).
3. Pattern $\langle (NDVI, 2) (G, 20) (NDVI, 1) \rangle$ is represented by the purple dots. This pattern corresponds to a densification of industrial areas (*e.g.* increase in the number of warehouses). In fact, industrial areas have high response in the green band and show very low values of NDVI. Furthermore, the decrease of NDVI (nearly 30 % from slice 2 to slice 1) shows that vegetation almost disappeared from these areas. Eventually, the maximum level of green is typical of flat roofs (*e.g.* corrugated iron) of industrial areas.

7 Conclusion

Our pattern extraction principle allowed us to find a significant number of relevant patterns such as the sample described in Sect. 6. Our patterns all have a geographic meaning. They correspond either to cyclic behaviours (swamps) or to long term evolutions through the dataset (densifications). Our technique is

designed towards this specific extraction with a data mining process that takes into account a maximum support in the extraction. Indeed, when the support of a value is too high, it might lead to non-evolution patterns and numerous combinations. Thanks to our principle, this drawback is avoided and the discovered patterns are easy to read and understand.

References

1. Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: Proceedings of the 11th International Conference on Data Engineering (ICDE'95), pp. 3–14 (1995)
2. Masseglia, F., Cathala, F., Poncet, P.: The PSP Approach for Mining Sequential Patterns. In: Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (1998)
3. Bruzzone, L., Prieto, D.: Automatic analysis of the difference image for unsupervised change detection. *IEEE Transactions on Geoscience and Remote Sensing* 38(3), 1171–1182 (2000)
4. Todd, W.: Urban and regional land use change detected by using Landsat data. *Journal of Research by the US Geological Survey* 5, 527–534 (1977)
5. Johnson, R., Kasischke, E.: Change vector analysis: a technique for the multi-spectral monitoring of land cover and condition. *International Journal of Remote Sensing* 19(16), 411–426 (1998)
6. Foody, G.: Monitoring the magnitude of land-cover change around the southern limits of the Sahara. *Photogrammetric Engineering and Remote Sensing* 67(7), 841–848 (2001)
7. Nielsen, A., Conradsen, K., Simpson, J.: Multivariate Alteration Detection (MAD) and MAF Postprocessing in Multispectral, Bitemporal Image Data: New Approaches to Change Detection Studies. *Remote Sensing of Environment* 64(1), 1–19 (1998)
8. Jha, C., Unni, N.: Digital change detection of forest conversion of a dry tropical Indian forest region. *International Journal of Remote Sensing* 15(13), 2543–2552 (1994)
9. Andres, L., Salas, W., Skole, D.: Fourier analysis of multi-temporal AVHRR data applied to a land cover classification. *International Journal of Remote Sensing* 15(5), 1115–1121 (1994)
10. Kennedy, R.E., Cohen, W.B., Schroeder, T.A.: Trajectory-based change detection for automated characterization of forest disturbance dynamics. *Remote Sensing of Environment* 110(3), 370–386 (2007)
11. Julea, A., Méger, N., Trouvé, E., Bolon, P.: On extracting evolutions from satellite image time series. In: *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 5, pp. 228–231 (2008)
12. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297 (1967)

Sentences Generation by Frequent Parsing Patterns

Takashi Yanagisawa¹, Takao Miura¹, and Isamu Shioya²

¹ HOSEI University, Dept.of Elect.& Elect. Engr.
3-7-2 KajinoCho, Koganei, Tokyo, 184-8584 Japan

² SANNO University, Dept.of Informatics
Kamikasuya 1672, Isehara, Kanagawa, Japan

Abstract. We propose a sophisticated approach to generate sentences from syntax trees. Users are assumed to give their intent in text or equivalent ones (such as syntax trees). Here we generate standard sentences by examining how the syntax structure consist of frequent structures and how they are constructed. We examine corpus in some domains to extract elementary syntax structures appeared in the corpus as well as standard sentences using the trees.

Keywords: Parsing, Generating Sentences, Elementary Syntax Structures.

1 Introduction

Recently there have been several approaches to improve natural language sentences computationally proposed so far. Among other, we see sophisticated techniques about paraphrasing, summarization and any other research for restating sentences. Generally it is not easy to interpret complicated structure of sentences, and there are some investigation for replacing synonyms, removing redundancies and splitting long sentences.

For example, *decomposition*, may provide us with better readability. Kanda et al. proposes how to decompose long sentences into shorter ones [2]. They examine adverbial clauses based on frequent decomposable patterns and morphological analysis [1], then decompose the sentences by making the phrases nominative. However, they depend on individual word attributes but no general rule. To extract semantics from sentences in formal languages, there have been investigated for long time by analysing structural aspect using Chomsky grammar. Under Chomsky assumption, all the parts of grammar constitute complete sentences functionally and each part plays some role to describe users' intent. Thus we analyze sentences (syntax) to examine whether they follow grammatical rules or not.

One of the problems is that, given semantics (in terms of the trees), we may have many ways to generate sentences in natural languages. We should avoid ambiguity and personality about the sentences but we like standardized, simple and non-ambiguous sentences.

In this investigation, we assume typical situation such as local announcement where we face to *standard* sentences but many changes in detail or sometimes

big difference to describe sudden affairs. People are disgusted with daily typical contents for the announcement because of many but predictable changes. On the other hand, once they listen to announcement they never heard, they feel uneasy because the people wonder whether they can see the contents correctly and precisely. There should be some mechanism by which people can go well with the announcement without any confusion and ambiguity.

Given some intents in a form of parsing trees, we like to generate sentences automatically. It seems better to be able to construct sentences from *typical* tree patterns. However, it is not clear whether we can generate *natural* sentences in such a way to easily understand what's going on. Since we target for obtaining simple, precise and clear sentences, we'd better avoid long or complicated ones. Here we propose some rules to transform parsing trees into elementary ones for this purpose.

In summary, we examine some issues from the 4 points of views:

- (1) How to specify parsing trees without any ambiguity to describe users' intent correctly and precisely.
- (2) How to transform a tree into a collection of elementary tree patterns and to relate them.
- (3) How to generate sentences from a collection of elementary parsing trees.
- (4) How to obtain elementary tree patterns.

Here we discuss mainly (2),(3) and (4). Also we examine only tree structures but not words although it is important to restate words into easier ones. Generally, to understand sentences easily and quickly, we'd better avoid complicated structures such as deeply nested sentences and sentences containing a variety of morphemes. When transforming tree structures, we might improve these situation by tree decomposition (into smaller and simpler ones). But we should think about what, when, and how to do that. Our basic approach comes from corpus-based approaches, i.e., extracting frequent tree-patterns.

In section 2 and 3, we describe morphological analysis to Japanese, and define sentence generation and general framework while, in section 4, we discuss how to tackle the issues. In section 5, we show some experimental results and we conclude our discussion in section 6.

2 Morphological Analysis

Documents consist of mainly texts, figures and tables, texts contain many sentences which are sequences of words. A *word* means a character string separated by space or punctuation, but the problem is not really simple: how can we think about *compound words* such as "U.S.A.", *idioms* (a group of words carrying a different meaning when used together) or *collocation* (the way that some words occur regularly when other words are used)

A sentence in natural languages consists of *morphemes*. A morpheme is a unit of strings carrying minimal meaning. Each sentence can be decomposed into a sequence of morphemes in a form of *token* (single word), *inflection* (stemming) and *part-of-speech* (POS) as noun and verb. The process is one of the primitive

step, called *morphological analysis*. It plays important roles on syntax and semantic analysis and context analysis. In this work, by morpheme, we mean a pair of token and part-of-speech attributes.

We know the fact that, in English for example, a word describes grammatical roles such as *case* and *plurality* by means of word order or inflection. The difference between “John calls Mary” and “Mary calls John” corresponds to the two interpretation of *who calls whom* over John and Mary. Such kind of language is called *inflectional language*. On the other hand, in some languages as Japanese and Chinese, grammatical relationship can be described by means of postpositional particles, and such kind of languages is called *agglutinative language*. For example, “I”, “My” and “Me” correspond to “*watashi wa*” (*I*), “*watashi no*” (*my*) and “*watashi wo*” (*me*) respectively where “*watashi*” means the first personal pronoun. The differences are 3 postpositional particles “*wa*”, “*no*” and “*wo*” which define subjective, possessive and objective respectively. As for John and Mary, let us see the two sentences:

"ジョンが(john ga) メアリーを(mary wo) 呼ぶ(yobu)" (*John calls Mary*)

"ジョンを(john wo) メアリーが(mary ga) 呼ぶ(yobu)" (*Mary calls John*)

In the two sentences note the positions of “*john*” (*John*), “*mary*” (*Mary*) and “*yobu*” (*call*). They are exactly same but the difference of postpositional particles.

Another aspect is *word order*. Because of cases, basically we can put any words to any places, as we shown in the above example. Only one exception is a *predicate* part which appear as the last. In any documents in Japanese, the predicate appears as a *last verb* in each sentence. We apply *morphological analysis* and extract the predicate parts easily. If there is no verb around the last, we could estimate the one. In this investigation, we propose an experimental and efficient approach of sentence generation based on Japanese characteristics. We apply our theory to Japanese and examine relationship between morphemes.

3 Generating Sentences from Trees

In this investigation we assume each sentence has no ambiguity within and we must have a unique parsing tree. A *well-structured* parsing tree means a tree which contains completely specified description of users’ intent. Once we obtain such a tree, it is not hard to preserve original semantics during grammatical transformation. Generally the longer sentences we write, the more information we talk about. However, they cause difficulty against users: the users should memorize much more information. If we decompose them into sequences of semantic units, each sentence may become shorter and we could get the intent of the sentence easier while we should understand more relationship among the sentences.

In this investigation, we discuss a sophisticated approach by which users are able to grasp sentences easily. Sentences contain several internal information such as morphemes and dependency structure among them. By using *frequent parsing patterns*, users can capture their roles in an integrated manner such as relationship of subjects and predicates, qualifications and conjunction. Practically there is few sentence consisting of one simple structure, but very often we

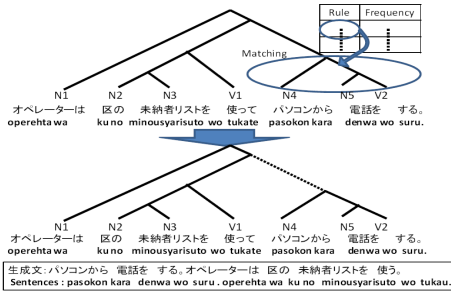


Fig. 1. Decomposing Sentences

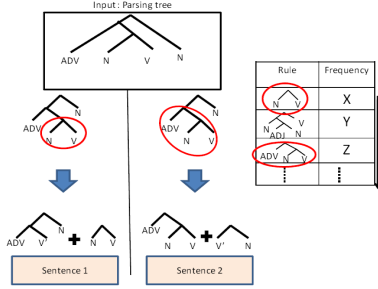


Fig. 2. Decomposing Parsing Trees

see nested structures of sentences inside. We like to decompose them into frequent parts and generate simpler sentences based on the parts. Here we consider frequent patterns as rules.

Let us illustrate how to decompose sentences in a figure 1. In the figure we assume a sentence and parse this sentence:

オペレーターは(operehta wa) 区の(ku no) 未納者リストを(minohsha risutowo) 使って(tsukatte) パソコンから(pasokon kara) 電話を(denwa wo) する(suru).
 (an operator calls by PC using a list of tax defaulters in a ward.)

パソコンから(pasokon kara) 電話を(denwa wo) する(suru). オペレーターは(operehta wa) 区の(ku no) 未納者リストを(minohsha risuto wo) 使う(tsukau).
 (Calling by PC, an operator uses a list of tax defaulters in a ward.)

In this sentence, “operehta” (an operator) has two actions, “tsukau” (use) and “denwa suru” (call). Decomposing the sentence, we have two sentence each of which contains these verbs respectively. Because we have simple sentences, it seems easier to understand intent of each sentence, but there arises a new relationship between the two. This means we could apply our idea to practical application under limited situation and available patterns.

As shown in the figure 1, simple sentences have shallow parsing trees so that we put our attention on how we decompose parsing trees.

4 Decomposing Trees

4.1 Decomposition Algorithm

In this section, let us outline our algorithms to decompose parsing trees. Here we assume a collection of decomposition conditions (in terms of partial parsing trees) in advance. Given a sentence (and its parsing tree), we apply morphological analysis in advance and add the result to each word. We begin with examining the tree in a post order to see whether any part of the tree satisfies decomposition conditions or not, as shown in a figure 2. That is, for each condition, we examine whether any part satisfies it or not. Once we see the part satisfies the condition, we extract the part and construct a new tree, which is a partial parsing tree. We mark the point by a circle in the figure 1. The new tree contains all the leaf

information with some relation to the original tree, while the remaining part is replaced with a pronoun or an auxiliary verb. We repeat the process until no more condition can be applied.

In our running example, we have the partial parsing tree:

オペレーターは(operehta wa) 区の(ku no) 未納者リストを(minohsha risutowo) 使って
(tsukatte)

(an operator uses a list of tax defaulters in a ward.)

Note we replace the final verb “tsukatte” by the basic form “tsukau” (*use*) to make a sentence. We visit the tree in post order and generate sentences in this order. Eventually we generate a sentence:

パソコンから(pasokon kara) 電話を(denwa wo) する(suru). オペレーターは(operehta wa) 区の(ku no) 未納者リストを(minohsha risuto wo) 使う(tsukau).

(Calling by PC, an operator uses a list of tax defaulters in a ward.)

Let us describe our algorithm. Given a tree T with the depth D , we select all the decomposition conditions whose depth are D or lower. For each condition, we examine whether we can decompose the tree or not. Whenever some part satisfies the condition, we separate the partial tree. Finally we obtain several partial parsing tree and the one remaining. Then we generate sentences using all of them.

4.2 Frequent Parsing Patterns

Sentences contain several internal information such as morphemes and dependency structure among them. To obtain rules for extraction, we analyze all the sentences of our corpus in advance by using *KNP*(Kurohashi Nagao Parse) which is a rule-based dependency analyzer to Japanese [4]. First we analyze each (complete) sentence and build the parsing tree as well as morphemes and dependency structure, then we count frequencies of the days (i.e., how many days each structure appears). If a parsing tree appears more than α times, it is called a *frequent parsing pattern*. Here we put priority on frequency because we examine *news corpus* and the corpus contains well-structured sentences very often. We also examine depth of trees to see the complexity.

In our case, we examine parsing-results such as *KNP*'s. To each word, we analyze dependency relation and put *attachment*, *morpheme* to each word where *attachment* means “to which word does this word have the relation”, denoted by non-negative integer corresponding to word position and “-1” means the last. Let us illustrate our situation. We note that N means noun, V verb, S suffix, PP postpositional particle and SP special. The first line 4 : $\{N\}\{S\}\{PP\}\{PP\}$, for instance, says that this has the relation to the 4-th line -1 : $\{V\}\{SP\}$. We examine all the leaf information as well as its structure to see matching.

5 Experimental Results

5.1 Preliminaries

In this experiment, we examine *Asahi news corpus* Jan. 01 to Jan. 11, 2007. Then we get 3268 articles, 67257 sentences and 1332819 words in total from the base data.

Table 1. Corpus Data (Jan.01-11, 2007)

Date	Articles	Sentences	Words
Jan 1	160	8336	142822
Jan 3	220	4455	85485
Jan 4	244	5276	102388
Jan 5	377	7756	153687
Jan 6	452	8976	178732
Jan 7	279	5554	110649
Jan 8	223	4727	89216
Jan 9	313	5612	117691
Jan 10	519	8316	176263
Jan 11	481	8249	175886
Total	3268	67257	1332819

No.	Sentence
1	声を (koe wo) 聞いた (kiita) 多くの (ooku no) 人が (hito ga), 選出方法に (senshutsu houhou ni) 疑問を (gimon wo) 感じて (kanjite) いた (ita). (Many people who heard the opinion wondered about the way of the selection.)
4	同会議は (doukaigi wa) 行政に (gyousei ni) 加えて (kuwaste), 林業関係者や (ringyokankeisha ya) 経済, 教育, 学識経験者なども (keizai, kyoiku, gakushikikeikensha nadomo) 交えた (majieta) メンバーで (menba de) 構成 (kousei). (The committee consists of staffs in forestry, economy, education and experience or academic standing as well as governmental staffs.)
5	もしやと (moshiya to) 思い (omoi), 最近に (saikin ni) なって (natte) 恐る恐る (osoru osoru) 同クリニックで (doukurinikku de) 検診を (kenshin wo) 受けた (uketa). (Wondering the possibility, recently I had medical check up at the clinic timidly.)
9	医師会や (ishikai ya) 薬剤師会などの (yakuzaishi-kai nadono) 支援を (shien wo) 背景に (haikeni), 県内の (ken-nai no) 衆院議員 17 人が (shuin-giin 17 nin ga) 持ち回りで (mochimawari de) 集会を (shukai wo) 開き (hiraki), 支持を (shiji wo) 呼びかけて (yobikakete) きた (kita). (17 prefectural members of the House of Representatives held handed-round meetings and asked strong support with the help of the doctor association and the pharmaceutical association.)
16	史子は (ayako wa) 二の腕を (ninoude wo) かき抱き (kaki-idaki), 浮かびかけた (ukabikaketa) 妙な (myo na) 妄想を (mousou wo) 払い落とすために (haraiotosu tameni), ぶるりと (bururi to) 首を (kubi wo) 振った (futta). (Ayako seized him by the upper arm and shook her head to avoid odd fancy that crossed her mind.)

Fig. 3. Test sentence

We select 20 sentences randomly from the corpus (Jan. 12- Dec.31, 2007) shown in a table 3 for the purpose of test generation.

To evaluate generated sentences, we introduce 3 criteria:

- 1) How well we can see the intent of generated sentences ?
- 2) How well we can see the intent of an original sentence ?
- 3) How many parsing patterns we can apply ?

To see 1) and 2), we evaluate the generated sentences by A (easy to understand), B (medium) and C (hard to understand). To see 3), we define two kinds of parameters. First, to compare an original sentence d_x with generated ones d_y , we introduce *similarity* $Sim(d_x, d_y)$ based on morphemes defined as follows: $Sim(d_x, d_y) = \frac{\sum x_i \times y_i}{\sqrt{(\sum x_i^2) \times (\sum y_i^2)}}$ Generally, given two sentences that have similar contents with each other, we expect some of words contain similar morphemes. Thus we count frequency to each morpheme. Let x_i, y_i be the counts of i -th

Symbol	Morpheme	Input
ADV1	{ AD } { PP }	もしやと (moshiya to) (<i>possibly</i>)
V1	{ V } { SP }	思い (omoi) (<i>I wonder</i>)
N1	{ N } { PP }	最近に (saikin ni) (<i>recently</i>)
V2	{ V }	なって (natte) <i>it happens</i>
ADV3	{ AV }	恐る恐る (osoruosoru) (<i>timidly</i>)
N3	{ PF } { N } { PP }	同クリニックで (dou kurinikku de) (<i>at the clinic</i>)
N4	{ N } { PP }	検診を (kenshin wo) (<i>medical checkup</i>)
V3	{ V } { SP }	受けた (uketa) (<i>I had</i>)

Fig. 4. Original Sentence

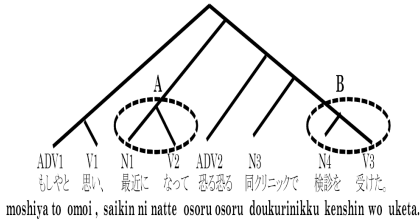


Fig. 5. Input Parsing Tree

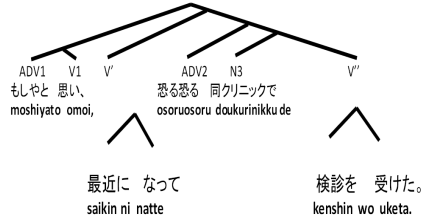


Fig. 6. Output Parsing Tree

Table 2. Frequent Patterns

Frequency	Depth d							
	1	2	3	4	5	6	7	8
2	15	88	103	41	19	7	2	1
3	12	37	21	9	1	0	0	1
4	7	14	11	1	0	0	0	0
5	1	11	3	0	0	0	0	0
6	3	3	5	1	0	0	0	0
7	2	3	0	0	0	0	0	0
8	2	3	0	0	0	0	0	0
9	6	1	1	0	0	0	0	0
10	4	3	0	0	0	0	0	0

morpheme in two sentences d_x and d_y respectively. Then we obtain the similarity $Sim(d_x, d_y)$ or Sim of the two sentences. Here we examine 11 kinds of morphemes: noun (N), postpositional particle (PP), verb (V), suffix (S), prefix (PF), special (SP), adjective (A), demonstrative pronoun (DP), adverb (AD), auxiliary verb (AV) and copula (C) where “special” means comma, period, sign or any other functional characters.

Another parameter shows how many words appear in both sentences. The parameter $Coincidence(d_x, d_y)$ or $Coin$ is defined as follows: $Coincidence(d_x, d_y) = \frac{W_{xy}}{W_x}$. Let us note that W_x means the number of word occurrences in an original sentence d_x , and W_{xy} means the number of word occurrences of generated sentences d_y which appear in the original sentence d_x .

Number	Generated Sentence	Score
1	声を (koe wo) 聞く (kiku). 疑問を (gimon wo) 感じて (kanjite) いた (ita). 多くの (ooku no) 人が (hito ga), 選出方法に (senshutsu houhou ni) [感じて (kanjite) いた (ita)] (People heard the opinion. They wondered. Many people [wondered about] the way of the selection.)	A
4	行政に (gyosei ni) 加える (kuwaeru). 同会議は (doukaigi wa) 加える (kuwaeru). 林業関係者や (ringyo kankeisha ya), 経済 (keizai), 教育 (kyoiku), 学識経験者なども (gakushiki keikensa nadomo) 交えた (majieta) メンバーで (menba de) 構成 (kousei). (Including governmental staffs, the committee contains [some], consisting of staffs in forestry, economy, education and experience or academic standing.)	B
5	最近に (saikin ni) なる (naru). 検診を (kenshin wo) 受ける (ukeru). もしやと (moshiya to) 思い (omoi), 恐る恐る (osoru osoru) 同クリニックで (doukurinikku de) [受ける (ukeru)]. (It was recent. I had medical check up. Wondering the possibility, I [did it] at the clinic timidly.)	A
9	集会を (shukai wo) 聞く (hiraku). 支持を (shiji wo) 呼びかけて (yobikakete) きた (kita). 医師会や (ishikai ya) 薬剤師会などの (yakuzaishi-kai nadono) 支援を (shien wo) 背景に (haikeni), 県内の (ken-nai no) 衆院議員 (shuin-giin) 17人が (17 nin ga) 持ち回りで (mochimawari de) [呼びかけて (yobikakete) きた (kita)] (They hold a meeting. They asked strong support. 17 prefectural members of the House of Representatives asked the support with the help of the doctor association and the pharmaceutical association.)	A
16	二の腕を (ninoude wo) かく (kaku). 史子は (ayako wa) かき抱き (kaki-idaki). 浮かびかけた (ukabikaketa) 妙な (myo na) 妄想を (mousou wo) 払い落とすために (haraiotosu tameni), ふるりと (bururi to) 首を (kubi wo) 振った (futta). (Ayako seized him by the upper arm. She shook her head to avoid odd fancy that crossed her mind.)	C

Fig. 7. Generated Sentences

Frequency	Position	Part
4	1:	{N}{PP}
	-1:	{V}
10	1:	{N}{PP}
	-1:	{V}{SP}

5.2 Extracting Frequent Parsing Patterns : Results

A table 2 shows the results of 442 frequent parsing patterns among 30158 patterns under several depths which appear more than once.

Let us note that 276 patterns (62.4%) appear twice and 356 (80.5%) appear 3 times at most among total 442. Also 215 patterns (48.6%) appear under depth 2 and 359 patterns (81.2%) under depth 3. Surprisingly 276 patterns (62.4%) appear 3 times at most under depth 3. These mean 62.4% of everyday sentences contain typical and simple patterns.

5.3 Generating Sentences : Results

As we said, we parse all the test sentences, build the parsing trees and examine whether we can apply any matching rules or not. We show the parsing tree and the morphemes for the test sentence 5 in a table 4 and a figure 5. We found one frequent pattern (frequency 4) in the part A in the figure and another (frequency 10) in the part B, and we generated sentences “saikin ni naru" (*It happens recently.*) and “kenshin wo ukeru" (*I had a medical checkup.*) shown as a tree in a figure 6. During the generation process, we applied two rules as follows:

Eventually we generated several sentences.

最近に(saikin ni) なる(naru).
(It happens recently).
検診を(kenshin wo) 受ける(ukeru).
(I had a medical check up.)
もしやと(moshiya to) 思い(omoi), 恐る恐る(osoruosoru) 同クリニックで
(dou kurinikku de) 受ける(ukeru).
(Wondering the possibility, I did it at the clinic timidly.)

Table 3. Similarity and Coincidence

	Morpheme(input)										Morpheme(output)										Sim	Coin	
	N	PP	V	S	PF	SP	A	AV	DP	AV	C	N	PP	V	S	PF	SP	A	AV	DP			AV
(Sentence)																							
1	6	5	2	1	0	2	0	0	0	0	2	2	2	1	0	1	0	0	0	0	0	0.93	0.50
4	10	6	2	2	1	3	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1	0.85	0.17
5	3	4	3	0	1	1	0	2	0	1	2	2	2	0	0	1	0	0	0	1	0	0.92	0.53
9	14	10	2	3	0	2	0	0	0	0	2	2	2	1	0	1	0	0	0	0	0	0.78	0.16
16	6	6	7	1	0	3	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0.79	0.13

Let us note that we change the last predicate (verb) into its normal form. In this way, we examined all the input sentences and we show all the results of the generation in a table 7. Also we have evaluated the generated sentences using *A*, *B* and *C*.

Finally we show the similarities and coincidences in a table 3. Here we got high values of the average similarity 0.88 and the coincidence 0.37.

5.4 Discussion

As in a table 7, we generated 15 sentences among 20 where 9 sentences scored *A*, 4 sentences *B* and 2 sentences *C*. Looking at the contents, we see about 66% of sentences keep similar semantics but not fully because frequent patterns can't provide all the variations of sentences. On the other hand, we got rather better results about structure similarity and word coincidence. These aspects may allow us to improve sentence generation. One of the reasons why we don't cover various sentences comes from the assumption that low frequency patterns are not useful for sentence generation. For instance we have an original sentence 4 shown in a table 7. In this example, we see a sequence of nouns : "ringyo kankeisha", "keizai", "kyoiku", "gakushiki keikensha" (*forestry, economy, education, experience or academic standing*). Very often such pattern appears rarely and we can't decide the frequent pattern. To improve the situation, domain knowledge may be helpful to detect fundamental patterns (independent of frequency). Also we may examine generalize description of tree structure.

6 Conclusion

In this investigation, we have discussed how to generate simpler sentences to a given sentence by using corpus in specific domains. To do that, we have obtained a collection of rules with morphemes and the frequencies without any knowledge of grammar nor dictionaries. We take our approach of completely different views from transformational grammar theory so that we expect to avoid computational overhead but to have limited applicability. We have collected rules carefully in terms of parsing trees with morphemes. We give a standard way to generate simple (less-structured) sentences to each rule. To see how well the approach works, we examined whether the rules are really suitable or not, and whether we

can interpret the generated sentences correctly or not. We examined only news corpus with no domain knowledge in advance. Nevertheless, we got 70 percents of sentences and recovered 60 percents from the original sentences. This is nothing but the preliminary work, but the approach seems to be promising.

There remain many issues to be fixed. We have examined one kind of corpus but multiple corpuses may cause distribution-dependency. Though we considered frequency, some other domain-specific conditions may provide us with difference kinds of sentence generation.

References

1. Ehara, T., Fukushima, T., Wada, Y., Shirai, K.: Automatic Sentence Partitioning of TV News Sentences for Closed Caption Service to Hearing Impaired People. IEICE Technical Report. Natural language understanding and models of communication 100, pp.17–22 (2000) (in Japanese)
2. Takahashi, T., Iwakura, T., Iida, R., Fujita, A., Inui, K.: KURA - A Transfer-based Lexico-structural Paraphrasing Engine. In: Proc. 6th Natural Language Processing Pacific Rim Symposium (NLPRS) Workshop on Automatic Paraphrasing: Theories and Applications, pp. 37–46 (November 2001)
3. Inui, K., Nogami, M.: A paraphrase-based exploration of cohesiveness criteria. In: Proc.of 8th European Workshop on Natural Language Generation, pp. 101–110 (2001)
4. Kurohshi, S., Nagao, M.: A Syntactic Analysis Method of Long Japanese sentences based on the Detection of Conjunctive Structures. Computational Linguistics 20(4), 507–534 (1994)

Gallbladder Boundary Segmentation from Ultrasound Images Using Active Contour Model

Marcin Ciecholewski

Institute of Computer Science, Jagiellonian University,
ul. Lojasiewicza 6, 30-348 Kraków, Poland
marcin.ciecholewski@ii.uj.edu.pl

Abstract. Extracting the shape of the gallbladder from an ultrasonography (US) image allows superfluous information which is immaterial in the diagnostic process to be eliminated. In this project an active contour model was used to extract the shape of the gallbladder, both for cases free of lesions, and for those showing specific disease units, namely: lithiasis, polyps and changes in the shape of the organ, such as folds or turns of the gallbladder. The approximate shape of the gallbladder was found by applying the motion equation model. The tests conducted have shown that for the 220 US images of the gallbladder, the area error rate (AER) amounted to 18.15%.

1 Introduction

These days, supporting software is increasingly frequently developed for radiology diagnostics. However, for some important organs like the gallbladder there are no ready, practical solutions to help doctors in their work. The basic job of such software is to extract the shape of the organ from the image, to locate and detect disease units.

The job of extracting the gallbladder structure from US images is a difficult process because images have uneven backgrounds, as shown in Fig. 1. In addition, there is a large variety of gallbladder shapes in US images due to individual traits of patients, among other reasons. US images can also present such disease units as lithiasis, polyps, changes of the organ shape like folds, turns and others which hinder extracting the contour.

In general, literature includes many publications about extracting shapes of organs from US images. One group of algorithms are these that detect edges in the image [1,2]. Edges are usually located in areas with a high gradient value on the image, where the values of the grey level clearly change, e.g. from black to white. Edge algorithms yield inexact results when detecting an edge that is dotted and unclear. They are also computationally complex and leave noise which needs to be eliminated later. Another solution is offered by algorithms based on textures. Richard and Keen [11] have developed an algorithm designed for detecting edges in US images using the classification of pixels corresponding to specific characteristics of textures. Although the algorithm is fully automatic, the authors note that it is computationally complex. The computational complexity

of methods based on texture analysis is usually equal to $O(n^4) : W \times H \times r^2$ where: W is the image width, H is its height, and r denotes the length of the ROI side.

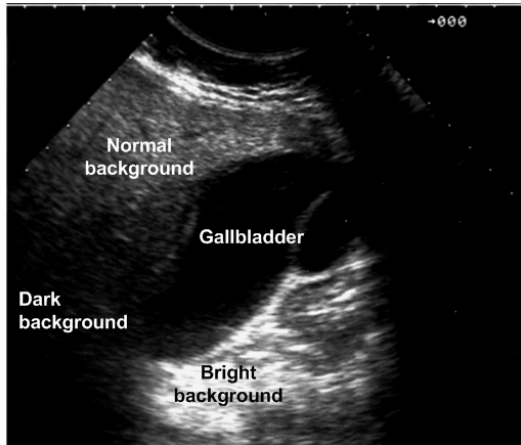


Fig. 1. An example US image of the gallbladder

Algorithms based on deformable models like 2D AAM (the active appearance model) and the active contour (ACM) yield very exact results with relatively low calculation [12,14]. They are usually semi-automatic methods where the initial contour or the average shape model is initiated by the user. AAM models contain information about the average shape of an object, e.g. the lumbar section of the spine on a digital x-ray image [12] and data describing the most characteristic modifications of this shape observed in the training set. The form of the model may be modified by algorithms which try to fit it to the actual shape while not allowing unnatural deformations to appear. The active contour is a mathematical model of a deformable curve located within a two-dimensional environment of an external field created by the local characteristics of the image. The fitting of the model to the shape of the object is an iterative process just as in the case of AAM. Active contour models have been used for US images to determine the shape of such organs as: the carotid artery [7] and the liver [6]. However, they have not yet been used to support the US diagnostics of the gallbladder. In this publication, the motion equation model was used to extract the shape of the gallbladder. The research was conducted on 220 cases from different patients, including US images without lesions and ones showing lesions like: lithiasis, polyps and changes in the shape of the organ, such as folds or turns of the gallbladder. The method for extracting the shape of the gallbladder from US images is presented in the next section. The following section describes the experiments conducted and the research results. The last section contains a summary and sets directions of future research.

2 Extracting the Gallbladder Shape from US Images

This section presents the method of extracting the shape of the gallbladder from US images. The approximate edge of the gallbladder was determined using motion equation model.

2.1 Active Contour Method

An active contour is a mathematical model of a deformable curve made of an abstract, flexible material which reacts to deformations like rubber and springy wire at the same time [8]. In a 2D image analysis context, an active contour is a flat curve which can change its shape dynamically and fit itself to image elements such as edges or borders. The concept of contour shape formation for matching image edges is explained in Fig. 2. The objective of contour movements is to find the best fit, in terms of some cost function, as a trade-off between the contour curvature and the boundary of the image object under analysis. In [8] the potential energy function of the active contour has been proposed to play the role of this cost function. The energy function is given by the following integral equation:

$$E_S = \int_0^{S_{m-1}} [E_i(v(s)) + E_e(v(s)) + E_p(v(s))] ds \quad (1)$$

where the parametric equation $v(s) = (x(s), y(s))$ defines the position of the curve, E_i represents the internal potential energy of the contour, E_e is the energy which models external constraints imposed onto the contour shape, and E_p represents component energies derived from image features, e.g. the image brightness distribution. The notation of the energy function in the discrete format is more convenient in the computer implementation of deformable models:

$$E_S = \sum_{s=0}^{S_{m-1}} [E_i(v(s)) + E_e(v(s)) + E_p(v(s))] \quad (2)$$

In this case, the energy equation is interpreted as the total of component energies of all nodal points. The symbol s symbol is the index identifying the nodal point.

2.2 Motion Equation Model

In this project, the motion equation model proposed in article [9] has been used. This model is treated here as a flexible object of a specific mass moving within an environment of a defined viscosity. Energy E_S is minimized by changing it into the kinetic energy of moving masses of nodal points, subsequently lost as a result of moving within a viscous environment. To model the shifts of individual nodal points, a motion equation of the following form is used:

$$m \frac{\delta^2 v(s, t)}{\delta t^2} + l \frac{\delta v(s, t)}{\delta t} = F(s, t) \quad (3)$$

$$F(s) = -\nabla E_S(s) \quad (4)$$

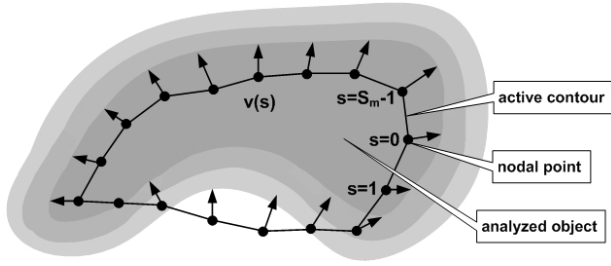


Fig. 2. Building a model of the active contour method. Arrows represent the directions in which nodal points move towards the edge of the analyzed object.

where $v(s, t)$ is the vector of the nodal point coordinates, m is the mass assigned to every node of the graph, l is the viscosity coefficient of the environment, and F is the vector representing all forces acting on the nodes of the structure. The force F for a single nodal point can be determined as the negated value of the gradient of energy E_S calculated in the image (4). The use of the motion equation (3) to describe contour dynamics makes it possible to quickly determine the contour balance state and does not require determining the total minimum value of energy E_S shown by equation (2). In the computer implementation, equation (3) is presented in the discrete form of:

$$m[v(s, t) - 2v(s, t - 1) + v(s, t - 2)] + l[v(s, t) - v(s, t - 1)] = F(s, t - 1) \quad (5)$$

After determining the location of the nodal point at the moment t , we obtain a formula allowing the location of nodal point at the time t to be calculated iteratively based on the values of forces F and their location in the previous two iterations. We obtain:

$$v(s, t) = \frac{F(s, t - 1) + m(2v(s, t - 1) - v(s, t - 2)) + lv(s, t - 1)}{m + l} \quad (6)$$

The numerical convergence and stability of equation (6) depends on the values of parameters m and l , as well as on the way in which force F has been defined. In the case of deformable models, the value of this force depends on many factors, including the features of the analyzed image. The energy minimization method coupled with the motion equation makes it possible to subsequently, in individual iterations, change the location of individual nodal points or of all points at the same time. In the first case, the order of node location modification can be random or defined. If the location of all nodes is modified in the same iteration, equation (6) can be written in the matrix form. The locations of nodes in iteration t are determined based on the values calculated in the previous iteration. The iterative equations have the following form:

$$\begin{aligned} x_t &= \frac{Ax_{t-1} + f_x(x_{t-1}, y_{t-1}) + m(2x_{t-1} - x_{t-2}) + lx_{t-1}}{m + l} \\ y_t &= \frac{Ay_{t-1} + f_y(x_{t-1}, y_{t-1}) + m(2y_{t-1} - y_{t-2}) + ly_{t-1}}{m + l} \end{aligned} \quad (7)$$

In the case of the active contour, matrix A is a pentadiagonal one. For other models, it is a sparse matrix in which the number of elements per row is constant. Consequently, the number of operations increases linearly along with the increasing number of nodal points, and not with the square of their number. This is why this method is more convenient for models with a large number of nodal points.

3 Completed Experiments and Selected Research Results

In order to estimate the precision of models used to determine the approximate contour of the gallbladder, the area error rate (AER) was used. The material from the Department of Image Diagnostics of the Regional Specialist Hospital in Gdańsk, Poland, was used.

3.1 Area Error Rate

The area error rate AER is an estimated value which allows a percentage change in the difference between occupied areas of an image to be compared. The difference in areas is calculated between the area extracted using the active contour model and that extracted manually (MSR).

Let $Lv_{accon} \subset Z^2$ be the fragment of the image obtained using the active contour method and $Lv_{manual} \subset Z^2$ signify the image fragment extracted manually. Let $UR = Lv_{accon} \cup Lv_{manual}$ and $IR = Lv_{accon} \cap Lv_{manual}$. The AER is defined as follows:

$$AER = \frac{a_{UR} - a_{IR}}{a_{MSR}} \times 100\% \quad (8)$$

It was assumed that a_{UR} is the number of pixels within the area UR , while a_{IR} signifies the number of pixels within the area IR , and a_{MSR} is the number of pixels in the manually extracted area MSR . The MSR area of the gallbladder surface was determined by a radiologist and the AER for the 220 US images analyzed amounted to 18.15%.

Table 1 presents the results of experiments for particular disease units in relation to the number of cases. Fig. 3(a) shows an image without lesions, while Fig. 3(b) shows a polyp. Fig. 3(c) shows an image with lithiasis, and Fig. 3(d) a fold of the gallbladder.

Table 1. Test results for 220 US images of the gallbladder

Patients	No. of images	AER
No lesions	100	12.7%
Lithiasis	70	22%
Polyp	20	18.7%
Fold/Turn	30	19.2%
Total	220	18.15%

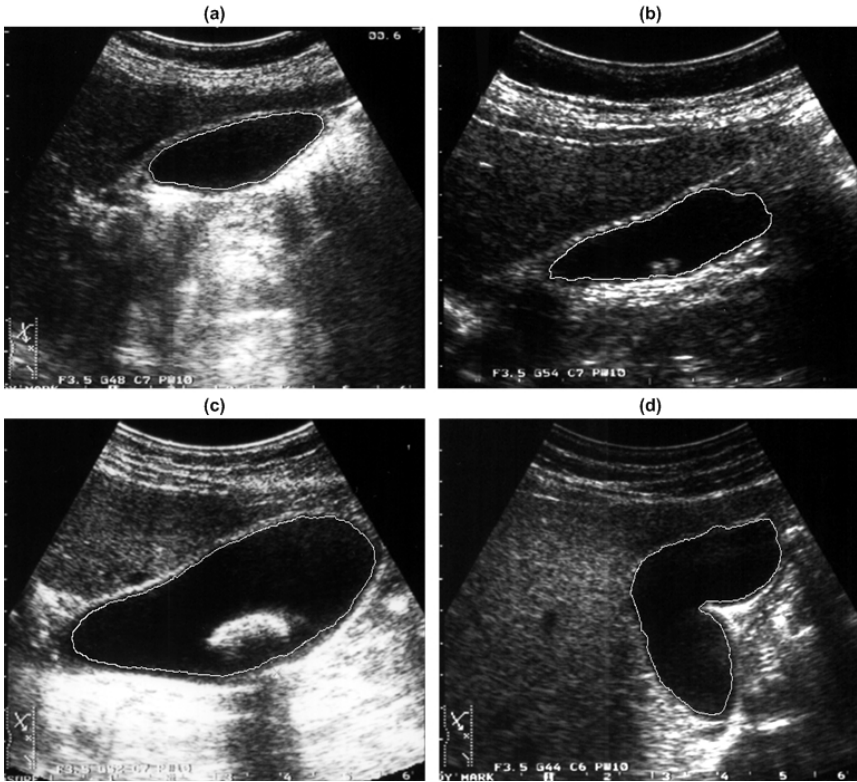


Fig. 3. Extracting the shape of the gallbladder in US images using active contour methods. (a) An image of a gallbladder free of lesions (b) an image with visible cholelithiasis (c) an image showing a polyp inside the gallbladder (d) a gallbladder fold.

4 Summary and Further Research Directions

This article presents a method of extracting the shape of the gallbladder from US images developed for a computer system supporting the early diagnostics of gallbladder lesions. The approximate edge of the gallbladder was determined using motion equation model. The active contour method yielded quite precise results for both healthy organs and those showing specific disease units, namely: lithiasis, polyps, folds and turns of the gallbladder. For the 220 US images analysed, the area error rate amounted to 18.15%. Further research will be aimed at reducing the AER for images showing lesions such as gallbladder folds or turns as well as lithiasis and polyps, if they are located close to the gallbladder edge. Another direction in the research on supporting the US image diagnostics of the gallbladder will be to develop an accurate and reliable methods for recognition and segmentation of disease units.

Acknowledgements

This research was financed with state budget funds for science for 2009-2012 as research project of the Ministry of Science and Higher Education: N N519 406837.

References

1. Aarnink, R.G., Pathak, S.D., de la Rosette, J.J., Debruyne, F.M., Kim, Y., et al.: Edge detection in prostatic ultrasound images using integrated edge maps. *Ultrasonics* 36, 635–642 (1998)
2. Bodzioch, S.: Information reduction in digital image and its influence on the improvement of recognition process. *Automatics, Semi-Annual Journal of the AGH University of Science and Technology* 8(2), 137–150 (2004)
3. Ciecholewski, M., Dębski, K.: Automatic Segmentation of the Liver in CT Images Using a Model of Approximate Contour. In: Levi, A., Savaş, E., Yenigün, H., Balcisoy, S., Saygın, Y. (eds.) *ISCIS 2006*. LNCS, vol. 4263, pp. 75–84. Springer, Heidelberg (2006)
4. Ciecholewski, M., Ogiela, M.: Automatic Segmentation of Single and Multiple Neoplastic Hepatic Lesions in CT Images. In: Mira, J., Álvarez, J.R. (eds.) *IWINAC 2007*. LNCS, vol. 4528, pp. 63–71. Springer, Heidelberg (2007)
5. Cohen, L.D., Cohen, I.: Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(11), 1131–1147 (1993)
6. Cvancarova, M., Albrechtsen, T.F., Brabrand, K., Samset, E.: Segmentation of ultrasound images of liver tumors applying snake algorithms and GVF. *International Congress Series (ICS)*, pp. 218–223 (2005)
7. Hamou, A.K., Osman, S., El-Sakka, M.R.: Carotid Ultrasound Segmentation Using DP Active Contours. In: Kamel, M.S., Campilho, A. (eds.) *ICIAR 2007*. LNCS, vol. 4633, pp. 961–971. Springer, Heidelberg (2007)
8. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. *International Journal of Computer Vision* 1(4), 321–331 (1988)
9. Leymarie, F., Levine, M.D.: Simulating the Grassfire Transform using an Active Contour Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(1), 56–75 (1992)
10. Neuenschwander, W., Fua, P., Kuebler, O.: From Ziplock Snakes to Velcro Surfaces, Automatic Extraction of Man Made Objects from Aerial and Space Images, pp. 105–114. Birkhaeuser Verlag, Basel (1995)
11. Richard, W.D., Keen, C.G.: Automated texture-based segmentation of ultrasound images of the prostate. *Comput. Med. Imaging Graph.* 20(3), 131–140 (1996)
12. Roberts, M.G., Cootes, T.F., Adams, J.E.: Automatic segmentation of lumbar vertebrae on digitised radiographs using linked active appearance models. In: *Proc. Medical Image Understanding and Analysis*, vol. 2, pp. 120–124 (2006)
13. Schilling, R.J., Harris, S.L.: *Applied numerical methods for engineers*. Brooks/Cole Publishing Com., Pacific Grove (2000)
14. Szczypiński, P., Strumiłło, P.: Application of an Active Contour Model for Extraction of Fuzzy and Broken Image Edges. *Machine Graphics & Vision* 5(4), 579–594 (1996)

On the Power of Topological Kernel in Microarray-Based Detection of Cancer

Vilen Jumutc¹ and Pawel Zayakin²

¹ Riga Technical University, Meza 1/4, LV-1658 Riga, Latvia
Jumutc@gmail.com

² Latvian BioMedical Research & Study Center, Ratsupites 1, LV-1067 Riga, Latvia
Pawel@biomed.lu.lv

Abstract. In this paper we propose a new topological kernel for the microarray-based detection of cancer. During many decades microarrays were a convenient approach in detecting and observing tumor-derived proteins and involved genes. Despite of its biomedical success microarray-based diagnostics is still out of common sense in practical biomedicine due to the lack of robust classification methods that would be capable of correct and insensitive to underlying distribution diagnosis of unseen serum samples. This dismal property of microarray datasets comes from probabilistically infeasible difference between cancer specific and healthy samples where only very small number of (anti)genes has prominent tumor-driven expression values. Kernel methods such as SVM partially address this problem being a “state-of-art” general-purpose classification and regression toolbox. Nevertheless, a purely performed normalization or preprocessing steps could easily bias encoded via SVM kernel similarity measures preventing from proper generalization on unseen data. In this paper, the topological kernel effectively addresses the above mentioned issue by incorporating indirect topological similarities between samples and taking into consideration ranking of every attribute within each sample. The experimental evaluations were performed on different microarray datasets and verify that proposed kernel improves performance on purely conditioned and even very small datasets resulting in statistically significant P-values. Finally we demonstrate that proposed kernel works even better without applying cross-sample normalization and rescaling of input space.

1 Introduction

In SVM optimal kernel selection is a crucial and necessary condition for successful classification and good generalization capabilities. Merely good generalization depends on right similarity measure encoded into kernel via appropriate expansion of inner product. This basic feature of all kernel methods provides classification with very flexible but demanding some deeper understanding (meaning one of the problem domain) estimation of similarity.

As we have noticed during post-processing and analysis of collected microarray data there was no significant difference in signal means for cancer specific and

healthy samples resulting in very limited applicability of statistical inference to this kind of classification problem. This also meant that for classical 2-norm SVM with standard RBF kernel absence of major discriminating attributes could potentially lead to highly non-sparse solution with as many support vectors as training examples. Later we have confirmed on almost all evaluated datasets.

In contrast to standard RBF kernel our approach (developed in terms of topological measures) doesn't depend on signal distribution across each attribute but rather works solely with rank information available for every attribute in the sample. This property basically helps to avoid expensive and time-consuming normalization and provides classification with robust and even more accurate estimation of similarity without even proper rescaling of input space.

In general we train and test standard RBF and our proposed kernel within SimpleMKL framework [2] in order to avoid time-expensive cross-validation and provide more accurate estimation of "tuning" parameters.

2 Background

In this section we omit some commonly recognized SVM basics [1] and Multiple Kernel Learning (MKL) extension [3,6,7] for learning from combination of kernels but rather describe briefly our primary data source and problem outline. To avoid misunderstandings we claim MKL method to perform general classification task and estimate optimal bandwidth parameter (γ) of the standard RBF kernel being employed into our topological kernel as well.

2.1 Problem Outline and Data Source

Circulating autoantibodies against tumor-derived proteins have been observed in the most if not all cancer patients therefore they seem to be very attractive targets for the development of noninvasive serological tests for the diagnosis and early detection of cancer. Moreover, the induction of tumor-specific B cell responses by immunotherapy and standard treatments such as radiation and hormone therapies has been observed suggesting that autoantibodies potentially could be exploited as biomarkers of response to therapy. With a phage-display library derived from melanoma, gastric and prostate cancer tissues, we developed and used phage protein microarrays from a set of 1229 different serum-reactive phage clones to analyze serum samples from 172 patients with gastric cancer, 167 patients with melanoma cancer, 52 patients with prostate cancer and 147 samples from healthy control group (healthy donors, HD) as it was developed and described in [8,9]. Further each subset of cancer specific patients was examined versus all HD samples in classification trials of SVM with proposed and RBF kernels.

3 Topological Kernel

In this section we propose new topological kernel as well as present some preliminary generalization bounds obtained for this kernel in terms of Rademacher complexity.

3.1 Formal Definition

Our proposed kernel uses rank information available for each attribute in the sample. This information is acquired by introducing so-called topological measure of every attribute that has its own relative disposition in the sample that doesn't depend on other samples and can be completely regarded as an ordinal ranking of this attribute within each sample. This topological measure can be viewed as two separate quantifiers that account higher and lower ranked attributes in comparison to evaluated attribute. The proposed self-normalized "higher-ranking" based topological measure for i -th attribute of each sample is given as follows:

$$\Omega_{high}(x^{(i)}) = \frac{1}{|\tau|} \sum_j I(\tau_j \geq x^{(i)}). \quad (1)$$

Hereby I is an indicator function and τ is a vector of all possible unique "signals" within sample x . Consequently topological measure for each attribute is self-normalized by the underlying topology of the whole sample x and given cardinality of the vector τ . Similarly "lower-ranking" based topological measure for i -th attribute of each sample is given as follows:

$$\Omega_{low}(x^{(i)}) = \frac{1}{|\tau|} \sum_j I(\tau_j < x^{(i)}). \quad (2)$$

Finally newly proposed kernel that incorporates mentioned topological measures (1) and (2) of two independent samples can be viewed as a dot product of such quantifiers among all attributes. Linear representation of this kernel can be obtained as follows:

$$K_{linear}(x_i, x_j) = \sum_{k=1}^m \Omega_{high}(x_i^{(k)}) \cdot \Omega_{high}(x_j^{(k)}) + \sum_{k=1}^m \Omega_{low}(x_i^{(k)}) \cdot \Omega_{low}(x_j^{(k)}). \quad (3)$$

Hereby m is an input dimensionality of a classification problem. Analogically RBF kernel in terms of the newly proposed kernel function (3) could be expressed as an expansion of the previously defined linear kernel to some highly dimensional Hilbert space:

$$K_{RBF}(x_i, x_j) = e^{-\gamma(K_{linear}(x_i, x_i) - 2 \cdot K_{linear}(x_i, x_j) + K_{linear}(x_j, x_j))}, \quad (4)$$

where γ is a bandwidth parameter of RBF kernel.

3.2 Generalizations Bounds

In this section we propose effective upper generalization bound in terms of Rademacher complexity and specificity of proposed topological kernel. In [5] authors yielded effective upper bound on generalization error using Rademacher complexity as follows:

$$P\{Y \neq f(X)\} \leq P_n\{Y \neq f(X)\} + \frac{R_n(F)}{2} + \sqrt{\frac{\ln 1/\delta}{2n}}, \quad (5)$$

where $R_n(F)$ is Rademacher complexity of a function class F , $P_n\{Y \neq f(X)\}$ is a training error and δ is a confidence interval. Our bound has a similar inference to [3] and is stated for both versions of the self-normalized topological kernel as follows (proof omitted):

$$P\{Y \neq f(X)\} \leq P_n\{Y \neq f(X)\} + \frac{1}{n\gamma} \mathbb{E}\left[\sum_{i=1}^n \sigma_i \cdot C(x_i, m)\right] + \sqrt{\frac{\ln 1/\delta}{2n}}, \quad (6)$$

$$C(x_i, m) = \sqrt{\sum_{j=1}^m \sup |\Omega_{low}(x_i^{(j)})|^2 + m}, \quad (7)$$

where m is an input dimensionality of a classification problem and σ is a vector of Rademacher independent random variables.

4 Experiments

4.1 Preliminaries

In our experiments we have tested proposed model under predefined $C = 10$ (error trade-off) value of the soft-margin SVM that showed most comprehensible performance for imbalanced data sets (number of dimensions \gg number of samples) and varying γ value of RBF Gaussian kernel that trade-offs kernel smoothness and could be effectively estimated via SimpleMKL framework [2]. To exhaustively test our dataset(s) we decided to use 2 different versions of it:

1. Non-normalized dataset directly obtained by scanning microarray chip.
2. Normalized via OLIN [10] and other ad-hoc techniques dataset.

4.2 Additional Scaling

Verifying a performance of the proposed kernel we have found that pseudo-normality of a signal distribution within each sample plays essential role in classification success for all datasets and scaling of input space sometimes leads to performance degradation due to ‘‘corrupted’’ normality. During preliminary trials this observation was commonly met in prostate dataset and thus we have decided to fix the normality of this dataset after scaling by the following δ_i term:

$$\delta_i = \text{mean}(X^{(i)}) - \text{mean}(X), \quad (8)$$

where mean is an averaging operator across all samples and δ_i applies to all values under attribute i by the following update rule: $X^{(i)} = X^{(i)} - \delta_i$.

4.3 Experimental Setup

To verify and test topological and RBF kernels under selected performance measures it was decided to conduct following experimental setup that consisted of some prefixed number of iterations ($I=100$ in our experimental setup) where

every iteration verification set was composed of i.i.d. selected N samples ($N=50$ for all datasets) and all remaining samples were used as the training set. After fixing number of iterations each independent trial we have employed MKL approach to estimate optimal “tuning” parameters for SVM classifier. As it will be seen further we have conducted two separate sets of experiments for scaled and non-scaled invariants of evaluated datasets in order to estimate importance of scaling for proposed and RBF kernels. Finally each set of experiments was divided according to different origin of data sources (melanoma, gastric, prostate etc.) and every separate data source was evaluated jointly in all available invariants (normalized vs. non-normalized) on the same verification set.

4.4 Numerical Results

In the following subsection we have summarized experimental results for all datasets under fixed C parameter and enclosed subspace for γ parameter with some initial guess of its corresponding scaling factor¹. In the Table 1 we present performance measures obtained by MKL approach for SVM with standard RBF kernel and SVM with proposed topological kernel (4). Additionally we analyze and present results under scaled and non-scaled invariants of evaluated datasets with provided number of selected by MKL kernels. All results are averaged across 100 independent trials and provided with standard deviations.

4.5 Statistical Tests

In this section we present results of two-sided t-test applied to classification errors derived from 100 independent trials of MKL method for each invariant of evaluated dataset. The corresponding P-values that indicate level of confidence of the paired comparisons under null-hypothesis of equal error means are represented in Table 2 and Table 3. Note that we have conducted additionally two joint comparisons for normalized and non-normalized invariants of microarray datasets (assuming identical validation sets) in order to show that topological kernel performs significantly (in terms of achieved P-values) better than RBF kernel especially on non-normalized dataset.

4.6 Results Analysis

As we could see from performance measures on different data sources almost everywhere topological kernel attains the same or even better results bringing some useful discrimination capabilities to SVM classifier. For normalized invariant of microarray datasets newly proposed kernel attains just a slight improvement in a total generalization error but for non-normalized one it surprisingly outperforms RBF kernel in all aspects attaining even better results for normalized dataset. The significance of this result could be clearly proven by P-values obtained from

¹ We have defined range of $b_\gamma \cdot 10^{[-10...10]}$ with the step 0.25 resulting in a total of 81 kernels where b_γ is a corresponding scaling factor of γ stated as follows: $b_\gamma = 1/2 \cdot \sqrt{\text{median}(X)}$ where X is a vector of all dataset values.

Table 1. Averaged performance measures

Dataset	Sensitivity	Specificity	Nr.of kernels	Error
Melanoma A ^a (RBF kernel)	0.962±0.037	0.814±0.075	1.980±0.140	0.108±0.038
Melanoma A (Top. kernel)	0.949±0.044	0.845±0.075	2.190±2.312	0.101±0.040
Melanoma B ^b (RBF kernel)	0.861±0.070	0.802±0.092	14.45±3.851	0.167±0.057
Melanoma B (Top. kernel)	0.978±0.037	0.874±0.067	5.540±8.407	0.071±0.036
Melanoma C ^c (RBF kernel)	0.972±0.034	0.791±0.089	1.950±0.219	0.113±0.046
Melanoma C (Top. kernel)	0.936±0.043	0.883±0.076	2.000±0.000	0.088±0.043
Melanoma D ^d (RBF kernel)	0.875±0.070	0.798±0.085	16.74±12.14	0.163±0.050
Melanoma D (Top. kernel)	0.984±0.024	0.881±0.066	5.760±9.342	0.064±0.031
Gastric A (RBF kernel)	0.903±0.070	0.247±0.097	15.98±5.596	0.401±0.069
Gastric A (Top. kernel)	0.917±0.065	0.241±0.090	19.36±2.772	0.396±0.062
Gastric B (RBF kernel)	0.726±0.110	0.505±0.142	5.810±4.948	0.381±0.070
Gastric B (Top. kernel)	0.839±0.078	0.517±0.120	1.960±0.197	0.314±0.063
Gastric C (RBF kernel)	0.821±0.089	0.317±0.110	8.910±9.151	0.416±0.076
Gastric C (Top. kernel)	0.843±0.088	0.360±0.151	14.97±9.066	0.389±0.081
Gastric D (RBF kernel)	0.719±0.114	0.533±0.130	12.39±12.31	0.376±0.068
Gastric D (Top. kernel)	0.889±0.087	0.412±0.141	1.810±0.394	0.338±0.077
Gastric E ^e (RBF kernel)	0.898±0.070	0.227±0.085	11.78±7.819	0.419±0.066
Gastric E (Top. kernel)	0.899±0.070	0.239±0.081	13.58±4.522	0.411±0.066
Gastric F ^f (RBF kernel)	0.719±0.107	0.518±0.135	5.870±5.438	0.380±0.075
Gastric F (Top. kernel)	0.793±0.086	0.665±0.119	12.31±6.799	0.272±0.062
Prostate A (RBF kernel)	0.858±0.104	0.996±0.016	1.140±0.349	0.042±0.032
Prostate A (Top. kernel)	0.886±0.104	0.990±0.024	1±0	0.039±0.032
Prostate B (RBF kernel)	0.635±0.127	0.954±0.034	10.36±5.410	0.131±0.040
Prostate B (Top. kernel)	0.909±0.074	0.996±0.011	1.310±0.465	0.028±0.020
Prostate C (RBF kernel)	0.868±0.116	0.931±0.050	1.060±0.239	0.088±0.043
Prostate C (Top. kernel)	0.608±0.145	1±0	1.080±0.273	0.102±0.048
Prostate D (RBF kernel)	0.664±0.143	0.959±0.032	9.380±10.02	0.118±0.047
Prostate D (Top. kernel)	0.901±0.087	0.999±0.006	1±0	0.026±0.023

^a Normalized scaled dataset

^b Non-normalized scaled dataset

^c Normalized non-scaled dataset

^d Non-normalized non-scaled dataset

^e Normalized rescaled by δ_i dataset

^f Non-normalized rescaled by δ_i dataset

comparison of classification errors on different invariants of microarray datasets². Poor results on normalized versions of datasets lead us to assumption that even very slight (and might be improper) normalization can decrease classifier's discrimination capabilities and classification accuracy. Another interesting assumption is related to more precise nature of topological measures being incorporated into proposed kernel and their superior capabilities in estimating similarity by ignoring attribute's signal distribution across samples and taking into consideration distribution of a signal only within each sample separately.

² Meaning global cross-sample normalization.

Table 2. P-values that indicate the confidence level of significance when comparing proposed topological kernel to RBF kernel in terms of classification error

Dataset	Top. kernel vs. RBF kernel
Melanoma A	0.176
Melanoma B	0.000
Melanoma C	0.000
Melanoma D	0.000
Gastric A	0.598
Gastric B	0.000
Gastric C	0.018
Gastric D	0.000
Gastric E	0.422
Gastric F	0.000
Prostate A	0.482
Prostate B	0.000
Prostate C	0.026
Prostate D	0.000

Table 3. P-values that indicate the confidence level of significance when comparing normalized and non-normalized datasets in terms of classification error (assuming strictly topological kernel to RBF kernel comparison)

Dataset	Norm. vs. non-norm.	Non-norm. vs. norm.
Melanoma AL ^g	0.000	0.000
Melanoma BL ^h	0.000	0.000
Gastric AL	0.000	0.118
Gastric BL	0.000	0.229
Gastric CL ⁱ	0.000	0.002
Prostate AL	0.000	0.000
Prostate BL	0.000	0.021

^g Scaled dataset^h Non-scaled datasetⁱ Rescaled by δ_i dataset

5 Conclusion

In this paper we propose new topological kernel that perfectly fits classification purposes on microarray data and shows very comprehensible results in comparison to standard RBF kernel. The major improvement is observed on very difficult gastric and melanoma datasets that were supposed to be of the greatest biological interest. Nevertheless, obtained results show that there is still a great work to be done in order to maintain good generalization capabilities of microarray-based diagnostic tools that predict a diagnosis irrespectively to a number and underlying distribution of involved (anti)genes. Finally we present

upper bound on generalization error with respect to proposed topological kernel and show that suggested bound correlates with dimensionality of a classification problem and upper bound on involved “lower-ranking” based topological similarity measure. For the future we are considering further research in the field of new kernels and kernel methods for biomedical purposes and CAD³ systems.

References

1. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, Heidelberg (1995)
2. Rakotomamonjy, A., et al.: SimpleMKL. *Journal of Machine Learning Research* 9, 2491–2521 (2008)
3. Lanckriet, G., et al.: Learning the Kernel Matrix with Semidefinite Programming. *Journal of Machine Learning Research* 5, 27–72 (2004)
4. Chapelle, O., et al.: Choosing Multiple Parameters for Support Vector Machines. *Machine Learning* 46, 131–159 (2004)
5. Bartlett, P.L., Mendelson, S.: Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* 3, 463–482 (2002)
6. Sonnenburg, S., et al.: Large scale multiple kernel learning. *Journal of Machine Learning Research* 7(1), 1531–1565 (2006)
7. Bach, F., et al.: Multiple kernel learning, conic duality, and the SMO algorithm. In: *Proceedings of the 21st International Conference on Machine Learning*, Montreal, Canada, pp. 41–48 (2004)
8. Kalnina, Z., et al.: Autoantibody profiles as biomarkers for response to therapy and early detection of cancer. *Current Cancer Therapy Reviews* 4(2), 149–156 (2008)
9. Kalnina, Z., et al.: Evaluation of T7 and Lambda phage display systems for survey of autoantibody profiles in cancer patients. *J. Immunol. Methods* 334(1-2), 37–50 (2008)
10. Futschik, M.: Introduction to OLIN package,
<http://bioconductor.org/packages/2.5/bioc/vignettes/OLIN/inst/doc/OLIN.pdf> (accessed March 25, 2010)

³ Computer aided diagnostics.

An Evolutionary Multi-objective Optimization of Market Structures Using PBIL

Xinyang Li and Andreas Krause

School of Management, University of Bath, Bath BA2 7AY, Great Britain
{x1221, mnsak}@bath.ac.uk

Abstract. We evaluate an agent-based model featuring near-zero-intelligence traders operating in a call market with a wide range of trading rules governing the determination of prices, which orders are executed as well as a range of parameters regarding market intervention by market makers and the presence of informed traders. We optimize these trading rules using a multi-objective population-based incremental learning (PBIL) algorithm seeking to maximize the trading price and minimize the bid-ask spread. Our results suggest that markets should choose a relatively large tick size unless concerns about either the bid-ask spread or the trading price are dominating. We also find that in contrast to trading rules in actual markets, reverse time priority is an optimal priority rule.

1 Introduction

Market microstructure theory as used in conventional finance suggests that the trading rules applied by a market affect the prices at which trades occur, see [12] for an overview. In the highly structured models of market microstructure theory it is, however, difficult to evaluate a wide range of trading rules in a single model. Furthermore, the behavioral assumptions in those models make it difficult to assess the impact the changed trading rules have on the outcome, relative to behavioral influences.

In order to overcome these difficulties we develop an agent-based model in which traders use a very simple trading algorithm which does not assume rational behavior or any other optimizing rule. Such zero-intelligence (ZI) traders have been first introduced in [3] with the explicit aim to investigate the importance of the trading rules for the outcomes of trading. The strategic behavior has been considered to be a dominant influence factor for the market dynamics in previous research. However, [3] find that many of the major properties of double auction markets including the high allocative efficiency are primarily derived from the constraints imposed by the market mechanism, independent of traders' behavior. [4] use such ZI traders to determine the optimal type of auction market. The use of appropriate automatons would allow us to focus on the influence the market structure, i.e. set of trading rules, has on the outcomes. In [5,6] a single-objective optimization of this model has been conducted using the trading-volume and bid-ask spread as objective functions and in this paper we extend this framework to a multi-objective setting to evaluate how any conflicts between different interests in market characteristics might be resolved.

Using the results obtained from this research it is possible to derive recommendations to exchanges, regulators on establishing the optimal market structure, for securities

issuers to choose the best exchange for their listing and for investors to choose the most suitable exchange for trading.

2 Description of the Market

2.1 The Behavior of Traders

We investigate a market in which a fixed number of N traders trade a single asset in a call market. At any time each trader is either a buyer or seller of the asset and submits buy orders $B_i, i = 1, \dots, N$ such that at time t the limit price is taken from a log-normal distribution:

$$\ln B_i^t \sim iidN \left(\ln \bar{P}_t + \mu_{buy}, \sigma_{buy}^2 \right), \quad (1)$$

where \bar{P}_t is the long-term fundamental value in time period t , which we here assume to be equal to the initial price P_0 . μ_{buy} denotes the average amount by which the bid price exceeds the fundamental value, and σ_{buy}^2 represents the variance of bid prices around the mean. While experiments have shown that the exact specification of the decision-making process is not affecting results, we require a minimal amount of information which traders use as a common anchor for their decision; this is necessary to avoid the limit prices and thereby transaction prices to evolve such that an infinitely large bubble emerges. This constraint on the behavior of traders thus implicitly acts as a budget constraint as too large limit prices are not permitted and similarly too small limit prices will not be observed, acting as a minimum size requirement for entering the market.

If we denote by \hat{P}_i^{t-1} the price at which a trader bought the asset the last time, the limit price of a sell order is chosen according to

$$\ln S_i^t \sim iidN \left(\ln \hat{P}_i^{t-1} + \mu_{sell}, \sigma_{sell}^2 \right), \quad (2)$$

in which μ_{sell} denotes the average amount by which the ask price exceeds the price previously paid by the trader, and σ_{sell}^2 represents the variance of ask prices. A trader will only be able to sell those shares he actually holds, i.e. we do not allow for any short sales, thereby acting implicitly as a budget constraint on the behavior of traders.

The order size for a sell order will always be equal to the number of shares held. The order size for buy orders Q_i^t is a random variable with

$$\ln Q_i^t \sim iidN \left(\mu_{size}, \sigma_{size}^2 \right), \quad (3)$$

where μ_{size} denotes the average of the order size, and σ_{size}^2 is the variance of the order size.

An order remains in the order book until it is filled or canceled; for partially filled orders the remainder of the order remains in the order book. An order not filled after T_i^t time steps is canceled, where

$$\ln T_i^t \sim iidN \left(\tau, \sigma_\tau^2 \right), \quad (4)$$

in which τ is the average time of order remains in the order book, and σ_τ^2 denotes the variance of this time.

The canceled order is replaced by a new order taken from the following distributions:

$$\begin{aligned} \ln B_i^t &\sim iidN(\ln \bar{P}_t + \mu_{buy}, \sigma_{buy}^2), \\ \ln S_i^t &\sim iidN(\ln P_i^{t-1} + \mu_{sell}, \sigma_{sell}^2), \end{aligned} \quad (5)$$

where P_i^t denotes the market price at time t .

Whether a trader is a buyer or a seller is determined as follows: if his last transaction was to buy the asset he becomes a seller and if his last transaction was to sell the asset he becomes a buyer. A change from buyer to seller or vice versa only occurs if he has no order remaining in the order book. In the initialization of the experiments buyers and sellers are determined randomly.

2.2 Determination of Transaction Prices

Following the price formation approach applied in [34], the transaction price is determined where the demand and supply curves intersect, i.e. the price at the maximal trading volume is chosen as the trading price; we thus use the concept of a call market in our model. In this market, limit orders with the highest bid prices are first traded and cleared in the market; oppositely, the cheapest sell orders are traded with priority. If we find that there are multiple prices at which the trading volume shows the same maximal value, we employ trading rules to determine which of the prices will be chosen. Any imbalances between buy and sell orders at the transaction price will lead to the need for rationing; how this rationing of buy or sell orders is conducted will depend on the trading rules as outlined below.

2.3 Trading Rules Considered

Tick size. In the market we are able to vary a wide range of trading rules. We will firstly investigate different *tick sizes*, (t), i.e. minimum differences between prices at which orders can be submitted. In order to make limit prices to comply to the tick size, we will lower any limit price of buy orders as determined in (1) and (5) to the next permissible price and similarly raise the limit price of sell orders determined by (2) and (5) to the next permissible price.

Priority rules. Secondly, different *priority rules* are employed to determine the rationing of orders in the case of an imbalance between buy and sell orders at the transaction price, see [78] for an overview of the different priority rules found in several markets. The enforcement of priority rules, as the primary difference between market structures, is another important design feature of trading systems. We use in particular *time priority*, which is the most commonly used rule. It adheres to the principle of first-come first-served, and ensures that orders submitted earlier will be filled first; *reverse time priority* in which orders submitted later will receive priority to be filled; another frequently used rule to promote traders to place larger orders is the *size priority* in which larger orders receive priority; *random selection* in which the orders to be filled are selected randomly and with *pro-rata selection*, a common practice on many financial market such as the Stock Exchange of Hong Kong, the old Toronto Stock Exchange and the batch systems, in which all orders get filled partially to the same fraction.

Multiple prices. Thirdly, for the case of *multiple prices* at which the trading volume is maximal we determine the transaction price to be either the price closest to the previous price, the price furthest from the previous price, the highest price, the lowest price, the price with minimum order imbalance (the absolute difference between the volume of buy and sell orders at the transaction price), the price with maximum order imbalance or a randomly selected price.

Market transparency. Fourthly, we also consider *market transparency*, which is defined by [1] as "the ability of market participants to observe the information in the trading process". In this context, information refers to knowledge about the prices, the size and direction of orders, and the identities of market participants. In a transparent market, traders are able to have access to information on the order book and react to any orders submitted by other traders. In order to replicate this aspect of the market we assume that a fraction of γ of the traders has access to the order book and can observe the potential transaction price as well as the ensuing order imbalance if the trades were to happen instantly. They use this information to revise their own order size according to the size of the order imbalance δ for a buy and sell order, respectively:

$$\begin{aligned}\widehat{Q}_i^t &= Q_i^t - \alpha\delta, \\ \widehat{Q}_i^t &= Q_i^t + \alpha\delta,\end{aligned}\tag{6}$$

where α represents the fraction of order size revised, Q_i^t is the order size before revision, and \widehat{Q}_i^t is the order size after revision. This revised size is then used to determine the transaction price.

Market making. As a final aspect we consider the *intervention of a market maker* into the trading process. A market maker would intervene or influence the prices such that he is prepared to trade a fraction θ of the order imbalance at any time in the market with the existence of imbalance between demand and supply at the transaction price by submitting an offsetting order with price

$$\widehat{P}^t = P^t + \lambda I^t,\tag{7}$$

where I^t denotes the inventory of the market maker, i.e. the number of shares held by him, λ is the price adjustment of market maker; such a model of market making is in line with the literature, e.g. [9][10][11].

2.4 Optimization of Market Structures

The methodology used to optimize the market structure is a computer experiment in which trading is simulated over a given number of time periods with a given market structure. The optimization of the trading rules is conducted evolutionary by population-based incremental learning (PBIL), maximizing the average trading price and minimizing the average bid-ask spread during a simulation as the performance functions for our multi-objective optimization.

The PBIL attempts to create a probability vector, measuring the probability of each bit position having a "1" in a binary solution string who is evolving towards ever better solutions until each is bit is either 1 or zero with certainty [12]. This probability vector π_t is updated based on the following rule

$$\pi_t = (1 - \eta)\pi_{t-1}^* + \eta\hat{v} \quad (8)$$

where π_{t-1}^* denotes the probability of containing a 1 in each bit position that was used in the previous generation, and \hat{v} represents the best solution in the previous generation, selected according to the fitness function of the optimization and η the learning rate. The "best solution" in will be determined as the Pareto-efficient solution closest to the currently used vector. "Closest" is defined by the Euclidean norm and a "Pareto-efficient" solution is one for which there is no solution which is superior for both objective functions. The probabilities are subject to mutation at a mutation rate ξ and the actually chosen probability π_t^* will be

$$\pi_t^* = (1 - \xi)\pi_t + \xi\varepsilon \quad (9)$$

with $\varepsilon \sim U[-1; 1]$ and π_t^* restricted between 1 and 0. This algorithm is capable of maintaining diversity in search as the same probability vector could generate distinct populations.

In each generation we determine 100 different parameter constellations using π_t^* and then determine the best performing parameter constellation from these 100 different market simulations that then makes \hat{v} , subsequently we update π_t^* according to equations (8) and (9). Each trading rule is coded into a vector v , where the precision of the continuous variables $\alpha, \lambda, \gamma, \theta$ is such that each variable is divided into 17 bits each, the tick size t into 20 bits; the discrete variables (priority rules, multiple prices) are coded such that all rules are covered.

As is common with multi-objective optimization, we do not observe an easy convergence of results (even after 5,000 generations no convergence towards a clearly identifiable Pareto-efficient frontier was observed). For this reason we run the optimization for 500 generations and use the entire population of the resulting final generation to analyze our results and determine the Pareto-efficient frontier. This length is about 4 times the length it took the single-objective optimization in [5] and [6] to converge and should therefore represent an adequate time length for the evolutionary algorithm to evolve.

3 Results of Computer Experiments

3.1 Parameter Constellations Considered

We consider a market with 100 traders, which consist of 50 buyers and 50 sellers for the first round. The order book contains the traders' ID number, whether they are buying or selling, their limit price, order size, order submission time and length until the order is to be revised. The initial order book is constructed randomly using the parameter settings given below and the initial price P_0 is set at 100. We assume that the trading price equals the previous price if there is no trading.

We set $\mu_{buy} = -0.1$, $\mu_{sell} = 0.1$, $\sigma_{buy} = \sigma_{sell} = 0.1$, $\tau = 1 + \ln 100$, $\sigma_\tau = 1$, $\mu_{size} = \sigma_{size} = 1$ and investigate tick sizes $t \in [1; 10]$, $\alpha \in [0, 1]$, $\lambda \in [0, 1]$, $\gamma \in [0, 1]$, $\theta \in [0, 1]$, alongside the priority rule and multiple price rules described above.

Each simulation is run for 2,000 time steps, where the first 1,000 data is eliminated for the investigation. The multi-objective PBIL optimization is conducted using a population size of 100 over 500 generations with learning rate of 0.2 and a mutation rate of 0.01. We repeat the multi-objective optimization 25 times with the same parameter constellation to reduce the amount of noise remaining from the lack of convergence.

3.2 Evaluation of Computer Experiments

In the top left panel of figure [1](#) we show the average trading price bid-ask spread of a simulation in the final generation for the entire population of all 25 runs from our computer experiments, restricted to the area close to the Pareto-efficient frontier. We observe a trade-off between the average trading price and the spread, the approximate location of the Pareto-efficient frontier is sketched by the line to the lower right. This figure clearly shows that a low spread will be associated with a low trading price while a high trading price will necessitate a large spread. The optimal combination between trading price and spread will depend on the preferences of the decision-maker and how he values the importance of these two aspects.

We have identified 6 markets that approximately determine the Pareto-efficient frontier, these points are identified as large red points and associated with numbers; the market structures of these six markets are shown in the remaining panels of figure [1](#). From these figures we can deduce some conclusions on the optimal market structure. Firstly we observe that where a decision-maker puts more emphasis on a small bid-ask spread, the tick size should be larger than when the emphasis is on a high trading price. In [6](#) we also found that when minimizing the bid-ask spread, the optimal tick size remained relative high with values up to 7.5%, consistent with our results here. The optimal market structure with respect to the fraction of informed traders as well as the intervention of market makers is much more ambiguous; it is not possible to identify a clear trend in those parameters as we move along the efficient frontier. We can, however, observe from the parameter values shown that the fraction of informed traders should be relatively small and also the degree of intervention by the market maker can be interpreted as small. In all cases the optimal priority rule is reverse time priority and the optimal multi-price rule is to choose the nearest price to the previous transaction price (both results are not depicted in figure [1](#)), thus results showing no sensitivity to them. In [6](#) when minimizing the spread in a single-objective optimization, we also found that the optimal priority rule should be reverse time priority, consistent with our results, while the results for multi-price rules was ambiguous in those instances.

We can summarize our results in stating that for determining the location on the Pareto-efficient frontier only the tick size seems to have a significant impact. For decision-makers that put more emphasis on a low spread, the tick size should be much larger than for those that put more emphasis on a high trading price; in all cases the market should employ reverse time priority and choose the nearest price to the previous in case of multiple potential transaction prices.

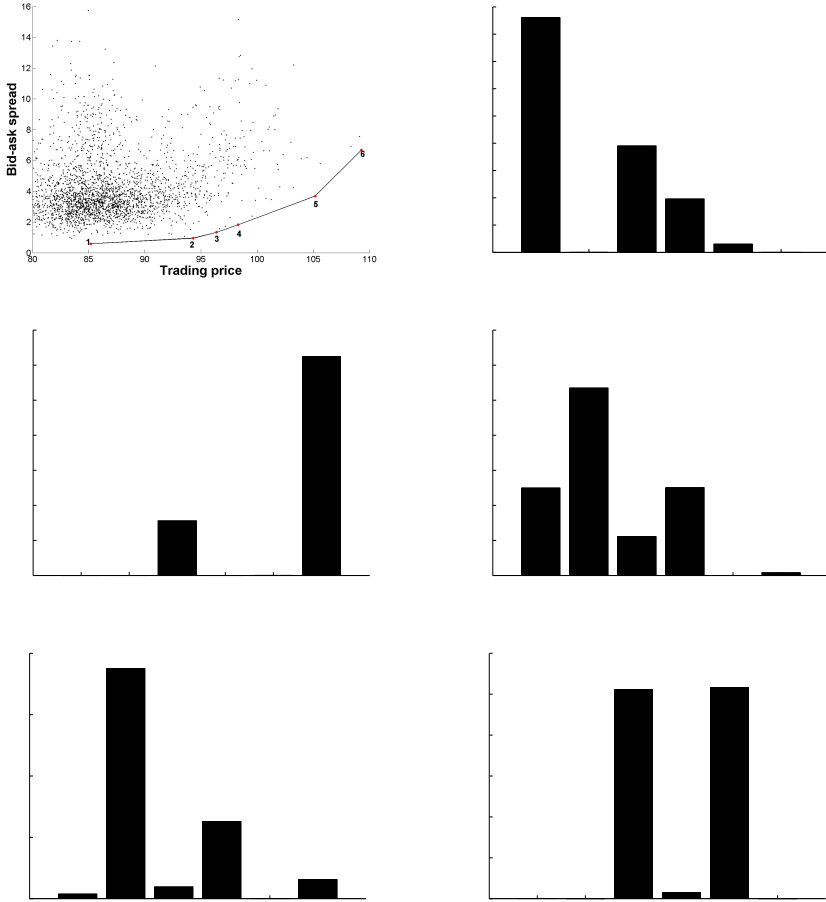


Fig. 1. Market performance and market structures near the Pareto-efficient frontier

4 Conclusions

In this paper we investigate the combination of a wide range of trading rules in a multi-objective optimization by employing population-based incremental learning (PBIL) in call markets seeking to maximize the trading price and minimize the bid-ask spread. As trading rules we include the tick size, priority rules, multi-price rule, intervention of market makers and market transparency. In order to eliminate the influence of complex trader behavior we use an agent-based model in which traders behave nearly randomly, such that any properties arising can be attributed to the impact of the trading rules directly rather than trader behavior.

Conducting such an analysis we analyze the market structures of those markets close to the Pareto-efficient frontier. The results show that when concerns about the bid-ask spread are dominating, a large tick size should be chosen and a small tick size whenever

the trading price is of more concern to the decision-maker. The results on the fraction of informed traders and intervention of market makers are ambiguous but suggest a very limited role. We also find that as a priority rule markets should use reverse time priority rather than time priority as currently done in nearly all markets. These results have direct consequences for the optimal design of financial markets in terms of ensuring high market prices for stocks while also allowing low trading costs in form of the bid-ask spread, and thus might inform any market reforms considered by stock, bond or derivatives markets.

In future research the proposed framework can easily be extended to include different objective functions, like minimizing volatility or maximizing trading volume as alternative or additional objective functions. Such research would allow us to balance a wider range of interest in the market and investigate the sensitivity of the optimal trading rules to the different preferences of decision-makers, thereby giving a more complete picture of the influences on market performance.

References

1. O'Hara, M.: *Market Microstructure Theory*. Blackwell, Oxford (1995)
2. Madhavan, A.: Market microstructure: A survey. *Journal of Financial Markets* 3, 205–258 (2000)
3. Gode, D.K., Sunder, S.: Allocative efficiency of markets with zero-intelligence traders: market as a partial substitute for individual rationality. *Journal of Political Economy* 101, 119–137 (1993)
4. Cliff, D., Bruten, J.: Minimal intelligence agents for bargaining behaviors in market-based environments. HP Lab Report HPL-97-91, HP (2001)
5. Li, X., Krause, A.: Determining the optimal market structure using near-zero intelligence traders. University of Bath (2009)
6. Li, X., Krause, A.: Determination of the optimal trading rules minimizing the bid-ask spread. University of Bath (2009)
7. Schwartz, R.A.: *Equity markets: Structure, trading, and performance*. Harper &, New York (1988)
8. Domowitz, I.: A taxonomy of automated trade execution systems. *Journal of International Money and Finance* 12, 607–631 (1993)
9. Stoll, H.R.: The supply of dealer services in securities markets. *Journal of Finance* 33, 1133–1151 (1978)
10. Ho, T.Y., Stoll, H.R.: On dealer markets under competition. *Journal of Finance* 35, 259–267 (1980)
11. Ho, T.Y., Stoll, H.R.: The dynamics of dealer markets under competition. *Journal of Finance* 38, 1053–1074 (1983)
12. Baluja, S.: Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report CMU-CS-94-163, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (1994)

New Application of Graph Mining to Video Analysis

Hisashi Koga, Tsuji Tomokazu, Takanori Yokoyama, and Toshinori Watanabe

Graduate Schools of Information Systems, University of Electro-Communications,
Chufugaoka 1-5-1, Chofu-si, Tokyo 182-8585, Japan
koga@is.uec.ac.jp

Abstract. Given a graph, frequent graph mining extracts subgraphs appearing frequently as useful knowledge. This paper proposes to exploit graph mining that discovers knowledge without supervision to realize unsupervised image analysis. In particular, we present a background subtraction algorithm from videos in which the background model is acquired without supervision. The targets of our algorithm are videos in which a moving object passes in front of a surveillance camera. After transforming each video frame into a region adjacency graph, our method discovers the subgraph representing the background, exploiting the fact that the background appears in more frames than the moving object.

1 Introduction

As the volume of digital images has increased recently, automatic image annotation has gained much attention. Unsupervised object segmentation from images is an essential task toward automatic image annotation.

On the other hand, frequent pattern mining is a technique to discover useful knowledge from a large amount of data by searching frequent patterns. Its principle is that frequent patterns are not generated by accident. Since knowledge is found by simply seeking frequent patterns, this method may be thought of as an unsupervised knowledge discovery method. Graph mining is a frequent pattern mining which searches frequent subgraphs from a graph.

Whereas many previous researches utilize graphs in image processing like [1] since graphs can express spacial information with edges, very few papers have applied graph mining to image processing. Hence this paper exploits graph mining to realize unsupervised image analysis. Specifically, this paper utilizes the graph mining algorithm SUBDUE [2]. Different from conventional graph mining algorithms like gSpan [3], SUBDUE ranks frequent subgraphs with an evaluation formula. This paper devises an algorithm for the background removal from videos supported by SUBDUE. Our algorithm operates on videos in which the moving object passes in front of a static surveillance camera. In such videos, there exist many frames including the background only before and after the passage of the moving object. Thus, the background appears more frequently than the moving object. Utilizing this feature, our algorithm converts each video frame to a region adjacency graph abbreviated as RAG and discovers the background model as the

top-ranking subgraph in SUBDUE. Our method realizes unsupervised image analysis in that the background model need not be given prerequisites.

In literatures, a few works combine frequent pattern mining with image processing. For static images, Nowozin *et al.* [4] utilize frequent itemset mining to remove outliers from the images collected via the keyword retrieval without supervision. In a supervised situation, they also develop an image classifier using gSpan. Using frequent itemset mining, Yuan *et al.* [5] discover meaningful components of objects from an image database with a single class of objects. For videos, Quack *et al.* [6] apply frequent itemset mining to all the feature points in the video and extract frequent sets of the feature points as interesting patterns. They extend the itemset to encode the spatial information between items while we use the graph instead. Furthermore, we aim to segment the moving object from the background, while [6] does not have such an intension. Our main contribution is to widen the scope of graph mining up to video analysis. In general, graphs are not necessarily suitable for video analysis, since the object actions can change the graph structure. However, the background removal is fit to be solved with graphs, because the background remains stationary and stable.

This paper is organized as follows. Sect. 2 introduces the graph mining algorithm SUBDUE in detail. Sect. 3 explains our background removal algorithm. Sect. 4 reports the experimental results. Sect. 5 mentions the conclusion.

2 SUBDUE

SUBDUE receives a labeled graph $G = (V, E)$ as the input, where V and E are the nodes and the edges, and outputs frequent subgraphs in G . SUBDUE enumerates subgraphs of G and evaluates them by an evaluation formula based on the MDL (Minimum Description Length) principle. Then it ranks the subgraphs according to the evaluation values and outputs the ones in high ranks.

SUBDUE enumerates subgraphs by creating new subgraphs by adding an edge to the subgraphs enumerated in the past. At the beginning, nodes in G with different labels form a set of subgraphs S_1 with one node. Next, new subgraphs are created by adding one edge or (one edge and one node) to the elements in S_1 in all possible ways as guided by G . Then, after computing the evaluation values for these new subgraphs, the high ranking subgraphs among them form the set of subgraphs S_2 that are to be expanded next. Subsequently, the above procedure is repeated until all the subgraphs are considered or the computational time imposed by the user expires. Finally, SUBDUE outputs high-ranking subgraphs of all the enumerated subgraphs. SUBDUE outputs not only the high-ranking subgraphs but also outputs their instances in G .

Let s be a subgraph of G . SUBDUE evaluates s by using the function $val(s) = DL(s) + DL(G|s)$. $DL(x)$ represents the minimum number of bits necessary to describe the graph x . $G|s$ is the graph obtained by shrinking every instance of s in G to a single node. In $val(s)$, s corresponds to the model and $G|s$ grows the compression of G by the model.

Here, the definition of $DL(x)$ is given as $vbits + ebits + rbits$, where $vbits$, $ebits$ and $rbits$ are the minimum number of bits to describe the nodes, the edges

and the rows of the adjacent matrix of x respectively [2]. Let $|V_x|$ and $|E_x|$ be the number of nodes and edges in x and $|L_x|$ be the number of different labels in x . Then, $vbits = \log |V_x| + |V_x| \log |L_x|$ and $ebits = |E_x| + |E_x| \log |L_x| + (1 + \sum_{v \in V_x} v's \text{ degree}) \log (\max \# \text{ of edges between two nodes})$.

According to MDL, SUBDUE values s higher as $val(s)$ gets smaller. $DL(s)$ increases as the size of s reflecting the number of nodes and edges gets larger. On the contrary, $DL(G|s)$ decreases as s gets larger and appears more frequently in G . In total, $val(s)$ becomes smaller when s is larger and appears more frequently in G , since the decrease in $DL(G|s)$ tends to dominate the increase in $DL(s)$. SUBDUE places a subgraph fulfilling the above condition at high ranks.

3 Our Background Removal Algorithm

The targets of our algorithm are videos filmed by a static surveillance camera which monitors the comings and goings of moving objects in indoor environments, e.g., a surveillance camera which monitors people at the entrance of some office. In such a video, moving objects do not appear in the scene so frequently. Furthermore, they pass through the scene in a short time, since they do not stop for long. Hereafter, the moving object is referred to as the foreground. We also assume that the illumination condition does not change throughout the whole video. This condition is satisfied by segmenting the whole video into multiple parts of adequate length and treating each of them individually.

Let F be a video consisting of n frames f_1, f_2, \dots, f_n . In F , the frames before and after the passage of the foreground include the background only as follows.

$$\underbrace{f_1, f_2, \dots, \dots, \dots}_{\text{background only}}, \underbrace{\dots, \dots, f_i, f_{i+1}, \dots, \dots}_{\text{foreground and background}}, \underbrace{\dots, \dots, f_{n-1}, f_n}_{\text{background only}}$$

Thus, the background is more frequent than the combination of the foreground and the background. Let us call “the combination of the background and the foreground” simply as “the *bothground*”. Our algorithm exploits this frequency gap between the background and the bothground to discover the background.

Our algorithms are described below in details. Remarkably it can remove the background from the frames in which the foreground occludes the background and only the subgraph of the graph representing the background exists. In the explanation, we assume that each f_i has been already segmented into regions.

Step 1: Each f_i is converted to a RAG $g_i = (v_i, e_i)$: Every region in f_i becomes a node in g_i . An edge runs between two nodes in g_i , if the two regions associated with the nodes are adjacent in f_i . A node in g_i has a label that shows the attribute class of the associated region in f_i . Next, from g_i for $1 \leq i \leq n$, their sum graph $G = (V, E)$ is constructed by uniting all the edges and the nodes in g_i for $1 \leq i \leq n$ into a large single graph. That is, $V = \cup_{i=1}^n v_i$ and $E = \cup_{i=1}^n e_i$. Hereafter, we refer to the subgraph of G representing the background (the bothground) as the background graph (the bothground graph).

Step 2: We apply SUBDUE to G . Let s_i be the subgraph of G which achieves the i -th rank in the result of SUBDUE.



Fig. 1. Occlusion of the Background Graph

Step 3: This step subtracts the background from each frame under the assumption that the top ranking subgraph s_1 becomes the background graph correctly. The way to subtract the background from a single frame changes, depending on whether the frame contains s_1 in full. Since SUBDUE records all the instances of the discovered subgraphs, the frames including s_1 completely are easily identified. Fig. 1 illustrates the case when the background graph is not contained in the frame in full. For the two frames in Fig. 1, the person is the foreground. Whereas the background graph exists completely in Fig. 1(a), Fig. 1(b) contains only a subgraph of the background graph, since the foreground occludes the window of the house. When f_i contains s_1 in full, the background is removed by subtracting s_1 from g_i . When f_i does not contain s_i completely, only the subgraphs of s_1 appear in g_i because the foreground occludes the background. In this case, the subgraph of s_1 in g_i with the highest rank is removed from g_i . Because SUBDUE remembers the instances of the discovered subgraphs, we have only to examine the discovered subgraphs one by one in the order of $s_2, s_3, s_4 \dots$ until encountering the subgraph of s_1 which has an instance in g_i .

Characteristics of Our Algorithm: By expressing the frames as RAGs, the region-based background removal is enabled. Therefore, even if the camera moves moderately, our algorithm removes the background well except the new regions that arise after the camera moves, so long as the camera motion does not break the background graph.

3.1 Modification of Evaluation Formula

The correctness of our algorithm depends on whether the background graph indeed grows the top rank in SUBDUE. Recall that SUBDUE evaluates subgraphs in terms of the size and the frequency both. Though the background graph appears more frequently than the bothground graph, the bothground graph has a larger size than the background. Hence, the background graph need to appear more frequently enough than the bothground to become the top rank. This paper modifies the evaluation formula $val(x)$ of SUBDUE, so that the background graph may get the top rank even if the frequency gap between the background and the bothground is small. Particularly, we change the definition of $DL(x)$ in $val(x)$ to $DL(x) = vbits = \log |V_x| + |V_x| \log |L_x|$. Namely, we ignore $ebits$ and $rbits$ that are associated with the edges. Note that we still continue to use the edges to judge the graph isomorphism which determines the frequency of a subgraph s and affect $DL(G|s)$.

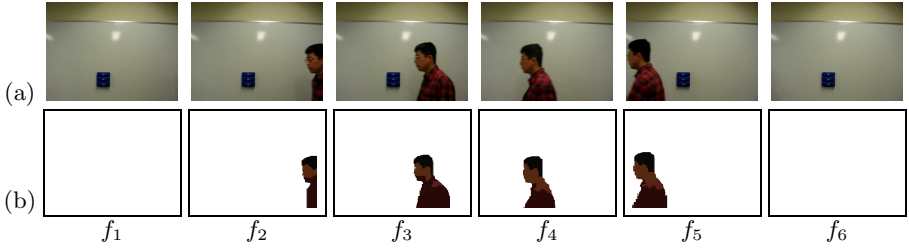


Fig. 2. Case 1: (a) Original Video, (b) Video after Background Removal

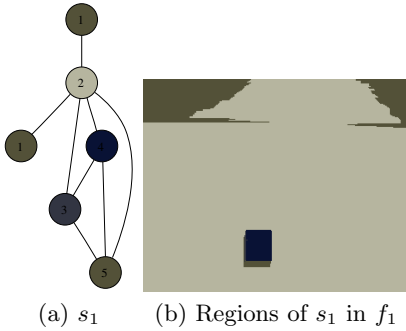


Fig. 3. Relation between s_1 and f_1

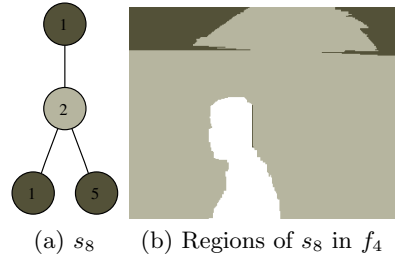


Fig. 4. Relation between s_8 and f_4

This modification considers the nodes only as the graph size. We state the rationale of the modification below: In a RAG x , ordinarily $|E_x| \geq |V_x|$ (except the special case when x is a tree). From this fact and the definitions of $vbits$ and $ebits$ in Sect. 2, the edges dominate the original definition of $DL(x)$. Since $DL(x)$ expresses the graph size in $val(x)$, neglecting the edges in $DL(x)$ weakens the influence of the graph size relatively to the graph frequency in $val(x)$.

In fact, the number of nodes approximates the number of the object components and is adequate to measure the object size. However, because the edges only describe the adjacency relation between the regions, the number of edges is improper to measure the object size.

4 Experimental Results

We evaluate our algorithm by two experiments with real videos. The experimental environment is a PC (CPU: Intel Pentium D 3.20GHz, Memory: 2.0GB). Initially, the input video is preprocessed. First, each frame is segmented into regions by some segmentation algorithm [7]. Next, the attributes of all the regions are clustered to decide the attribute class, where the attributes of a region are the average color in L^*a^*b space and the number of its pixels. Each region is assigned a label according to the clustering result.

Case 1: When the Background Graph is Ocluded Partially

Our algorithm is applied to the video in Fig. 2(a) containing the frame where the background is occluded. In the video, an eraser is placed on a whiteboard and a person walks by them. The video contains 6 frames. The background is formed by the three objects, i.e., the eraser, the whiteboard and the wall over the whiteboard. In f_4 , the eraser, a part of the background is occluded by the foreground. The graph produced from these 6 frames have 53 nodes and 90 edges.

Fig. 3(a) draws the top ranking subgraph s_1 yielded by SUBDUE. s_1 becomes the background graph indeed. For example, Fig. 3(b) shows the regions covered by s_1 in f_1 . As is seen, s_1 covers the whole background area. s_1 does not appear in f_4 , as the foreground hides the eraser there. In f_4 , s_8 in Fig. 4(a) becomes the highest-ranking subgraph of s_1 . Compared with s_1 , s_8 misses the two dark blue nodes representing the eraser. In f_4 , s_8 covers the regions not painted in white in Fig. 4(b). Thus, s_8 does not include the person.

The result after the background removal in Fig. 2(b) demonstrates that our algorithm performs well. The execution time is 0.62s.

Case 2: When the Camera Moves Moderately

To confirm the power of the region-based background removal, our algorithm is applied to the video in Fig. 5 which involves the moderate camera motion. In this video, a green envelop is placed on the whiteboard and a person walks by them. The camera moves rightward between f_1 and f_2 and returns to the previous place by moving leftward between f_2 and f_3 . The background consists of the envelop, the whiteboard, and the wall over the whiteboard in all the frames except f_2 . In f_2 , due to the camera motion, the background consists of the eraser, the whiteboard and the wall over the whiteboard.

After applying SUBDUE, the subgraph in Fig. 6(a) becomes s_1 . s_1 coincides with the background graph: For instance, s_1 covers the whole regions in Fig. 6(b) in f_1 . s_1 contains the four nodes with the label “2” each of which corresponds to a region on the whiteboard generated by the reflection of the light. s_1 does not appear in f_2 whose background differs from other frames. In f_2 , s_{30} in Fig. 7(a) becomes the highest-ranking subgraph of s_1 . s_{30} misses the two green nodes for the envelop and the two nodes for the reflection of the light on the whiteboard, compared with s_1 . In f_2 , s_{30} occupies the regions not painted in white in Fig. 7(b). Thus, s_{30} does not include the eraser.

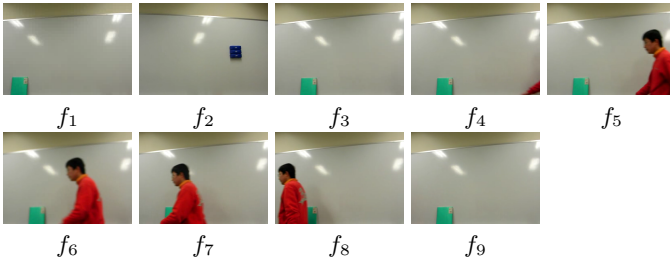


Fig. 5. Original Video

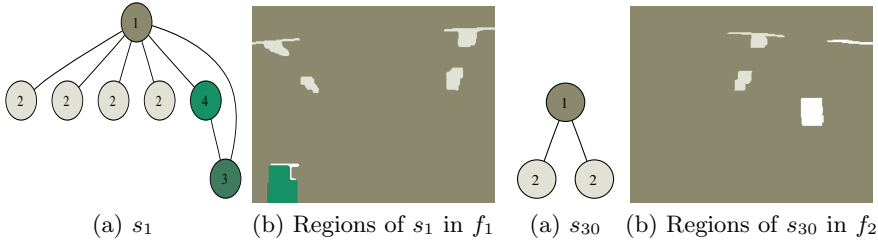


Fig. 6. Relation between s_1 and f_1

Fig. 7. Relation between s_{30} and f_2

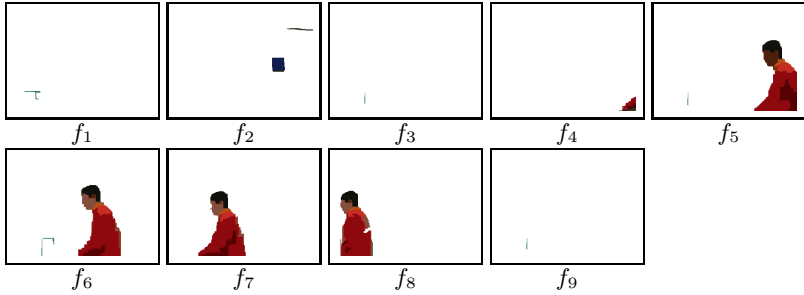


Fig. 8. Video after Background Removal

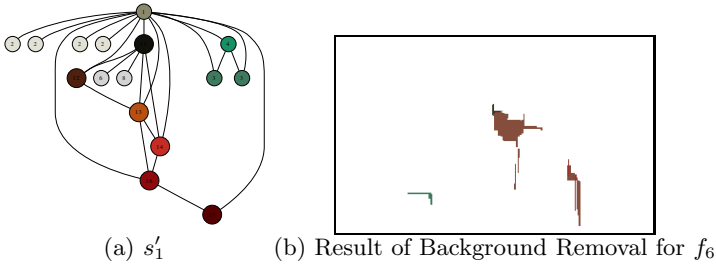


Fig. 9. Background Removal without Modifying the Evaluation Formula

Fig. 8 shows the result after the background removal. The background is removed almost perfectly except f_2 . For f_2 , our region-based background removal algorithm removes the regions which have already existed before the camera motion, despite their locations change. The execution time is 15.3s.

Finally, we discuss the effect of modifying the evaluation formula in SUBDUE. Fig. 9(a) shows the top ranking subgraph s'_1 with the original evaluation formula. s'_1 combines the background and the part of the foreground and has more nodes than s_1 in Fig. 6(a). Though s'_1 has had only 3 instances, it acquires a higher rank than s_1 because s'_1 has more nodes and edges than s_1 . After the background subtraction, the foreground is partially subtracted falsely as shown in Fig. 9(b).

This shows that, by ignoring the edges, the frequency of the graphs gains much importance, making the background graph get the top rank easily.

5 Conclusions

We apply the frequent graph mining algorithm SUBDUE to remove the background from videos. Our method treats videos filmed by a static surveillance camera. By utilizing the property that the background appears more frequently than the foreground in such videos, we acquire the background model as the top ranking subgraph in SUBDUE. By modifying the evaluation formula of SUBDUE, we enable the background graph to get the top rank easily even if the frequency gap between the foreground and the background is small. Our algorithm realizes the region-based background subtraction and, therefore, removes the background despite the moderate camera motion.

Acknowledgments

This work is supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (C), 22500122, 2010.

References

1. Lee, J., Oh, J., Hwang, S.: STRG-Index: Spatio-Temporal Region Graph Indexing for Large Video Databases. In: Proceedings of 2005 ACM SIGMOD International Conference on Management of Data, pp. 718–729 (2005)
2. Cook, D., Holder, L.: Substructure Discovery using Minimum Description Length and Background Knowledge. *J. of Artificial Intelligence Research* 1, 231–255 (1994)
3. Yan, X., Han, J.: gSpan: Graph-based Substructure Pattern Mining. In: Proceedings of IEEE ICDM 2002, pp. 721–724 (2002)
4. Nowozin, S., Tsuda, K., Uno, T., Kudo, T., Bakir, G.: Weighted Substructure Mining for Image Analysis. In: Proceedings of IEEE CVPR 2007, pp. 1–8 (2007)
5. Yuan, J., Wu, Y., Yang, M.: From Frequent Itemsets to Semantically Meaningful Visual Patterns. In: Proceedings of 13th ACM SIGKDD 2007, pp. 864–873 (2007)
6. Quack, T., Ferrari, V., Gool, L.V.: Video Mining with Frequent Itemset Configurations. In: Sundaram, H., Naphade, M., Smith, J.R., Rui, Y. (eds.) CIVR 2006. LNCS, vol. 4071, pp. 360–369. Springer, Heidelberg (2006)
7. Tsunoda, N., Watanabe, T., Sugawara, K.: Image Segmentation by Adaptive Thresholding of Minimum Spanning Trees. *IEICE Transactions on Information and Systems* J87-D-2(2), 586–594 (2004) (in Japanese)

Classification by Multiple Reducts-kNN with Confidence

Naohiro Ishii¹, Yuichi Morioka¹, Hiroaki Kimura¹, and Yongguang Bao²

¹ Aichi Institute of Technology
Yachigusa, Yakusacho, Toyota, Japan 470-0392
ishii@aitech.ac.jp
² Aichi Information System
baoyg_860@hotmail.com

Abstract. Most classification studies are done by using all the objects data. It is expected to classify objects by using some subsets data in the total data. A rough set based reduct is a minimal subset of features, which has almost the same discernible power as the entire conditional features. Here, we propose multiple reducts with confidence, which are followed by the k-nearest neighbor to classify documents to improve the classification accuracy. To select better multiple reducts for the classification, we develop a greedy algorithm for the multiple reducts, which is based on the selection of useful attributes for the documents classification. These proposed methods are verified to be effective in the classification on benchmark datasets from the Reuters 21578 data set.

1 Introduction

Rough sets theory firstly introduced by Pawlak[1,2] provides us a new approach to perform data analysis, practically. Rough set has been applied successfully and widely in machine learning and data mining. An important task in rough set based data analysis is computation of the attribute or feature reduct for the classification. By Pawlak[1,2]s rough set theory, a reduct is a minimal subset of features, which has the discernibility power as using the entire features. Then, the reduct uses a minimum number of features and represents a minimal and complete rules set to classify new objects. Reducts use partial data to classify objects, while conventional methods use all data. Finding all reducts of an information system is combinatorial NP-hard computational problem[3,4]. The proposed algorithm starts with the set of CORE features in the rough set. Through backtracking, multiple reducts are constructed using discernibility matrix. After multiple reducts construction, k-nearest neighbor (k-NN) approach[6,7] is adopted for the data classification. Thus, we call here the method, reducts-kNN, which combines multiple reducts with k-NN classifiers, respectively. To improve the classification performance, reducts-kNN with confidence is newly developed. The proposed reducts-kNN with confidence improves the classification accuracy. To select better multiple reducts for the classification, we develop a greedy algorithm for multiple reducts on the selection of useful attributes for the classification. These proposed methods are verified to be effective in the classification on benchmark datasets from the Reuters 21578 data set[5].

2 Rough Set and Multiple Reducts

2.1 Information Systems for Rough Set

An information system is composed of a 4-tuple as follows,

$$S = \langle U, Q, V, f \rangle$$

where U is the closed universe, a finite nonempty set of N objects (x_1, x_2, \dots, x_N) , Q is a finite nonempty set of n features $\{q_1, q_2, \dots, q_n\}$, $V = \bigcup_{q \in Q} V_q$, where V_q is a domain (value) of the feature q , $f : U \times Q \rightarrow V$ is the total decision function called the information such that $f(x, q) \in V_q$, for every $q \in Q, x \in U$.

Any subset P of Q determines a binary relation on U , which will be called an indiscernibility relation denoted by $INP(P)$, and defined as follows: $xI_P y$ if and only if $f(x, a) = f(y, a)$ for every $a \in P$.

2.2 Reduct of Rough Set

Reduct is a fundamental concept of rough set[1,2]. A reduct is the essential part of an information system S that can discern all objects discernible by original information system.

```

Step 1 Create the discernibility matrix DM:[ Cij ];
CORE =  $\bigcup \{c \in DM : card(c) = 1; i = 1;$ 
Step 2 While ( $i \leq m$ ) do begin
    REDU = CORE; DL = DM - REDU;
    /*forward selection*/
    While ( $DL \neq \emptyset$ ) do begin
        Compute the frequency value for each feature  $q \in Q - \bigcup REDU_i$ ;
        Select the feature  $q$  with maximum frequency value and add it to REDU;
        Delete elements  $dl$  of DL which  $q \in dl$  from DL;
    End
    /*backward elimination*/
    N = card(REDU - CORE);
    For  $j = 0$  to  $N - 1$  do begin
        Remove  $a_i \in REDU - CORE$  from REDU;
        If  $COMP(REDU, DM) = 0$  Then add  $a_i$  to REDU;
    End
    REDUi = REDU;  $i = i + 1$ 
End

```

Fig. 1. Generation of multiple reducts

Let $q \in Q$. A feature q is dispensable in S , if $IND(Q - q) = IND(Q)$; otherwise feature q is indispensable in S . The set $R \subseteq Q$ of feature will be called a reduct of Q , if $IND(R) = IND(Q)$ and all features of R are indispensable in S . We denoted it as $RED(Q)$ or $RED(S)$. The set of all indispensable from the set Q is called $CORE$ of Q and denoted by $CORE(Q): CORE(Q) = \bigcap RED(Q)$.

2.3 Generation of Multiple Reducts

Our algorithm to make reducts of data, starts with $CORE$ features. Multiple reducts are generated by using forward stepwise selection and backward elimination based on the significance values of features as shown in Fig. 1. Let $COMP(B, ADL)$ denote the comparison procedure. Result of $COMP(B, ADL)$ is 1 if for each element c_j of ADL has $B \cap c_j \neq \emptyset$ otherwise 0, m be the parameter of reducts number.

3 k-Nearest Neighbor Method

By using multiple reducts developed above, the data classification is needed. The basic k-nearest neighbor(kNN)[6,7] is one of the simplest methods for classification. It is intuitive and easy to understand it's concept. Much attention is paid to combining a set of individual classifiers with the hope of improving the overall classification accuracy[7].

3.1 k-Nearest Neighbor Classification with Multiple Reducts

The multiple reducts can be formulated precisely and in a unified way within the framework of rough set theory[1,2,3,4]. An attempt of combining multiple reducts followed by the k-nearest neighbor classification is proposed here, which is shown in Fig.2.

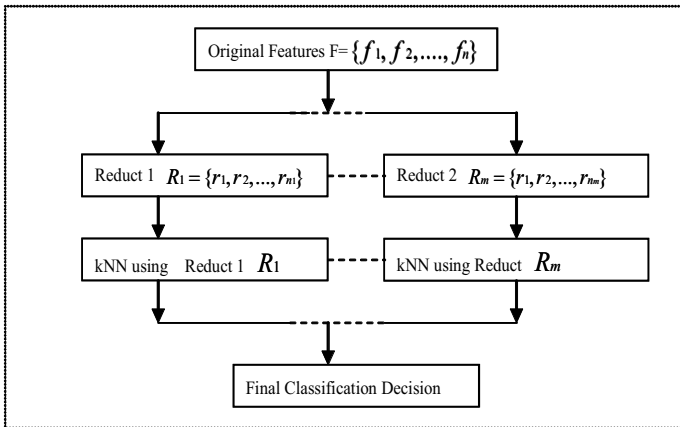


Fig. 2. Multiple reducts-kNN classifiers

In Fig.2, we assume a set of m reducts $\{R_1, R_2, \dots, R_m\}$. Each reduct R_j classifies into a certain class by k-NN. The algorithm by the reducts-kNN classification is given in the following steps,

(1) Let the notation of training data set, be D and the total feature set be $F = \{f_1, f_2, \dots, f_x\}$. From the data set D , multiple reducts R_1, R_2, \dots, R_m are computed.

(2) Each kNN classifier computes rank of each class by using only features included in the given reduct. The distance $sim_i(d_a, d_b)$ between data d_a and d_b in the classifier kNN_i corresponding to reduct R_i ($1 \leq i \leq m$), is defined as follows,

$$sim_i(d_a, d_b) = \frac{\sum_{j=1}^x (count(R_i, f_j) \times x_{aj} \times x_{bj})}{\sqrt{\sum_{j=1}^x (count(R_i, f_j) \times (x_{aj})^2) \times \sum_{j=1}^x (count(R_i, f_j) \times (x_{bj})^2)}},$$

where $count(R, f)$ takes value 1, when feature f is included in reduct R , while it takes value 0 when it is not included. Rank of the class c_j for data d_q , denoted by $rank_{c_j, knni}(d_q)$, is given as

$$rank_{c_j, knni}(d_q) = \frac{\sum_{j=1}^{nk} sim(d_q, d_i) \times \delta(c_j, y_i)}{\sum_{j=1}^{nk} sim(d_q, d_j)}, \text{ where } \delta(c_j, y_i) = \begin{cases} 1: c_j = y_i \\ 0: c_j \neq y_i \end{cases}$$

(3) For integrating respective kNN classifiers, total rank score, $rank_{c_j}(d_q)$ is computed as follows,

$$rank_{c_j}(d_q) = \frac{\sum_{i=1}^m \{rank_{c_j, knni}(d_q)\}}{m}$$

Finally, the classes that satisfy the condition of $rank_{c_j}(d_q) \geq \theta$ are the final classification result.

3.2 Experimental Conditions and Results

For evaluating the efficiency of the kNN classification with multiple reducts, experimental computations are carried out. To measure the classification accuracy in the class C_i , three indexes, $Recall_{C_i}$, $Precision_{C_i}$, and $Accuracy_{C_i}$ are defined as follows. We assume here $\{c_i : i = 1 \sim m\}$ to be class set, where m is the number of class. Let TP_i be the number of correctly classified documents as in C_i . Let FP_i be the number of incorrectly classified documents as in C_i . Let FN_i be the number of incorrectly classified documents as in not C_i . Let TN_i be the number of correctly classified documents as in not C_i . These parameters, are shown as in Table 1. The indexes, $Recall_{C_i}$, $Precision_{C_i}$ and $Accuracy_{C_i}$ are defined for the class C_i as in Fig. 3.

Table 1. Contingency table

	Belong to C_i	Not belong to C_i
Classified to C_i	TP_i	FP_i
Not classified to C_i	FN_i	TN_i

To measure and improve the inter-class accuracy, the indexes: micro-average, and macro-average, are introduced. The former is the modified indexes for all class-es $\{c_i\}$, while the latter is the average values of all classes.

$$Recall_{c_i} = \frac{TP_i}{TP_i + FN_i} \quad Precision_{c_i} = \frac{TP_i}{TP_i + FP_i}$$

$$Accuracy_{c_i} = \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i}$$

Fig. 3. Classification indexes**Table 2.** Evaluation of kNN and reduces-kNN

kNN and multiple reducts		2-Clas s.	3- Class.	4-Cla ss.	5- Class.
kNN	Recall	0.971	0.961	0.860	0.811
	Precision	0.726	0.713	0.740	0.789
	Accuracy	0.797	0.851	0.869	0.898
Reducts- kNN	Recall	0.716	0.868	0.812	0.830
	Precision	1.000	0.905	0.875	0.863
	Accuracy	0.859	0.925	0.909	0.928

From experimental evaluation results, the number of reducts shows to be better for 5 reducts of 5 classes in recall, precision and accuracy values. We adopted 5 reducts in multiple reducts for Reuters data set[5]. The classification evaluations between kNN and the multiple reducts with classes 2, 3, 4 and 5 are compared in Table 2. Table 2 shows the better improvement for the classification accuracy in the reducts-kNN comparing to only kNN method. To improve further the algorithm of the reducts-kNN classification in section 3.1, a score $trank_{C_j}(d_q)$ is newly added by introducing reliable confidence coefficient as follows. The confidence implies the weighting factor of respective reduct-kNN classifier. Confidence approach is discussed in case-based reasoning[8,9,10].

$$trank_{C_j}(d_q) = \frac{\sum_{i=1}^m \{rank_{C_j, kmi}(d_q) \times reliable(d_q, knn_i)\}}{\sum_{i=1}^m reliable(d_q, knn_i)}$$

Where confidence(reliable) coefficient is defined as

$$\text{reliable}(d_q, knn_i) = 2 \times \{ \text{rank}_{Cok, check_kmi}(d_q) / \text{rank}_{Cng, check_kmi}(d_q) + \text{rank}_{Cng, check_kmi}(d_q) \} + 1$$

Then, the unseen object is classified as class C_j , which satisfies the following equation,

$$\text{trank}_{C_j}(d_q) \geq \theta$$

where θ is a given threshold value.

The experimental results with confidence are shown in Table 3, in which the improved multiple reduct-kNN is described in reducts-kNN with confidence coefficient. In Table 3, 5-class data (cocoa, copper, cpi, gnp, rubber in Reuters) was experimented. The reducts-kNN with confidence shows better than the reducts-kNN without confidence.

Table 3. Classification by kNN, reducts-kNN and reducts-kNN with confidence

Method and evaluation		cocoa	copper	cpi	gnp	rubber
kNN	Recall	0.867	0.823	0.615	1.000	0.750
	Precision	0.565	0.875	0.800	0.708	1.000
	Accuracy	0.872	0.947	0.851	0.851	0.968
Reducts-kNN	Recall	0.933	0.882	0.500	1.000	0.833
	Precision	0.823	1.000	0.867	0.791	0.833
	Accuracy	0.957	0.979	0.840	0.904	0.957
Reducts-kNN with confidence	Recall	1.000	0.941	0.538	1.000	0.917
	Precision	0.882	1.000	0.933	0.810	1.000
	Accuracy	0.979	0.989	0.862	0.915	0.989

4 Greedy Algorithm for Selection of Multiple Reducts

It is important to make clear what kind of multiple reducts should be chosen for classification. It is needed to investigate the behavior of the multiple reducts. So, we propose a greedy algorithm, which generates modified multiple reducts based on two steps as shown in Fig.1. The first step realizes the removal of redundant attributes, generation of discernible matrix and that of core from the given data. The second step realizes two kinds of operations; forward selection and backward elimination for reducts. The former operation makes reduct, which has the similar discrimination ability with the information system, while the latter removes the unnecessary attributes produced in the former process. Then, the proposed algorithm is changed in the step 2 in Fig.1 as follows.

Modified step 2

/ *forward selection* /

First, let $REDU = CORE$ be set.

Let C be the ordered set of $\{attr_i\}$ satisfying $attr \in m_{ij}$, $REDU \cap m_{ij} = 0$ and the set is sorted in the large number of attributes. A greedy factor, gf ($0 < gf < 1$) is given for the selection of reduct. We assume here that the first element(attribute) a of the set C be chosen as the probability, gf . The second attribute b is chosen as the probability, $(1 - gf) \times gf$. The next attribute c is chosen as the probability, $(1 - gf)^2 \times gf$. Then, $REDU = REDU \cup \{a\}$ is made. The backward elimination is the same as Fig.1. The proposed greedy algorithm shows that the higher ordered attribute is chosen, when the greedy factor, gf becomes large, i.e., approaches to the value 1, while the every attribute is chosen, when gf becomes low. This is derived from the following equation, which shows the ordered m -th attribute is selected by the probability

$$(1 - gf)^{m-1} \times gf$$

When gf approaches 1.0, this equation becomes 0, thus the probability of the selection of the lower ordered attribute becomes 0. To test the proposed greedy algorithm, experiments are carried out. Reuters 5 classes are classified by the selected multiple reducts, which are chosen by greedy factor algorithm.

Table 4. Classification accuracy of selected reducts by greedy factor

reducts	reducts:5 gf: 0.1	reducts:5 gf: 0.4	reducts:5 gf: 0.9	reducts:8 gf: 0.1	reducts: 8 gf: 0.3
cocoa	0.9362	1.0000	1.0000	1.0000	1.0000
copper	0.9894	0.9894	0.9894	0.9894	0.9894
cpi	0.8085	0.8298	0.8192	0.8404	0.7979
gnp	0.8511	0.9787	0.9681	0.9361	0.9681
rubber	0.9362	0.9894	0.9574	0.9361	0.9894

The experimental result is shown in Table 4. Table 4 shows classification accuracy of selected multiple reducts by greedy algorithm. Reducts with 5 number by greedy factor=0.4 show a higher accuracy than other factors.

Greedy factor =0.4 in Table 4 shows the overlapped attributes are often selected among multiple reducts, which will play an important role for the classification of classes of documents.

5 Conclusion

The reducts computation only uses partial data, which is determined from the feature subset of data.. In this paper, first, we propose multiple reducts with confidence, which are followed by k-nearest neighbor classification to classify documents with the higher classification accuracy. Second, to select better multiple reducts for the classification, we propose a greedy algorithm for the multiple reducts, which is based on the selection of useful attributes for the documents classification. By this algorithm, it was clarified that some overlapped attributes in the multiple reducts are useful in the classification of documents.

References

1. Pawlak, Z.: Rough Sets. *International Journal of Computer and Information Science* 11, 341–356 (1982)
2. Pawlak, Z., Slowinski, R.: Rough Set Approach to Multi-attribute Decision Analysis. *European Journal of Operations Research* 72, 443–459 (1994)
3. Skowron, A., Rauszer, C.: The Discernibility Matrices and Functions in Information Systems. In: *Intelligent Decision Support- Handbook of Application and Advances of Rough Sets Theory*, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
4. Skowron, A., Polkowski, L.: Decision Algorithms, A Survey of Rough Set Theoretic Methods. *Fundamenta Informaticae* 30(3-4), 345–358 (1997)
5. <http://www.daviddlewis.com/resources/testcollections/reuters21578>
6. Bao, Y., Aoyama, S., Du, X., Yamada, K., Ishii, N.: A Rough Set –Based Hybrid Method to Text Categorization. In: *Proc. 2nd International Conference on Web Information Systems Engineering*, pp. 254–261. IEEE Computer Society, Los Alamitos (2001)
7. Bao, Y., Tsuchiya, E., Ishii, N.: Classification by Instance-Based Learning. In: Gallagher, M., Hogan, J.P., Maire, F. (eds.) *IDEAL 2005*. LNCS, vol. 3578, pp. 133–140. Springer, Heidelberg (2005)
8. Momin, B.F., Mitra, S., Gupta, R.D.: Reduct Generation and Classification of Gene Expression Data. In: *Proc. International Conference on Hybrid Information Technology-ICHIT'06*, vol. I, pp. 699–708. IEEE Computer Society, Los Alamitos (2006)
9. Cheetham, W., Price, K.: Measures of solution accuracy in case-based reasoning system. In: Funk, P., González Calero, P.A. (eds.) *ECCBR 2004*. LNCS (LNAI), vol. 3155, pp. 106–118. Springer, Heidelberg (2004)
10. Delany, S.J., Cunningham, P., Doyle, D., Zamolotskikh, A.: Generating Estimates of Classification Confidence for a Case-Based Spam Filter. In: Muñoz-Ávila, H., Ricci, F. (eds.) *ICCBR 2005*. LNCS (LNAI), vol. 3620, pp. 177–190. Springer, Heidelberg (2005)

Towards Automatic Classification of Wikipedia Content

Julian Szymański

Gdańsk University of Technology,
Narutowicza 11/12, 80-952 Gdańsk, Poland
julian.szymanski@eti.pg.gda.pl

Abstract. Wikipedia – the Free Encyclopedia encounters the problem of proper classification of new articles everyday. The process of assignment of articles to categories is performed manually and it is a time consuming task. It requires knowledge about Wikipedia structure, which is beyond typical editor competence, which leads to human-caused mistakes – omitting or wrong assignments of articles to categories. The article presents application of SVM classifier for automatic classification of documents from The Free Encyclopedia. The classifier application has been tested while using two text representations: inter-documents connections (hyperlinks) and word content. The results of the performed experiments evaluated on hand crafted data show that the Wikipedia classification process can be partially automated. The proposed approach can be used for building a decision support system which suggests editors the best categories that fit new content entered to Wikipedia.

1 Introduction

The task of classifying documents is a well known problem [1] with increasing importance in present-days. Currently, humanity produces so much information that its manual cataloging is no longer possible. This forces the development of automated tools, supporting people in processing the information.

The problem of classification concerns also Wikipedia [2] – The Free Encyclopedia. This huge source of knowledge [2], is edited mainly by the volunteers community. Only in October 2009 English Wiki was enriched to an average of 1198 new articles per day [3] (Polish equivalent of about 266 [3]).

The process of classification of Wikipedia content is performed by editors of the article. An editor, that modifies an article, manually indicates to which category the article should be assigned. That task requires some knowledge of the structure of Wikipedia and its category system, but that frequently is beyond typical editor competence. Lack of this knowledge leads to human-caused mistakes – omitting or wrong assignments of articles to categories. Therefore, the purpose of the presented here experiment is to construct a classifier that operates in an automated way, and allows organizing Wikipedia content more efficiently and faster than manually.

¹ <http://en.wikipedia.org>

² <http://stats.wikimedia.org/PL/TablesWikipediaPL.htm>

³ <http://stats.wikimedia.org/EN/TablesWikipediaEN.htm>

2 Our Approach

The problem of automatically classifying documents requires making suitable text representation. The text classification task is relatively easy for humans, because they understand the point of the article they read. Text meaning interpretation is difficult for machines, which don't possess the competences of abstract thinking. Thus they require obtaining characteristic features of the text which allows to distinct one document from another.

In the article we study two typical [3] methods of text representation:

1. based on links – the representation assumes that, the more similar articles are the stronger they are connected via hiperlinks.
2. based on words – the representation of the text is based on the words the document contains. It treats document as a set of words and because it doesn't take into consideration words semantics is called BOW (Bag of Words).

This two approaches allow to construct the feature spaces where documents are represented. Let us assume that k is the number of documents, n denotes the number of features used to describe these documents, while c will mean the value of a certain feature. This allows each of k documents to be represented as a vector of characteristics in n -dimensional space, shown in [1].

$$d_k = [c_{k,1} \ c_{k,2} \ \dots \ c_{k,n}] \quad (1)$$

The feature vectors representing documents are sparse, which is an important observation, since both k and n can be large (especially while using second representation method, size of n is equal to the number of all distinct words in all documents). Because of that we store the data in the form of feature lists related to each document, instead of storing the full matrix.

It should also be noticed that the representation method based on links (1) creates the square matrix of size $n = k$, giving possibility to link article to each other in the peer-to-peer way. In this case, the $c_{k,n}$ value of features take binary values, the corresponding 1 if the link exists, and 0 otherwise.

Articles representation based on words (method 2), assigns n to the number of words that occurred in all articles, which is usually a large value. The value of the feature (a weight that represents a word) in a particular document is computed in the same way as in well known method for text representation called Vector Space Model [4].

A weight c assigned to a word is a product of two factors: term frequency tf and inverse term frequency idf [2].

$$c_{k,n} = tf_{k,n} \cdot idf_n \quad (2)$$

The term frequency is computed as the number of word occurrences in a document and divided by the total number of words in the document. The frequency of a word in a text determines the importance of this word of describing the content of the document. If a word appears more often in the document, it is considered as more important. The inverse word frequency increase the weight of words that occur in small number of documents. This measure describes the importance of the word in terms of differentiation.

Words that appear in fewer number of texts brings more information about a text in a documents set. Such a measure is denoted as [3]

$$idf_n = \log\left(\frac{k}{k_{word(n)}}\right) \quad (3)$$

where $k_{word(n)}$ denotes the number of documents that contain term n .

Having the representation, we are able to perform the classification process. In our approach we used the kernel method of Support Vector Machines [5] that is proved to be suitable in text categorization [1].

The Wikipedia category system is hierarchical: the categories may contain the articles and other (sub)categories. Hence it may be concluded that assigning an article to a category is ambiguous. A selected article belongs to the category, which it is directly assigned. However, the article belongs also to the category to which it is assigned indirectly.

This observation led us to perform the tests using two methods of classification:

- first (simplified) – in which all articles (including those in subcategories) belong directly to the main category. This is a simplified approach which assumes that a document belongs to one class.
- second (detailed) – in which each subcategory of the main category is considered as a separate class. This is closer to real-word case and assumes that one document can belong to more than one category.

2.1 Software

Experiment evaluation requires implementation of the appropriate software, which allows to extract and process relevant information from the Internet Encyclopedia. We implement three modules that brings three different functionalities:

- *WikiCategoryDigger* – the application extracts data about connections between articles. Since all the Wikipedia data are publicly available [4], some of the metadata can be downloaded and put into a local database. The Wikipedia database structure is complex [5], but to perform our experiments only three of the available tables were needed:
 1. `page`, which contains the basic meta-information about a selected article and identifies it unambiguously;
 2. `pagelinks`, which contain references between articles and serve as a main source of information
 3. `categorylinks`, which allows to traverse the category graph.

In Wikipedia, categories and articles are treated in the same way, i.e. the only distinction between them within the database table is the namespace to which they belong. The application allows a user to select certain starting categories and the

⁴ Wikipedia download page:

http://en.wikipedia.org/wiki/Wikipedia_database

⁵ Mediawiki database layout

http://www.mediawiki.org/wiki/Manual:Database_layout

depth of category traversing (it can also be infinite – traversing to the leaf categories). This allows to extract only selected parts of Wikipedia and also allows to assign articles to a category in a user-defined way.

- *WikiCrawler* – the application extracts words used in articles. It is made in the form of a web crawler that retrieves a selected list of articles generated by the previous application. Then the application downloads the data and preprocesses its content by removing punctuation, numbers, stop words, performing stemming and storing the results into a local file. The use of the crawling method was necessary because of the volume of the encyclopedia itself and the time needed to put and preprocess its content into local database.
- *WikiClassifier* – the application for classifying the prepared textual data while using SVM approach. The program uses Matthew Johnson’s SVM.NET library⁶ which is a .Net implementation of libsvm library⁷ developed by Chih-Chung Chang and Chih-Jen Lin.

3 Experiments and Results

The experiments we have performed aim at verifying the approach of SVM classification to Wikipedia articles. It would be ideal to perform test on the whole set of articles, but the size of the data should be limited for efficiency reasons. Thus we performed the experiments only within arbitrary chosen categories. Positive verification of the proposed method would lead to implementation of a large scale classifier that would improve the process of assigning articles to categories.

A standard SVM is a two-class classifier it was used as multi-classifier using technique OVA (one-versus-all). The performance of the results have been estimated using the cross-validation technique. The size of the test and the learning set were 90% and 10% respectively. The results of the experiments presented below are averaged values of 10 repetitions of the learning procedure with random selection of objects to a learning set.

3.1 Category Selection

To obtain reliable results we performed experiments in different parts of Wikipedia. Using proposed two methods of text representation we constructed four data sets (packages) used in experiments. Each of the packages has been constructed from four different categories. The categories we’ve used are presented in Table I. Note that the selected categories significantly differ and they do not overlap one another. It allows to test relatively wide range of Wikipedia, it allows to test both methods of classification: simplified one – when classification is performed only for several main categories, and the extended version in which we select subcategories of the main categories.

It should be also noticed that available computing power strongly restricts the diameter of each category field. The number of analyzed articles from each category was

⁶ SVM.NET: <http://www.matthewajohnson.org/software/svm.html>

⁷ LIBSVM: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Table 1. Categories used to construct data sets (packages)

Name of category (Original name – Translation)	Level
Package 1	
Oprogramowanie Microsoftu – Microsoft Software	2
Jezióra – Lakes	2
Zwierzęta jadowite – Venomous animals	2
Piechota – Infantry	3
Package 2	
Komunikacja – Communication	2
Katastrofy – Disasters	2
Pożarnictwo – Fire manship	2
Prawo nowych technologii – New technology Low	2
Package 3	
Filmowcy – Moovie makers	3
Sport – Sport	3
Astrofizyka – Astrophysics	3
Ochrona przyrody – Wildlife conservation	3
Package 4	
Kultura – Culture	2
Religie -Relligions	4
Polska – Poland	2
Literatura – Literature	3

Table 2. Average size of data for different packages and for both methods of text representation

Package	Articles/Category		Words/Category	
	Classification method 1	Classification method 2	Classification method 1	Classification method 2
Package 1	208,25	8 477,5	14,12	574,75
Package 2	172,25	10 844,5	26,5	1 668,38
Package 3	137,25	13 628	11,94	1 185,04
Package 4	172	20 763,75	13,23	1 597,2

limited to about 700 because all tests had to be performed on ordinary PCs. The limitation of the data set has been done by traversing category tree, and selected set of articles that belong to subcategories. Term „level” denotes the depth of the category tree and it limits the number of subcategories used to construct the package. All articles that are connected directly to the category root create level one and those which are connected indirectly create the next levels.

Table 2 presents the level of granularity for each data set and for each method of classification. It should be noticed here that average number of articles in the second method is much smaller than it is in the first one, where categories contain 1 or 2 articles usually. Such situations cause problems for proper classification by SVM because of a small learning set. In practical application the size of the category should be considered i.e. what is the minimal number of objects that forms category.

It is also worth paying attention to the fact that categories within package 2 and 4 are related because there are some articles associated to more then one category. Categories

are completely independent in the rest of packages. Such selection was caused by an attempt to simulate more realistic situation in which an article is hardly ever associated with only one category. Usually it is related to 3 or more categories. The described preparation of data sets containing different assignments of articles to categories aims at examining if it is possible to obtain good results of SVM classification while multi-category articles exist.

3.2 Results

For each of 4 data packages we performed 4 tests where two worked for the first method of classification and the next two for the second one (methods have been described at the end of section 2). Different methods of text representation were analyzed for both tests, giving 16 tests in total. The results have been averaged using cross-validation and they are presented in Figure 1 using links representation on the left figure and for representation based on words on the right.

Most tests of automatic classification performed using SVM give very good results. However, results of article content analysis for the second method of classification differs much from the rest of experiments. The reason is that they are the most difficult problem for classification: the categories can overlaps each other and what the results of the experiments have shown the text representation we used is not perfect – it does not bring enough features to perform classification properly.

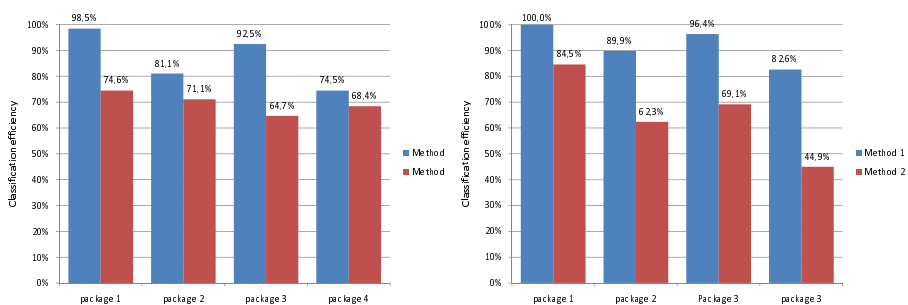


Fig. 1. Results of articles classification for method 1 and 2 using links representation (left) and words representation (right)

Table 3. Average times of learning process for SVM using two text representations

	Classification method 1	Classification method 2	Classification method 1	Classification method 2
Data set	representation by links		representation by links	
Package 1	48 sec.	2" 28 sec.	14" 1 sec.	42" 8 sec.
Package 2	25 sec.	45 sec.	17" i 5 sec.	24" 33 sec
Package 3	12 sec.	33 sec.	11" 17 sec.	30" 45 sec
Package 4	45 sec.	2" 19 sec.	31" 27 sec.	55" 39 sec.
Average	32 sec.	1" 31 sec.	18" 27 sec.	38" 16 sec.

Average time of SVM learning for each data sets is presented in Table 3. The learning and testing processes were executed on hardware listed below:

- results for 1 and 2 data sets (packages) where calculated on a machine with Intel Core Duo 1,7 GHz processor and 1,5 GB RAM memory
- results for 3 and 4 data sets (packages) where calculated on a machine with Intel Core 2 Duo 1,8 GHz processor and 2 GB RAM memory

Averaged results for performed experiments are presented in Table 4. It can be clearly seen the first method of classification gives much better results than the second. It is not surprising because it is an easier case for classification. Moreover, data for the first method of classification give approximately the same results no matter what text representation method is used. It is because of the fact that the problem of classifying objects that significantly differ from one another is relatively easy for machine learning because the data contain features that describe general categories well.

The second method of classification, when one object can be assigned to more than one category and when categories can overlap causes some problems for SVM. We think the fundamental thing here to improve the results is to introduce more effective text representation that brings more informative (in sense of text semantic) features to a classifier.

Table 4. Average measure of classification efficacy

Classification method 1 + text representation with links	86,65%
Classification method 2 + text representation with links	68,70%
Classification method 1 + text representation with words	92,21%
Classification method 2 + text representation with words	65,22%

4 Discussion and Future Plans

The article presents an approach to Wikipedia document classification using the SVM approach. The obtained results of classification (Figure 1 blue bars) show that when classes significantly differ from one another (classification method 1) SVM method gives very good results. Analysis of results of the classification indicates the text representation based on links is better than words. What more, analysis of the efficiency, given in Table 3, indicates the approach using links representation is also much faster and it will allow to build a large scale classifier in a reasonable time. It is because of the fact that the representation based on links is more compact and produces fewer features that are more informative in terms of classification.

The basis of good results of the text classification is text representation. The approach based on links and words presented in the article should be extended to allow calculate text similarity better. A sample modification, which surely improve text classification is combining both presented approaches to text representation.

All the performed experiments were based on the Polish version of Wikipedia. An interesting experiment will be to repeat them in the English version of the encyclopedia. The articles contained there are not only longer and richer (which can improve the results of semantic analysis), but also there are much more of them. This increases the

number of data in test categories and because the linkage graph is denser it can improve the results of the classification through the links.

The proposed approach that operates only on selected parts of the Wikipedia determined by arbitrarily chosen categories was used due to the number of the analyzed data. It seems impossible to conduct experiments in the form presented here for the whole Wikipedia and some optimizations should be considered. One of them is to perform dimension reduction, which allows to combine strongly correlated features (and thus having the smallest information value in terms of classification) in one, and minimize the size of vectors representing articles.

We also plan to research methods of text representation. We plan to improve presented here representation on words by extending it such that it can deliver semantic. The main idea is to map articles into a proper place of the Semantic Network and then calculate distances between them. We plan to use WordNet dictionary [6] as the Semantic Network. We will use word disambiguation techniques [7] that allow to map words to its proper synsets to perform proper mappings. We made some research in this direction and the first results seem very promising [8].

Acknowledgment

This work was supported by the Polish Ministry of Higher Education under research grant no. N N519 432 338.

References

1. Sebastiani, F.: Machine learning in automated text categorization. *ACM computing surveys (CSUR)* 34, 1–47 (2002)
2. Voss, J.: Measuring wikipedia. In: *Proc. of International Conference of the International Society for Scientometrics and Informetrics (2005)*
3. Liu, N., Zhang, B., Yan, J., Chen, Z., Liu, W., Bai, F., Chien, L.: Text representation: From vector to tensor. In: *Proceedings of the Fifth IEEE International Conference on Data Mining*, pp. 725–728. IEEE Computer Society, Los Alamitos (2005)
4. Wong, S.K.M., Ziarko, W., Wong, P.C.N.: Generalized vector spaces model in information retrieval. In: *SIGIR '85: Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 18–25. ACM Press, New York (1985)
5. Hearst, M., Dumais, S., Osman, E., Platt, J., Scholkopf, B.: Support vector machines. *IEEE Intelligent systems* 13, 18–28 (1998)
6. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: An On-line Lexical Database. In: *Cognitive Science Laboratory*. Princeton University Press, Princeton (1993)
7. Voorhees, E.: Using WordNet to disambiguate word senses for text retrieval. In: *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 171–180. ACM, New York (1993)
8. Szymański, J., Mizgier, A., Szopiński, M., Lubomski, P.: Ujednoznacznianie słów przy użyciu słownika WordNet. *Wydawnictwo Naukowe PG TI 2008* 18, 89–195 (2008)

Investigating the Behaviour of Radial Basis Function Networks in Regression and Classification of Geospatial Data

Andrea Guidali¹, Elisabetta Binaghi¹, Mauro Guglielmin², and Marco Pascale¹

¹ Department of Computer Science and Communication
University of Insubria

Via Mazzini 5, 21100 Varese, Italy

² Department of Structural and Functional Biology

University of Insubria

Via Dunant 3, 21100 Varese, Italy

andrea.guidali@uninsubria.it, elisabetta.binaghi@uninsubria.it,

mauro.guglielmin@uninsubria.it, pascalemarco@gmail.com

Abstract. This work investigates learning and generalisation capabilities of Radial Basis Function Networks used to solve function regression and classification tasks in the environmental context. In particular RBFN is applied to solve the problem of snow cover thickness estimation in which critical aspects such as minimal training condition, weak pattern description and inconsistency among data arise. The RBFN shows good performances and high flexibility in coping with regression, hard and soft classifications which are complementary tasks in the analysis of complex environmental phenomena.

Keywords: Radial Basis Function Networks, Regression, Classification, Fuzzy sets, Geospatial data.

1 Introduction

The increasing amount and quality of available geospatial, environmental data, drive the need for new models with analytical, recognition and predictive capabilities. These models are rooted in new techniques based on knowledge-based systems, neural networks, fuzzy logic and hybrids soft computing frameworks. These advances in geospatial computational methods open up new possibilities for the integrated analysis of multisource, multitemporal data to provide accurate estimation of environmental parameters and to provide new information describing our environment [1].

Recent works demonstrated that neural networks (NNs) represent an efficient tool for automatic complex classification and function estimation tasks modelling a variety of non-linear transfer functions [2]. NNs are distribution free and do not require that data conform to a fixed model, an aspect of great potential in the context of current environmental studies which are based on the fusion of multiple, heterogeneous acquired data sets. The attractiveness of NNs also

comes from high parallelism, robustness, learning, ability to handle imprecise and fuzzy information [8]. They can provide practically accurate solutions for precisely or imprecisely formulated problems and for phenomena that are only understood through experimental data and field observations.

In geoscience studies, two neural networks, Radial Basis Function Networks (RBFNs) [4] and Multilayer Perceptron (MLP) [7] have been chosen predominantly among the variety of neural models for their interesting comparable properties in the approximation of nonlinear functions [2]. In this work attention is focused on RBFNs for its proven training speed, robustness to outliers and the capacity to produce confidence values when performing classification and regression tasks [8]. The objective of the study consists in the investigation of learning and generalisation capability of RBFNs specifically addressing situations characterized by minimal training conditions, weak pattern description and inconsistency among data due to error measurement. The objective is pursued by conducting three experiments aimed at the investigation of how a RBFN performs the estimation of snow cover thickness in function of climate and topographic parameters. The snow cover thickness estimation task is modelled in terms of function regression, hard and soft classification.

2 Radial Basis Function Networks

RBFNs are characterized by a very simple three layer architecture. The input layer propagates input values to a single hidden layer. In the output layer, each neuron receives a linear combination of the output of hidden neurons. In case of one output node, the global non linear function computed by the network can be expressed as a linear combination of M basis functions associated to each hidden layer neuron. In formula we have

$$f(\mathbf{x}) = \sum_{j=1}^M w_j h_j(\mathbf{x}) \quad (1)$$

where $\mathbf{x} = [x_1, \dots, x_k]^T$ is the K -dimensional input vector, w_j are the weighting coefficients of the linear combination and $h_j(\mathbf{x})$ represents the output of the Gaussian shaped basis function, with scale factor r_j , associated with the j^{th} neuron in the second layer. The response of j^{th} neuron decrease monotonically with the distance between the input vector \mathbf{x} and the centre of each function $\mathbf{c}_j = [c_{j1}, \dots, c_{jk}]^T$:

$$h_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{r_j}\right) \quad (2)$$

During the training phase, the RBFN learns an approximation for the true input-output relationship basing on a given training set of examples constituted by N input-output pairs $\{\mathbf{x}_i, y_i\}, i = 1, 2, \dots, N$. Following [4], the training scheme is two-phased:

1. phase one is unsupervised and decides values for $\mathbf{c}_j, j = 1, \dots, M$,
2. phase two solves a linear problem to find values for $w_j, j = 1, \dots, M$.

The model configuration requires two user parameters:

1. the number M of first level local processing units and
2. the number p of the p -means heuristic [4], used to determine the scale factor $r_j, j = 1, \dots, M$ of basis functions associated with first level processing units.

The second phase, having model parameters $M, \mathbf{c}_j, j = 1, \dots, M, r_j, j = 1, \dots, M$ known, computes $w_j, j = 1, \dots, M$ minimizing the difference between predicted output and truth by Least Mean Squares, computed through the pseudoinverse (\mathbf{H}^+). In formula

$$\mathbf{w} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} = \mathbf{H}^+ \mathbf{y} \quad (3)$$

where

$$\mathbf{H} = \begin{pmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_M(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & h_M(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & h_2(\mathbf{x}_N) & \cdots & h_M(\mathbf{x}_N) \end{pmatrix} \quad (4)$$

and $\mathbf{y} = [y_1, \dots, y_N]$ is the vector of output data, $\mathbf{w} = [w_1, \dots, w_M]^T$ are second level weights. The trained network is tested using a proper set of examples never seen during training.

3 Hard and Soft Neural Computation

This work is focused on the problem of learning an input-output mapping from a set of examples that can be regarded as an approximation of a multidimensional function. We investigate the behaviour of RBFNs when coping with multidimensional function estimation modelled in the three different settings regression, hard classification and soft classification.

Regression. The RBFN learns from input-output pairs constituted as usual, by input patterns represented by vector of measurements and output values representing numerical function values. The network is configured with a single output neuron.

Hard Classification. In the classification task predefined classes corresponding to intervals of the function co-domain are defined. The underlying assumption is that precision required in regression task is arbitrary due to incompleteness and or inconsistencies among data. During training, input pattern vectors are put in correspondence with a predefined class labels, exemplifying an hard mapping at a lower granularity with respect to regression, with mutually exclusive classes. The network is configured with an output layer having a number of neurons equal to the number of classes.

Soft Classification. The classes are conceived as fuzzy sets with membership functions defined on the function co-domain. The output of the network must

be softened here and the values of the output neurons express the degree of compatibility to the corresponding classes. The learning phase is accomplished teaching the model the gradual membership to all the classes concerned for a given input pattern.

4 Experiments

The RBFNs configured for the three tasks described above has been applied to solve the problem of estimating the snow cover thickness basing on multitemporal, multisource geospatial data.

4.1 Problem Description

Snow cover thickness modelling is an important scientific topic that has been studied for different purposes such as snow avalanche risks, hydrological scope and permafrost distribution [3]. Furthermore snow plays a significant role as an environmental and societal variable and, at the same time, is also an important meteorological and climatological element. Therefore a precise estimates of the snow cover is crucial for different areas. One of the main challenge is related to the fact that snow cover thickness is strongly influenced by many climatic and topographic variables and for each of them it isn't well defined the contribution factor. The literature shows that there isn't a universal accepted method for the evaluation of snow cover thickness that can be applied in every conditions; often the choice of the most suitable method for the estimation of snow cover thickness (but also of others climatic data) depends on temporal resolution, spatial resolution, data quantity and also on the region of interest.

In addition to the theoretical issues many complication arise also during the experiments; for example the processes of gathering, intersecting and clipping the available data to obtain an appropriate and coherent set of patterns dramatically reduced the numerosity of the initial data set, forcing the learning model to work under minimal training conditions.

4.2 Study Area and Data Set

The study area is located in the north part of Italy, in the Italian Central Alps. The total area of this region is about $8,000 \text{ km}^2$. The elevation varies a lot over all the surface and reaches the minimum at 186 meters and the maximum at 4025 meters. We collected data from 136 climatic sensors¹ (not uniformly distributed on the study area), used as initial data set, composed by 23 hydrometers, 19 snow gauges, 46 rain gauges and 48 thermometers. The recording starting date for each sensor is different and the date of the measurements ranges from 1987 to 2003. Unfortunately we couldn't use the entire data set due to the fact that we need location where the measurements for different sensors were present.

¹ Data provided by "ARPA Regione Lombardia".

Moreover to obtain a well balanced data set we decided to take into account only the measurements included in a restricted time period (2002-2003) that has the same number of observations for each location.

In our experiments we decided to use a limited set of features in order to measure the robustness of RBFNs in dealing with weak pattern description. The features are: *elevation*, *aspect*, *slope*, *daily minimum temperature*, *daily mean temperature*, *daily maximum temperature*, *daily rain*, *cumulated rain* on a given temporal window, *average of the daily minimum temperature* computed on a given temporal window, *average of the daily mean temperature* computed on a given temporal window, *average of the daily maximum temperature* computed on a given temporal window. We ignored additional relevant features such as wind, solar radiation and vegetation. The final data set is composed by 5,476 patterns obtained by collecting measurements from 16 locations, in the years 2002 and 2003. Despite the high cardinality the training set represent a low spatial variability.

4.3 Results and Discussion

Different evaluation indexes have been adopted in our experiments. The agreement between truth and classification results has been analyzed by means of the confusion matrix and derived accuracy indexes [5]. The well known Overall Accuracy (OA) index capturing the simple percent agreement between truth and classification results, is complemented with the Cohen's Kappa coefficient (KAPPA) thought to be a more robust measure that takes into account the agreement occurring by chance [6]. The Root Mean Square Error (RMSE) index and its normalized version NRMSE are used to measure the magnitude of network mistakes. Following the train-and-test approach [9] the overall data set, composed 5476 patterns was used in the proportion of $\frac{2}{3}$, $\frac{1}{3}$ for training/validation and test phases respectively.

Regression. First of all we present the results obtained using the RBFNs performing regression task. The RBFN receives in input the vector of measurements derived from the set of features described above. Concerning the network architecture, the input layer has 11 neurons equal to the number of features and the output layer has 1 neuron representing the predicted snow cover thickness value. Several configurations of the RBFN were considered varying the temporal window used in the computation of the features *cumulated rain* and *average temperature*, which assumed values ranging from 10 to 45. For each window size different RBFN configurations were considered distinguished by the different number M of basis functions which assumed values 150, 250, 500. All the configurations considered were evaluated using the validation set. The RBFN showed best behaviour setting temporal window dimension at 45. Table 1 shows the results obtained in this configuration, varying the neural internal parameter M . Results are expressed in terms of RMSE and NRMSE index. In particular NRMSE values are computed dividing the RMSE values by the range of variability [0cm-400cm] of snow cover thickness. The network configuration with M

Table 1. Regression results evaluated on the validation set varying the number of centroids M of RBFN

Centroids	RMSE	NRMSE
150	23	5.7%
250	19	4.7%
500	18	4.5%

equal to 500 showed the best results with RMSE equal to 18. The best network configuration evaluated on the test set provides results with RMSE equal to 17.

The RMSE values obtained indicate an acceptable mean disagreement between reference and predicted values. However we have to consider that different intervals within the snow cover thickness range have different relevance in the environmental analysis and errors computed on these intervals become unacceptable making arbitrary numerical predicted values. We proceeded in modelling the snow cover thickness estimation task as a classification problem in an attempt to reduce the precision of the output values for the benefit of the significance.

Classification. Experts identify four classes, considering the snow cover thickness intervals significant for the net energy balance analysis [3]: Class A: absence of snow cover; Class B: 0cm-10cm; Class C: 10cm-90cm; Class D: greater than 90cm; Concerning the network architecture, the input layer has 11 neurons equal to the number of features and the output layer has 4 neurons equal to the number of classes. Several configurations of the RBFN were considered varying temporal window and parameter M as described for regression task. Also in this case the RBFN showed best behaviour setting temporal window dimension at 45 and M equal to 500. Results obtained with this configuration is showed in Table 2(a). The RBFN performs classification with a good level of accuracy, showing an OA and e KAPPA equal to 83% and 75% respectively when evaluated on the test set.

Soft Classification. As further analysis, we intend to exploit the intrinsic uncertainty underlying the overall classification process that may affect the significance of the hard partition operated on the snow cover thickness interval in the classification process. Proceeding from the above identified snow cover thickness intervals and assessing the imprecision in the measurements of an entity equal to 5 cm, we modelled the four classes as fuzzy sets having membership function (μ_X is the membership function related to the class X) described in Fig 1. Within soft classification framework classes are not mutually exclusive. The learning phase is accomplished submitting input-output pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$ to the RBFN, where \mathbf{x}_i is a given input pattern, \mathbf{y}_i is a vector whose j^{th} component represent the grade of membership of the crisp snow cover thickness value to the j^{th} fuzzy class. Table 2(b) shows best results using test data and expressed in terms of a confusion matrix in which reference and classification memberships to classes, hardened according to the winner take all rule are allocated. Again

Table 2. Confusion matrices for the classification tasks

(a) Confusion matrix for the hard classification

-	ω_1	ω_2	ω_3	ω_4	Tot U	UA
ω_1	375	25	4	0	404	92.82 %
ω_2	67	394	77	5	543	72.56 %
ω_3	21	94	599	7	721	83.08 %
ω_4	0	0	9	148	157	94.27 %
Tot P	463	513	689	160	-	-
PA	80.99 %	76.80 %	86.94 %	92.50 %	-	-

Overall accuracy: 83.0685 % (1516 hit, 309 miss, 1825 total)
 KAPPA value: 75.9328 %, KAPPA std.err: 0.0002

(b) Confusion matrix for the soft classification

-	ω_1	ω_2	ω_3	ω_4	Tot U	UA
ω_1	387	23	6	0	416	93.03 %
ω_2	54	396	68	2	520	76.15 %
ω_3	22	94	604	10	730	82.74 %
ω_4	0	0	11	148	159	93.08 %
Tot P	463	513	689	160	-	-
PA	83.59 %	77.19 %	87.66 %	92.50 %	-	-

Overall: 84.1096 % (1535 hit, 290 miss, 1825 total)
 KAPPA value: 77.4100 %, KAPPA std.err: 0.0002

results obtained are satisfactorily with OA and KAPPA equal to 84.1% and 77.4% respectively. The statistical difference between KAPPA computed using the Z test [5] allows to conclude that comparable performance are obtained for hard and soft classification task.

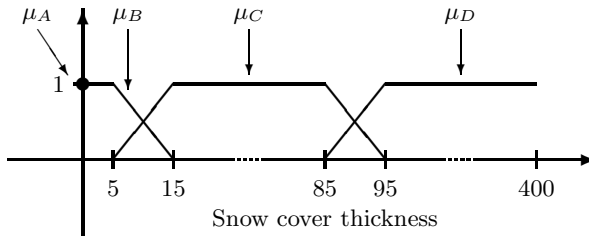


Fig. 1. Expert driven membership functions used for the soft classification task

As further investigation, we conducted a fuzzy classification partitioning the snow cover thickness range uniformly without considering the expert driven criteria. Three fuzzy classes *low*, *medium* and *high* were defined having triangular membership functions. In this configuration the RBFN reaches an OA equal to 97%. These results are consistent with the general claim that the hardening process, when delayed until a final stage, allows to reduce the loss of meaningful information.

5 Conclusions

A deep investigation on the use of RBFNs for environmental parameter estimation has been conducted. As seen in our experimental context, the RBFN model copes very well with critical aspects related to imprecision in the data, weak pattern description and low representation of spatial variability of the environmental variables. It also shows high flexibility to move from regression and hard/soft classification tasks which are usually complementary to understand complex environmental phenomena.

Future works contemplate a final assessment of the fuzzy classification made in the light of a deep discussion with domain experts of the results obtained. Moreover in order to use the maximum number of patterns available in the study area we intend to investigate semi supervised-learning paradigms.

References

1. Belward, A., Binaghi, E., Lanzarone, G.A., Tosi, G. (eds.): Geospatial Knowledge Processing for Natural Resource Management, Special Issue of International Journal of Remote Sensing, vol. 24(20). Taylor & Francis, Abington (2003)
2. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, Oxford (1996)
3. Guglielmin, M., Aldighieri, B., Testa, B.: PERMACLIM: a model for the distribution of mountain permafrost, based on climatic observations. *Geomorphology* 51, 245–257 (2003)
4. Moody, J., Darken, C.J.: Fast Learning in Networks of Locally-Tuned Processing Units. *Neural Computation* 1, 281–294 (1989)
5. Congalton, R.: A review of assessing the Accuracy of classifications of remotely sensed data. *Remote Sensing of Environment* 37(1), 35–46 (1991)
6. Cohen, J.: A Coefficient of Agreement for Nominal Scales Educational and Psychological Measurement, vol. 20, pp. 37–46 (1960)
7. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) *Parallel Distributed Processing*, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)
8. Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial Neural Networks: A Tutorial. *Computer* 29(3), 31–44 (1996)
9. Mitchell, T.M.: *Machine Learning*. McGraw Hill, NewYork (1996)

A Comparison of Three Voting Methods for Bagging with the MLEM2 Algorithm

Clinton Cohagan^{1,2}, Jerzy W. Grzymala-Busse^{1,3}, and Zdzislaw S. Hippe⁴

¹ Department of Electrical Engineering and Computer Science,
University of Kansas, Lawrence, KS 66045, USA

² Kansas City Plant, National Nuclear Security Administration,
Kansas City, MO 64141, USA
ccohagan@kcp.com

³ Institute of Computer Science, Polish Academy of Sciences, 01-237 Warsaw, Poland
jerzy@ku.edu

⁴ Department of Expert Systems and Artificial Intelligence,
University of Information Technology and Management,
35-225 Rzeszow, Poland
zhippe@wsiz.rzeszow.pl

Abstract. This paper presents results of experiments on some data sets using bagging on the MLEM2 rule induction algorithm. Three different methods of ensemble voting, based on support (a non-democratic voting in which ensembles vote with their strengths), strength only (an ensemble with the largest strength decides to which concept a case belongs) and democratic voting (each ensemble has at most one vote) were used. Our conclusions are that though in most cases democratic voting was the best, it is not significantly better than voting based on support. The strength voting was the worst voting method.

1 Introduction

Ensemble learning methods may improve classifier performance. These methods construct an ensemble (a set) of diverse classifiers (resembling the base classifier) that vote on the concept membership. Two methods are used frequently: bagging [1] and boosting [2]. In bagging each classifier is created using different subsets of the training set. These subsets are results of a bootstrap sampling, constructed by drawing cases from the training set with replacement. Thus, a bootstrapped sample has about 63.2% unique cases from the training set. In boosting, the distribution of training cases is adaptively changed in such a way that the classifier will focus on cases that are difficult to classify. A weight is associated with each case. Weights of previously misclassified cases are increased. Hence, during boosting, more difficult cases are more frequently included in the sample. In both bagging and boosting, final predictions from all classifiers are aggregated.

Ensemble learning is conducted in two stages: creation of classifiers and combining classifier predictions. Bagging and boosting are typical methods for creating classifiers, but there are many other techniques of ensemble learning, for

example, random forest methods [3] in which for every ensemble attributes are randomly selected, see also [4–9]. Usually classifier predictions may be aggregated by voting, stacking [10] and cascading [11]. In stacking, a learning algorithm is applied to classifier predictions. Cascading combines classifier predictions iteratively, in each iteration the training data set is associated with the predictions from previous iterations.

The performance of bagging and boosting, combined with some aggregating mechanisms for combining predictions, is comparable. Bagging is more accurate than variations of random forests, randomized C4.5 and random subspaces [12]. For data sets with many conflicting cases bagging is much better than boosting [13].

For our experiments we used as a base classifier the MLEM2 (Modified Learning from Examples Module, version 2) module of the LERS (Learning from Examples based on Rough Sets) data mining system. Additionally, we used bagging with three different techniques of combining predictions of ensemble classifiers: support, strength, and democratic voting. In all three techniques each rule set contributes at most one vote. The democratic voting, also called simple majority voting, is a standard technique of combining classifier votes in bagging and is known as very successful [6]. Our experiments show that the democratic voting is not significantly better than combining classifier votes based on support. On the other hand, combining predictions based on strength is the worst approach.

2 MLEM2

In our experiments, ensembles were rule sets induced by MLEM2 rule induction algorithm. The input data set of the MLEM2 is a lower or upper approximation of a concept [14]. The LEM2 computes a *local covering* and then converts it into a rule set. We will quote a few definitions to describe the MLEM2 algorithm [15–17].

A data set is a collection of cases, each case is characterized by attributes and a decision d . The value of an attribute a for case x will be denoted by $a(x)$. A block of an attribute-value pair $t = (a, v)$, where v is the value $a(x)$ of an attribute a for some case x , will be denoted by $[t]$.

Let X be a nonempty lower or upper approximation of a concept $[(d, w)]$, where w is the value of d for some case x . Set X *depends* on a set T of attribute-value pairs $t = (a, v)$ if and only if

$$\emptyset \neq [T] = \bigcap_{t \in T} [t] \subseteq X.$$

Set T is a *minimal complex* of X if and only if X depends on T and no proper subset T' of T exists such that X depends on T' . Let \mathcal{T} be a nonempty collection of nonempty sets of attribute-value pairs. Then \mathcal{T} is a *local covering* of X if and only if the following conditions are satisfied:

- each member T of \mathcal{T} is a minimal complex of X ,
- $\bigcup_{t \in \mathcal{T}} [T] = X$, and
- \mathcal{T} is minimal, i.e., \mathcal{T} has the smallest possible number of members.

MLEM2 processes numerical attributes differently than symbolic attributes. For numerical attributes MLEM2 sorts all values of a numerical attribute. Then it computes cutpoints as averages for any two consecutive values of the sorted list. For each cutpoint q MLEM2 creates two blocks, the first block contains all cases for which values of the numerical attribute are smaller than q , the second block contains remaining cases, i.e., all cases for which values of the numerical attribute are larger than q . The search space of MLEM2 is the set of all blocks computed this way, together with blocks defined by symbolic attributes. Finally, MLEM2 simplifies rules by merging intervals for numerical attributes.

3 LERS Classification System

This method is based on three parameters: strength, support and matching factor. Note that rule *strength* and *support* were introduced in [18]. *Matching factor* is an original idea of the LERS classification system.

First, the LERS classification system tries to match the case against all rules from the rule set. Each rule is associated with the *strength* that is equal to the number of training cases that are well classified by this rule. With any concept its *support* is computed as follows

$$\sum_{\text{matching rules } r \text{ describing } C} \text{Strength}(r)$$

The case is classified as a member of the concept with the largest support. If complete matching of a case against the rule set is impossible, all partially matched rules are identified and any concept is associated with a *modified support*

$$\sum_{\text{partially matching rules } r \text{ describing } C} \text{Matching_factor}(r) * \text{Strength}(r)$$

where *Matching_factor* is defined as the ratio of the number of matched attribute-value pairs of r with a case to the total number of attribute-value pairs of r . It is possible that a case does not match any rule, even partially. Such a case is not classified. The corresponding rule set abstains from voting. The idea of a rule abstaining from voting was also explored in [5].

4 Ensemble Voting Methods

We used ten-fold cross validation, so for every data set we computed ten pairs of large (90%) training data sets and ten small (10%) testing data sets. For every training data set 100 bootstrap samples were computed and for every such a

Table 1. Error rates for the ensemble size equal to 100

Data set	Ensemble voting methods			Original data set
	Support	Strength	Democratic	
Bankruptcy	4.55	6.06	4.55	4.55
Breast cancer (Slovenia)	29.37	29.37	29.72	29.72
Breast cancer (Wisconsin)	17.92	17.92	17.92	21.12
BUPA	39.13	42.03	33.91	34.78
Echocardiogram	27.03	31.08	29.73	31.08
Glass Identification	34.11	45.33	28.04	30.84
Hepatitis	16.77	20.65	14.58	17.42
Horse colic	37.12	40.13	33.44	35.45
House of Representatives	4.38	5.76	4.61	4.84
Image segmentation	16.19	27.14	14.29	19.05
Iris	4.67	6.67	4.67	4.67
Lymphography	18.24	23.65	15.54	18.24
Pima	33.33	34.38	30.47	31.77
Postoperative patient	28.89	28.89	33.33	37.78
Primary tumor	64.31	70.80	60.18	65.49
Wine recognition	9.55	28.09	7.30	11.24

sample the corresponding rule set was induced by MLEM2. For every testing case, for all applicable rule sets, the LERS classification method was applied.

Thus, all testing cases were associated with dominant supports, or dominant modified supports, or were labeled as not classified (if even partial matching was impossible). Then we used three different voting methods for ensemble voting.

The first method was based on the support, i.e., each rule set voted with the support (modified support) for the case, for any concept all supports were added, and the concept with the largest sum of all corresponding supports was the winner. In the second method, the choice of the concept was determined by the largest support among rule sets. In the third method, democratic voting, supports of rule sets were converted either into number one if the rule set support was greater than zero, or the corresponding rule set was not voting at all (if the corresponding case was not classified by the rule set).

5 Experiments

Our experiments were conducted on 16 data sets. All of these data sets, with the exception of *Bankruptcy*, are available on the University of California at Irvine *Machine Learning Repository*. The *Bankruptcy* data set is a well-known data set

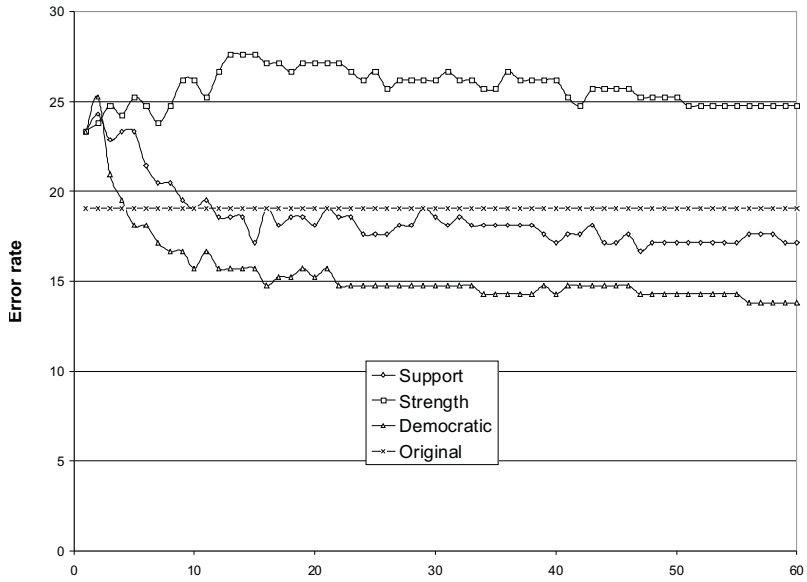


Fig. 1. Image segmentation

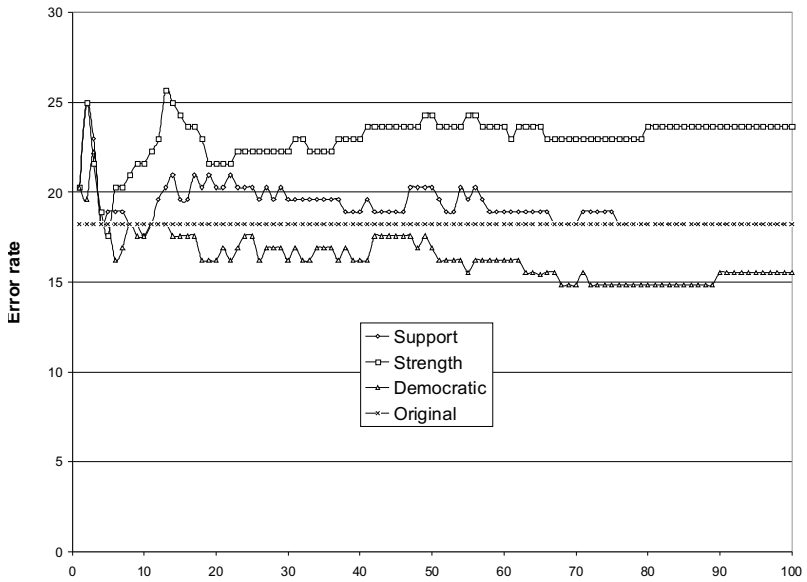


Fig. 2. Lymphography data set

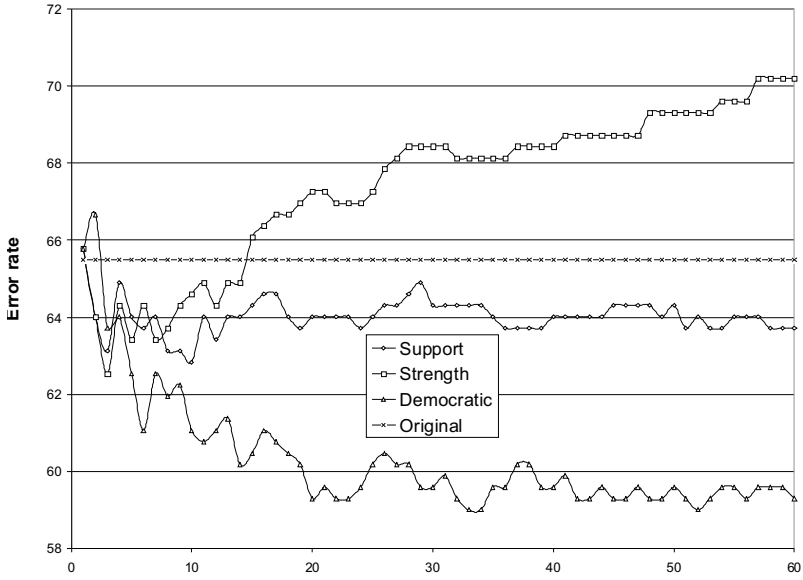


Fig. 3. Primary tumor data set

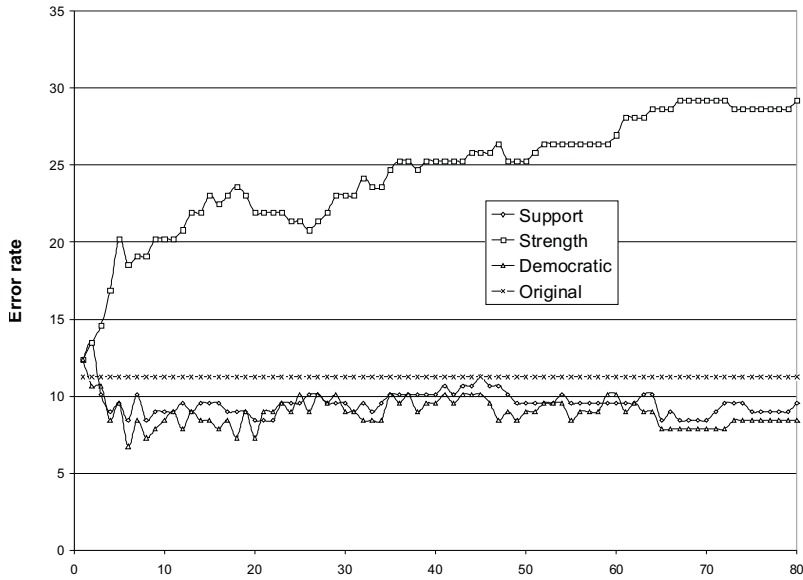


Fig. 4. Wine recognition data set

used by E. Altman to predict a bankruptcy of companies. Some of these data sets: *BUPA*, *Glass identification*, *Hepatitis*, *Image segmentation* and *Pima* were pre-discretized using the discretization method based on agglomerative cluster analysis [19].

Results of our experiments, for the size of the ensemble equal to 100, are presented in Table 1. All of these error rates are results of ten-fold cross-validation. The column *Original data set* in Table 1 represents the error rates, computed using 10-fold cross-validation, for the original data sets.

Another important feature of bagging is stability. When the size of the ensemble increases, starting from some number, the error rate becomes approximately constant, see Figures 1–4. It is clear that ensemble voting based on strength is not as stable as remaining two voting methods, again, see Figures 1–4. For the symbolic data set all three ensemble voting methods were more stable.

During our experiments all ensembles were growing incrementally, with adding a new rule set to the existing rule sets. All three methods of ensemble voting were conducted on the same ensembles (the other possibility would be to create a sequence of ensembles separately for every voting method).

6 Conclusions

Results of our experiments show that ordinary, democratic voting is—in the most cases—the best approach to ensemble voting. However, the Wilcoxon matched-pairs signed rank test shows that the democratic voting is not significantly better than voting based on support (two-tailed test, 5% significance level). On the other hand, voting based on strength is the worst approach to ensemble voting among our three explored methods.

For some data sets, such as *Bankruptcy* and *Iris*, bagging did not improve error rate. Most likely, it happens for high quality data sets.

In our experiments all rule sets were *certain*, i.e., they were induced from concept lower approximations. We are planning to compare certain rules with possible rules combined with bagging, in the future.

Acknowledgement. The NNSAs Kansas City Plant is operated by Honeywell Federal Manufacturing & Technologies under Department of Energy Contract No. DE-AC04-01AL66850.

References

1. Breiman, L.: Bagging predictors. *Machine Learning* 24, 123–140 (1996)
2. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: *Proceedings of the 13th International Conference on Machine Learning*, pp. 148–156 (1996)
3. Breiman, L.: Random forests. *Machine Learning* 45, 5–32 (2001)
4. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Machine Learning* 36, 105–139 (1999)

5. Blaszczynski, J., Stefanowski, J., Zajac, M.: Ensembles of abstaining classifiers based on rule sets. In: Proceedings of the International Symposium on Foundations of Intelligent Systems, pp. 382–391 (2009)
6. Kuncheva, L.I.: Combining Pattern Classifiers. Methods and Algorithms. John Wiley & Sons, Hoboken (2004)
7. Stefanowski, J.: The bagging and n^2 -classifiers based on rules induced by MODLEM. In: Proceedings of the Fourth International Conference on Rough Sets and Current Trends in Computing, pp. 488–497 (2004)
8. Stefanowski, J.: On combined classifiers, rule induction and rough sets. Transactions on Rough Sets 6, 329–350 (2007)
9. Zenko, B., Todorovski, L., Dzeroski, S.: On comparison of stacking with MDTs to bagging, boosting, and other stacking methods. In: Proceedings of the ECML/PKDD 01 Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning, pp. 163–175 (2001)
10. Wolpert, D.: Stacked generalization. Neural Networks 5, 241–260 (1992)
11. Gama, J.: Combining classifiers by constructive induction. In: Proceedings of the 10th European Conference on Machine Learning, pp. 178–189 (1998)
12. Hall, L.O., Bowyer, K.W., Banfield, R.E., Bhadoria, D., Kegelmeyer, W.P., Eschrich, S.: Comparing pure parallel ensemble creation techniques against bagging. In: Proceedings of the IEEE International Conference on Data Mining, pp. 533–536 (2003)
13. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Machine Learning 40, 139–157 (2000)
14. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
15. Chan, C.C., Grzymala-Busse, J.W.: On the attribute redundancy and the learning programs ID3, PRISM, and LEM2. Technical report, Department of Computer Science, University of Kansas (1991)
16. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. Fundamenta Informaticae 31, 27–39 (1997)
17. Grzymala-Busse, J.W.: MLEM2: A new algorithm for rule induction from imperfect data. In: Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 243–250 (2002)
18. Holland, J.H., Holyoak, K.J., Nisbett, R.E.: Induction. Processes of Inference, Learning, and Discovery. MIT Press, Boston (1986)
19. Chmielewski, M.R., Grzymala-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. International Journal of Approximate Reasoning 15(4), 319–331 (1996)

Simplified Self-Adapting Skip Lists

Jonathan J. Pittard and Alan L. Tharp

North Carolina State University School of Engineering, Dept. of Computer Science
Raleigh, North Carolina, USA
{jjpittar, alan_tharp}@ncsu.edu

Abstract. The Simplified Self-Adapting Skip List, a practical new extension of the Skip List data structure, is designed for use with data that exhibit bias, that is, a nonuniform distribution of queries to set elements. The structure observes an initially unknown degree of bias in queries to a data set and adapts itself to a consistently nearly-optimal configuration, improving search efficiency and speed. By modifying the original Skip List design in intuitive ways, self-optimization is achieved while maintaining an extreme simplicity of description, implementation, and operation unmatched by previous dynamic Skip Lists. The additional memory, time, and conceptual overheads introduced by this structure over the original Skip List are considerably less than in previous dynamic designs, but search speed is comparable or superior, making the SSASL better suited than its predecessors for operations in which time or memory efficiency is critical.

1 Introduction

The problem of rapid, efficient data access is one of the most fundamental to the discipline of Computer Science. Traditionally, attempts to advance solutions to this problem have targeted either the development of new hardware which can execute existing access techniques more quickly or the development of new data structures which can retrieve data with fewer accesses (probes) of existing hardware. With the boundary of the speed of light forcing the continual miniaturization of computer hardware as the demand for speed increases, it stands to reason that approaches in the first category using modern electronics will, or perhaps have already begun to, plateau. Despite this, the need for increased data access speed has not waned, making the development of faster, more efficient data structures now just as important as ever.

Numerous approaches exist for the rapid and efficient access of data. Tree structures are particularly well suited to data sets which may expand over time, and are effectively implemented as solutions for problems related to the searching of sorted lists. However, many tree solutions, such as IPR Trees, B-Trees and their variants, and Red-Black Trees, either provide sub-optimal performance or require balance mechanisms which can be expensive in execution and complicated in implementation. [8][9] Pugh introduces the Skip List, a probabilistic data structure which performs competitively with modern trees while avoiding

the need for erudite balance operations. [1] The extreme simplicity of the Skip List is valuable, and distinguishes it significantly from the complicated balanced tree structures to which it is an alternative. This simplicity is sufficient to warrant the presentation of Skip Lists in many introductory-level Data Structures texts. [7] In contrast, many of the state-of-the-art trees to which Skip Lists are a suitable alternative can be nearly incomprehensible without considerably more academic progress.

The issue of data structure efficiency becomes far more complex when considering data sets that exhibit *bias*; that is, sets in which certain set elements are accessed more frequently than others. Many real-world applications, ranging from packet routing to Web caching to general commercial databases, benefit from structures which exploit bias to organize information and access it more quickly. The problem takes on still more complexity for data with query distributions wherein the character or degree of bias changes over time. Current structures for effectively managing such data either fail to exhibit the efficiency of the Skip List, as with a dynamic bias-ranked linked list, or, as with dynamic biased tree solutions similar to the above, employ abstruse balancing operations.

2 Relevant Prior Work

A Skip List [1] is a data structure comprised of nodes arranged in a predetermined number of sorted doubly-linked lists called *levels*. Every Skip List node is included in level 0, enabling easy sequential access to data elements. Each subsequently higher level L contains a subset of the elements in the level below it. A *level x node* is found in all levels $L \leq x$. Each node contains its data element along with *previous* and *next* pointer arrays equal in size to the number of levels in the Skip List. Pointers in these arrays position each node within the level equal to the pointer's array index. That is, level L is a list of elements linked by the `previous[L]` and `next[L]` pointers of its constituent nodes. A node's pointer array elements for levels in which it is not found are null. All levels begin with the *head node* and end with the *tail node*. These have no data and the respective keys $-\infty$ and ∞ . Inclusion of a node in a given level of a Skip List is determined probabilistically. Upon insertion, a random number generator is consulted so that each node has an independent probability P of being promoted to each subsequent level. A general process for building a Skip List is as follows:

```

1 For each node, L=0:
2 While((New_Random_Number/Random_Number_Range < P) OR L==0):
3   Insert node into level L using sorted doubly-linked list
   insertion with pointers of index L in pointer arrays
4   If L<List_Top_Level, Loop to (2), Incrementing L

```

For a sorted list of nodes, this operation is $O(n)$. A search begins at the head node on the top level and follows next pointers successively. If a node with a key value greater than the sought value is found, then the search traverses back, and then forward once more on the next lower level. This continues until either

the sought node is found or a node with a key value greater than sought is encountered on level 0 Traversal of higher levels produces efficiency gains by “skipping” lower-level elements. An overview of the search procedure follows.

Beginning at head node on top list level L:

```

1 While Stored_node_key<Sought_key
2   Follow node's next[L] pointer.
3   If Stored_node_key==Sought_key, SUCCESS
4   If Stored_node_key>Sought_key
5     If L==0, FAILURE
6     Else
7       Follow node's back[L] pointer.
8       Decrement L
9   Loop from (1)

```

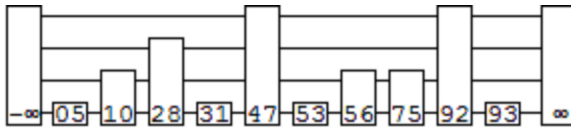


Fig. 1. A Skip List. Vertical boxes are nodes, horizontal lines are links between nodes on various levels, and node keys are listed inside respective nodes. The maximum level of each node is randomly determined. Note how traversal of nodes on higher levels effectively "skips" lower-level nodes.

To insert a new datum, the procedure for list creation is used for the new element's node in isolation. To delete an element, one searches for the target node and deletes it from list levels sequentially. Deletion from a single level L is doubly-linked list deletion using array elements of index L in nodes' pointer arrays. Search, insertion, and deletion are all $O(\log(n))$. Figure 1 illustrates one possible design for a standard Skip List.

An optimal Skip List for biased data is one in which all *level x nodes* have access probabilities greater than than all *level x-1 nodes*, but less than all *level x+1 nodes*. This arrangement minimizes search time by ensuring that as few traversals as possible are made during the average search. Because items are assigned to list levels at random, however, it is unlikely that the Skip List creation procedure will provide this optimal arrangement. The task of all Skip List designs for use with biased data is therefore to speed search by modifying list arrangement, structure, or procedures to enforce the above description of optimality. For data with dynamic bias, this is accomplished by rearranging nodes through the life of the list. Unfortunately, current proposals to achieve this goal incur undue overhead relative to the original Skip List design in terms of memory usage, code base size, implementation difficulty, and counts of executed instructions and memory probes during search. Search overhead is critical, as rapid, efficient

search is the primary purpose of all Skip List structures. Moreover, simplicity is one of the hallmarks of the original Skip List design. Current dynamic Skip Lists do not exhibit even a shadow of corresponding simplicity, demonstrating that this advantage has been lost as the structure has been extended.

In 1994, Martine and Roura defined an approach to handling query bias using Skip Lists, but did not address bias which is dynamic or initially unknown.^[6] In 2001, Ergun *et al.* proposed a “Biased Skip List” which provides list optimization in a *move to front* context.^[2] They also partially developed a *most frequently accessed* model. This highly contextual work is somewhat generalized in ^[3]. The design, similar in both papers, is functional, but incurs significant overhead due to the elaborate way in which it maintains list optimality. Later, Bagchi *et al.* discuss another form of dynamic “Biased Skip List.”^[4] The design is more general than that described by Ergun, applicable to all forms of bias, and mathematically robust. However, while theoretically optimal, the design suffers in practice from many shortcomings similar to those in ^[3]. Most problematic for the Bagchi design in particular is its degree of conceptual difficulty, which far exceeds that of both the original Skip List and the Ergun design. Finally, in 2008 Bose *et al.* build on the work of Bagchi *et al.* to define another dynamic Skip List for biased data.^[5] The technique partially resolves efficiency issues, and is easier to comprehend and implement than Bagchi’s design. Unfortunately, the optimization mechanisms of the structure are considerably more complicated than necessary, resulting in extraneous overhead similar to previous designs.

3 Contribution

The new Simplified Self-Adapting Skip List was designed directly from the original Skip List data structure. As it takes nothing from earlier dynamic designs, the new structure avoids many problematic issues present in earlier methods. The only additions to a Skip List necessary to form a SSASL¹ are a `query_count` field in each node and three intuitive procedures, `promote`, `demote`, and `optimize`. `Promote` inserts a *level x* node into level $x+1$. `Demote` deletes a node from the top level in which it is found. Both operate in $O(\log(n))$.

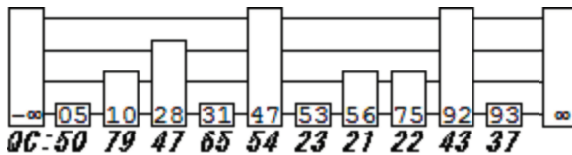


Fig. 2. A Simplified Self-Adapting Skip List, built from the Skip List in Figure 1, before optimization. Note how the Skip List in Figure 1 and the SSASL in Figure 2 exhibit nearly identical structure. Query counts are listed below each respective node in bold.

¹ Simplified Self-Adapting Skip List.

Promote:

- 1 Search for element using Skip List search
- 2 If the element is not found, terminate with error
- 3 If the element is found in level L, insert into the sorted doubly-linked list formed by pointers of array index L+1.

Demote:

- 1 Search for the element using Skip List search
- 2 If the element is not found, terminate with error
- 3 If the element is found in level L, delete from the sorted doubly-linked list formed by pointers of array index L.

Optimize corrects sub-optimal node arrangement by using promote and demote to “swap” nodes from adjacent levels. This “swapping” preserves the size of list levels and thus $O(\log(n))$ search without complex restructuring of the list. A major problem with current dynamic biased Skip Lists is that complex optimization procedures are executed at the time of search, increasing the memory access cost of the search operation and thus slowing search execution. Additionally, to accomplish search-time optimization, current designs must constantly maintain sizable and complex support structures. The very existence of such structures increases the memory footprint of dynamic Skip Lists considerably over that of the original Skip List. To solve these problems, the SSASL’s optimize is called at non-critical times when searches are not performed. Delaying optimization until times when speed is not crucial frees the search operation from nearly all optimization-associated costs. This allows for fast and efficient search with far less overhead than in earlier designs. In fact, SSASL search requires only one more memory probe than search of a standard Skip List. Delaying optimization does mean that a SSASL will not be perfectly optimal at all times. However, savings gained in the form of reduced search overhead offset the cost of searches to a marginally sub-optimal list. The net result is faster searches than with a constantly-optimized list in all but a very limited set of cases, a smaller memory footprint, and a more easily understood and implemented structure. Improvements to the optimize implementation below are possible at the expense of conceptual complexity.

- 1 For each level < the top level of the list, beginning at L=0:
- 2 Locate the level L node with highest query_count value.
- 3 Locate the level L+1 node lowest query_count value.
- 4 If(query_count of node in (2) > query_count of node in (3))
- 5 Promote node from (2)
- 6 Demote node from (3)
- 7 Set current level to minimum(L-1, 0)
- 8 Else
- 9 Set Current Level to L+1
- 10 Loop to step 1
- 11 For each node in the SSASL:
- 12 Modify query_count

Other operations to a SSASL are nearly identical to operations in a standard Skip List. Deletion from the list is identical. Search is identical, with the exception that a located node's `query_count` must be incremented. For data with initially unknown bias, SSASL creation and element insertion are identical to the comparable Skip List operations, save that `query_counts` are initialized to 0. If initial node access probabilities are known, then `query_counts` and node arrangement may be pre-configured accordingly. However, for most applications of the SSASL, initial query probabilities are expected to be unknown.

One feature of `optimize` critical to obtaining quality list optimization is the modification of `query_count`. `Query_count` modification, usually reduction, allows an element's past popularity to influence its position in the list while preventing historical trends from dominating recent bias observations. Ideal implementation is highly application-specific, but integer division of `query_count` by some floating-point $A > 1$ appears effective. Higher A values make list organization more responsive to changes in bias, while lower A values limit the impact of temporary bias changes, preventing frivolous reorganization. More sophisticated implementations of `query_count` modification are certainly possible. Finally, note that as list optimality can degrade between calls to `optimize`, the frequency of `optimize` calls, relative to the number of queries serviced by the list, should correlate with the rate of change in data bias. Ideal optimization frequency is highly application specific.

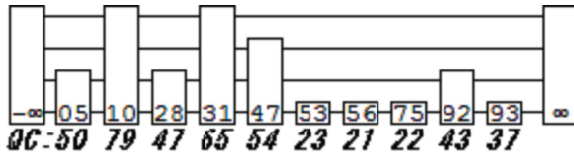


Fig. 3. The Simplified Self-Adapting Skip List from Figure 2 after optimization. Note how the number of queries to a node now correlates directly with that node's maximum level. Levels have not changed size.

4 Caveats

Three caveats exist to the practical use of the Simplified Self-Adapting Skip List. Least important is the issue of `query_count` overflow. In practice, if `query_count` is implemented as a 64-bit field, then overflow would occur only after $2^{64} - 1$ accesses to a single record. Even an application which did not modify `query_count` in `optimize` and received one million queries per record between `optimize` calls could operate through trillions of runs (spans of time between optimization) before an overflow. If `query_count` is implemented as a more narrow field, effective reduction in `optimize` is necessary to avoid overflow. The second issue facing the SSASL is trivial due to its rarity. If no non-critical times exist at which `optimize` may be called, then the SSASL is not applicable. However, this should only be an issue for a very limited range of applications, leaving an extremely wide domain to which the SSASL may contribute.

The SSASL provides large savings over a standard Skip List and performs well relative to older dynamic Skip Lists even in the face of degraded list optimality. The third caveat to the advantages of the SSASL is that this is true only when such degradation is not extreme. “Extreme” degradation is a situation in which query bias changes between `optimize` calls to a point where a large fraction of queries are for elements which would be relocated by a moderate fraction of list levels if `optimize` were invoked immediately. In such cases, performance gains of the SSASL can be expected to be limited due to the increased number of node traversals in the average search. Change in bias for elements comprising a small portion of queries which would cause relocation by a significant fraction of levels does not degrade performance significantly. Change necessitating the relocation of nodes comprising a large fraction of queries by a small fraction of levels is also not significantly damaging. This problem can be solved by increasing the frequency of `optimize` calls, thereby decreasing the number of queries serviced by drastically sub-optimal lists, but this may not be practical. The SSASL is therefore not suited for situations where `optimize` cannot be called frequently enough to accommodate major bias changes.

5 Results and Comparisons

The Simplified Self-Adapting Skip List exhibits extreme savings in terms of memory access counts over the original Skip List when used to service significantly biased queries. Approximately 89.77% savings is observed in the ideal case for a list of one million elements. “Ideal case” is bias sufficient that the vast majority of queries are for few enough elements that all such elements can be relocated to the top list level. Significant savings are still achieved for data in which query distribution skew is less prevalent, as documented in Table 1. Even in data with no bias, SSASL performance less than 5% worse than that of a standard Skip List. The SSASL is therefore suitable for use in applications in which query bias is not guaranteed. Previous dynamic Skip Lists have incurred such overhead as to make them uncompetitive with the standard Skip List without very significant bias.

Relative to existing dynamic Skip List designs, the SSASL also performs favorably. Although current designs maintain optimality at all times, they do so at the cost of large overheads of numerous forms. The SSASL eliminates many of the shortcomings of previous methods and drastically cuts all forms of overhead at the cost of providing optimization which is not constantly maintained. This yields net gains in search speed over previous implementations in all but the limited caveat cases. Additionally, the SSASL is far less difficult than previous designs to understand and implement, and should exhibit a smaller code base and memory footprint, making it preferable for reasons other than speed.

The Biased Skip List designed by Ergun *et al.* [3] contrasts sharply with the simple efficiency of the SSASL. Ergun’s structure stores a ranked list of nodes in memory at all times, which is consulted and maintained during optimization, increasing search-time probe counts. Additional data are necessary create the concept of level “class,” which plays a role in optimization calculations. The structure must also define the boundaries of each class and determine the

appropriate classification of every data node. When a search is performed, supporting ranking and classification data related to potentially many nodes must be updated. Finally, updating the structure's Skip List alone is significantly more costly than in the SSASL due to a far more complicated level structure. Insertion and deletion are also more complicated, and can require extensive restructuring of node classes. Unlike Ergun's design, the SSASL requires only one additional datum per node during search relative to the standard Skip List, resulting in a dramatically smaller memory footprint than any previous dynamic Skip List design. In addition, the only search-time memory access beyond those in the standard Skip List is the incrementation of `query_count`. No optimization costs are incurred at the time of search, yielding substantial savings over Ergun's method. Finally, the only complex concept beyond those of the standard Skip List necessary for the SSASL is the `optimize` procedure, which is far more basic than the optimization in the design of Ergun *et al.*

The 2004 Bagchi *et al.* structure^[4] could potentially resolve the time and memory efficiency issues of the Ergun design. However, while the Bagchi paper is mathematically indisputable, many details necessary for implementation of the structure are omitted. Efficiency will therefore vary based on implementation decisions. The design, however, does require node relocation as part of its `reweight` procedure, and indicates that re-weighting of nodes is expected on a per-search basis. Thus, even in the best implementation, optimization costs will be incurred during searches, causing greater overhead and slower search than in the SSASL. Bagchi's design also defines several invariants and restrictions on the configuration of the underlying Skip List which must be constantly enforced. Additional search overhead is incurred to maintain these conditions. Most critical for the design, however, is that it exhibits a conceptual complexity which dwarfs that of the original Skip List and even well eclipses the complexity of the other dynamic designs. Pugh's Skip List is sufficiently simple to be covered in introductory data structures texts,^[7] but both Bagchi's technique and the paper in which it is presented are best suited for readers with a very fresh and highly advanced knowledge of algorithmics. The work is sound, but may be unintelligible to many developers who could benefit from the design. In contrast, the SSASL aims to be as simple as possible, bringing to dynamic Skip Lists the simplicity which Pugh heralded as an advantage of the original Skip List design.

Finally, the design by Bose *et al.*^[5] builds on Bagchi's work, inheriting its overheads, complexities, and restrictions on list structure. Additionally, optimization involves the concept of "layers," not found in the original Skip List design, and consists of manipulating these layers in conjunction with numerous sublists which complicate design of Skip List levels. The result is a complex procedure in which the sizes of layers or levels may overflow prescribed bounds and protracted restructuring may be necessary to restore design invariants. As optimization is performed during search, all overhead associated with this procedure slows search execution. In contrast, the SSASL provides optimization which is intuitively simple, with negligible search overhead, and imposes no new structural requirements or concepts upon the original Skip List.

Table 1. Near-worst-case probe counts for a SSASL of 1M nodes. Average query depth correlates directly with degree of bias, with no bias at 500,000 average depth. A comparable standard Skip List averaged 28.41 probes per query.

Average Query Depth	SSASL Probes	Improvement %
100	9.91	65.12%
500	14.41	49.28%
1000	15.91	44.00%
5000	18.90	33.47%
10000	20.40	28.19%
100000	24.81	12.67%

Despite the problems with previous designs, readers may wonder why other approaches are not more efficient than the Simplified Self-Adapting Skip List. Earlier designs update node arrangement with each search, maintaining optimality at all times. A SSASL, however, is at least slightly sub-optimal at most times. Readers may surmise that savings over the SASSL experienced because of list optimality should offset the cost of constantly optimizing. Unfortunately, two principal sources in which current dynamic Skip Lists are developed [4] [5] provide no experimental results, and so metrics for comparison of the techniques are not readily available. Ergun [2] [3] does provide results, but only in terms of execution time on now-obsolete hardware. Implementing the techniques independently is only a partial solution. Bagchi’s work [4] focuses almost exclusively on mathematical definition. Practical implementation of the structure would therefore involve subjective design decisions, confounding any test results. This is true to a lesser extent of the work of Bose [5], and makes objective comparison with the SSASL design particularly difficult. Finally, the primary concern of Dynamic Skip Lists to date [3] [5] has been *move-to-front* optimization, with which this work is not concerned. Nonetheless, tests based on data and results from Ergun [2] [3] have displayed a greater speed increase over the standard Skip List with the SSASL than with the earlier design. Significant gains over the Skip List were also observed with the SSASL for queries exhibiting less bias than in any earlier tests.

Though limited, numerical comparisons of one earlier design to the new SSASL do demonstrate the effectiveness of the new design. Where comparison data are not available, analysis demonstrates that the SSASL performs either competitively or superiorly to earlier designs in many cases. The additional overhead of a SSASL search beyond that of standard Skip List search is only one probe. It is inconceivable that previous dynamic Skip Lists could perform optimization after every search without more average overhead than this. Minimal pointer operations alone amount to more overhead than the in the SSASL design, neglecting all probes for node rank comparison, supporting data modification, and the optimization of modified Skip List levels. All such tasks are considerably expensive in current dynamic Skip Lists. Further, the high frequency of updates necessary to maintain perfect optimality prevents amortization of overhead across any large number of searches. The Skip List’s efficiency makes the mild to moderate

degradation of list optimality found in the SSASL much less important than the relatively large overhead of constant optimization found in other designs. With even one probe, an efficiently-implemented Skip List can traverse an entire level of nodes. Thus, even if the overhead of optimizing a list were fewer than five probes per search on average (and analysis of current techniques finds it to be much higher) then a Skip List could traverse and search multiple levels with fewer probes than required for optimization. While constant optimization may yield higher efficiency than searching an entirely unoptimized list, a Skip List can accommodate a degree of sub-optimality at much lower cost than that of maintaining perpetual, perfect optimality. Given the complexity of the optimization activities needed in earlier methods for every search, the SSASL can be expected to search more quickly than previous dynamic Skip Lists even if list bias has changed somewhat since optimization.

6 Conclusion

The Simplified Self-Adapting Skip List is a new, independently-developed improvement upon the Skip List data structure designed for use with data sets to which the bias of queries is nonuniform and initially unknown. The SSASL provides optimization at non-critical times to conform to observed bias in list queries. By building directly upon the original Skip List, the SSASL endeavors to retain the simplicity which Pugh highlighted as a defining factor of his structure. While it is one of Skip Lists' major advantages, this simplicity appears to have been lost in the development of dynamic Skip List designs. Previous efforts at dynamic Skip Lists, while effective, have created structures which are less efficient, slower, more complicated, and more difficult to understand and implement than necessary. Even though the Simplified Self-Adapting Skip List is not always perfectly optimized, search is usually faster than in earlier dynamic Skip List approaches due to reduced search-time costs. This analysis is supported by all the limited available experimental data for dynamic Skip Lists as well as by analysis of dynamic Skip List designs. Thus, the SSASL is often preferable over other designs due to its reduced memory footprint, reduced executed instruction and memory access counts in search, and lower conceptual and implementation difficulty. It is hoped that the advantages of the Simplified Self-Adapting Skip List open the area of dynamic Skip Lists to wider understanding and application while providing results superior or competitive both to previous dynamic Skip Lists as well as other techniques for handling dynamically biased data sets.

References

1. Pugh, W.: Skip lists: A probabilistic alternative to balanced trees. *Comm. of the ACM* (1990)
2. Ergun, F., Mittra, S., Sahinalp, S., Sharp, J., Sinha, R.: A Dynamic Lookup Scheme for Bursty Access Patterns. In: *Proc. IEEE INFOCOM* (2001)

3. Ergun, F., Sahinalp, S., Sharp, J., Sinha, R.: Biased skip lists for highly skewed access patterns. In: Buchsbaum, A.L., Snoeyink, J. (eds.) ALENEX 2001. LNCS, vol. 2153, p. 216. Springer, Heidelberg (2001)
4. Bagchi, A., Buchsbaum, A., Goodrich, M.: Biased skip lists. *Algorithmica* 42 (2005)
5. Bose, P., Douieb, K., Langerman, S.: Dynamic optimality for skip lists and B-trees. In: Proc. of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (2008)
6. Martinez, C., Roura, S.: Optimal and nearly optimal static weighted skip lists. Technical report LSI-95-34-R. Dept. Llenguatges i Sistemes Informatics Universitat Politcnica de Catalunya (1995)
7. Sahni, S.: Data Structures, Algorithms, and Applications in Java, 2nd edn. Silicon Press (2005)
8. Tharp, A.: File Organization and Processing. Wiley, Chichester (1988)
9. Guibas, L., Sedgewick, R.: A dichromatic framework for balanced trees. In: Proc. 19th IEEE Symp. on Foundations of Computer Science (1978)

Multi-Agent Architecture with Support to Quality of Service and Quality of Control

Jose-Luis Poza-Luján, Juan-Luis Posadas-Yagüe, and Jose-Enrique Simó-Ten

University Institute of Control Systems and Industrial Computing (ai2)
Universidad Politécnica de Valencia. Camino de vera, s/n. 46022 Valencia, Spain
{jopolu, jposadas, jsimo}@ai2.upv.es

Abstract. Multi Agent Systems (MAS) are one of the most suitable frameworks for the implementation of intelligent distributed control system. Agents provide suitable flexibility to give support to implied heterogeneity in cyber-physical systems. Quality of Service (QoS) and Quality of Control (QoC) parameters are commonly utilized to evaluate the efficiency of the communications and the control loop. Agents can use the quality measures to take a wide range of decisions, like suitable placement on the control node or to change the workload to save energy. This article describes the architecture of a multi agent system that provides support to QoS and QoC parameters to optimize de system. The architecture uses a Publish-Subscriber model, based on Data Distribution Service (DDS) to send the control messages. Due to the nature of the Publish-Subscribe model, the architecture is suitable to implement event-based control (EBC) systems. The architecture has been called FSACtrl.

Keywords: Multi Agent System (MAS), Quality of Service (QoS), Quality of Control (QoC), Publish Subscriber model, Event Based Control (EBC).

1 Introduction

The optimal control of distributed systems has changed from the systems based on bus-oriented communications to the large industrial systems based on computer networks. The current trend joins all aspects of distributed systems with intelligent control in concept known as cyber-physical systems [1].

Systems must be optimum to carry out of the objectives fixed. For optimize a system is necessary to have the suitable information about which are the features that have more influence on throughput. The information about communication performance is known as quality of service (QoS). Information about the compliment of the control requirements is included in the concept of quality of control (QoC). To manage system performance, the architecture must provide to agents all necessary information to build their own quality indicators. One interesting question is, if the quality indicators can be used to take the usual decisions used in the distributed systems, for example move or clone an agent between two control nodes.

There are a lot of agent-based protocols, middleware and architectures. The treatment of the QoS is different depending on the standard used. The Common

Object Request Broker Architecture (CORBA) defines the QoS by means the concept of messaging policy. CORBA defines 14 policies to cover the basic time, order and routing aspects [2]. The Foundation for Intelligent Physical Agents (FIPA) defines 14 QoS policies mainly in terms of speed and reliability [3]. The Data Distribution Service model DDS [4] specification proposes 22 different QoS policies that cover all aspects of communications management: message temporal aspects, data flow and metadata.

To use the different points of view of the QoS with the system's QoC is necessary to have a uniform method to obtain the necessary QoS parameters. A multi-agent architecture, called Frame-Sensor-Adapter with Control support (FSACtrl), has been developed to provide to control components the QoS parameters. This architecture is the evolution of the model FSA, widely tested on mobile robot, and home automation systems [5].

The communication system on FSACtrl is based on the DDS model and can supply the DDS QoS policies and, for extension, the FIPA and CORBA policies. In FSACtrl, the control algorithms are implemented on the control components, this control components are very similar than the communications components. FSACtrl is characterized by its simplicity in the implementation of components, the efficient management of QoS and QoC parameters and the minimization of protocol operations. It makes possible to distribute the components in an efficient way.

The rest of the article has been organized as follows. The second section introduces the theoretical concepts of communications that are the base of the architecture shown. The third section describes basic concepts about FSACtrl architecture. The fourth section shows the main idea of the article, based on the cycle of QoS and QoC. Finally, concluding remarks about architecture are shown and possible research lines that emerges from the combination of QoS and QoC.

2 Theoretical Concepts

2.1 The DDS Model

There are different paradigms of communication with support to quality of service, among them publish-subscribe model is one of the most suitable [6] due that isolates publishers of subscribers, enabling a QoS negotiation based on the information topics. The agents only need to know the topics to send or receive the information, without knowing the current location of the other agents.

Object Management Group (OMG) has proposed DDS, based on the paradigm of publish-subscribe with support to QoS. Data Distribution Service (DDS) provides a platform independent model that is aimed to real-time distributed systems. DDS is based on publish-subscribe communications paradigm. Publish-subscribe components connect information producers (publishers) and consumers (subscribers) and isolate the publishers and the subscribers in time, space and message flow [7].

DDS specifies two areas: Data-Centric Publish-Subscribe (DCPS), which is responsible for data distribution, and Data Local Reconstruction Layer (DLRL) which is responsible for adjusting the data to local level of applications. DLRL area is optional due to the DCPS components can work directly with the control objects without data translations.

When a producer (component, agent or application) wants to publish some information, should write it in a Topic by means of a component called Data Writer which is managed by another component called Publisher. Both components, Data Writer and Publisher, are included in another component called Domain Participant. On the other hand, a Topic can deliver messages to both components: Data Readers and Listeners by means of a Subscriber. Data Reader provides the messages when the application requires. Instead of a Listener sends the messages without waiting for the application.

Quality of Service is defined as the collective effect of service performance, which determines the degree of satisfaction of a user of the service [8]. The concept of QoS is used to measure all relevant characteristics of a system. Generally, QoS is associated with a set of measurable parameters. In DDS model, QoS policy can be defined as the dynamic management of the QoS parameters with negotiated values. For example, by means the “Deadline” policy, that determines the maximum time for the message arrival, and the “Time-Based-Filter” policy, that determines the minimum time between two messages, a component can establish a temporal window to receive messages from other components.

2.2 Event-Based Control and Quality of Control

As control system complexity grows, timing requirement became difficult to be complied; therefore, the time-driven based control approach (TBC) is not the most efficient model. The Event Based Control (EBC) [9] complements the TBC model decreasing the messages needed to receive data and send control commands.

In the EBC model, messages between sensors, controllers and actuators, are only sent when an important condition is fulfilled. A wide range of conditions can generate events. The most common condition is related to error between the control action and the value obtained, and related to connection maintenance (keep-alive messages). EBC model requires a communications infrastructure based on events. From agent’s point of view, DDS model provides a system based on events, implemented by means of Publish-Subscribe paradigm in the communications components.

In the same way that to evaluate the efficiency of communications uses QoS parameters; control must provide the corresponding parameters (QoC). It’s considered a good control when the signal is sent to the actuator causes the signal measured by the sensor is identical to a reference signal, therefore there is no error between the measured signal and reference signal. Control error is used to modify the signal sent to actuator. The most commonly used QoC parameters are the value of the Integral Absolute value of Error (IAE) and the Integral of the Time and the Absolute value of Error (ITAE). Both parameters allow the system to know how evolve the error, and predict the new action. In the EBC model, the QoC should include the aspects related with the event management [10] which implies the existence of common parameters with the QoS such as throughput, delay and delay jitter.

3 FSACtrl Architecture

Figure 1 shows the main components of the architecture FSACtrl [11]. Each control node has a manager agent, the necessary control agents, the communications

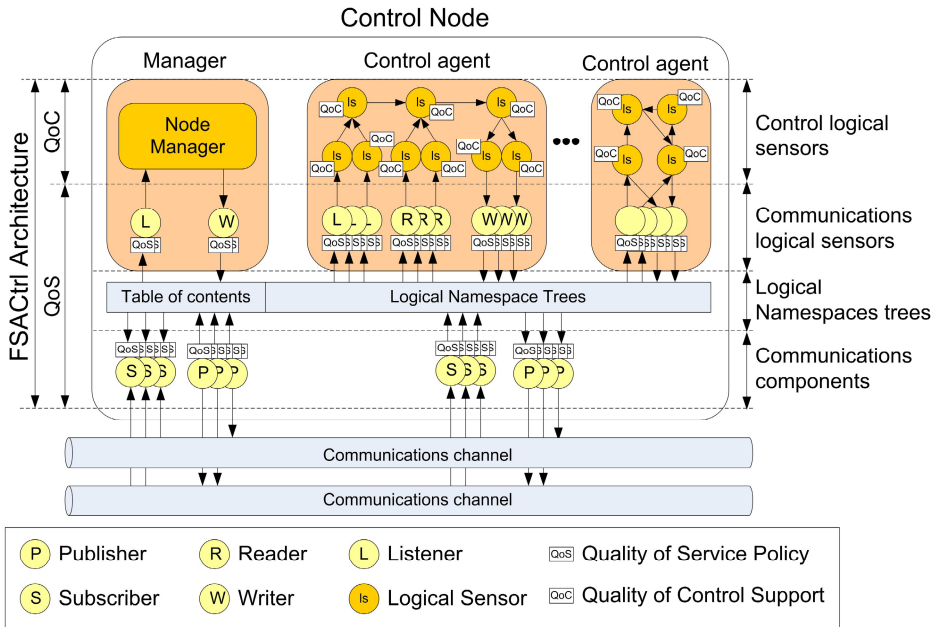


Fig. 1. This figure shows the overview of the architecture FSACtrl with the main communications and control components and connections between them

components to provide support at control agents, and a set of topics to connect the control agents with the communications components, Topics are organized in an ontology called Logical Namespace Tree.

Each control node contains a special ontology called Table of Contents (TOC). TOC describes the control node to the other control nodes of the system. The communications components are the components proposed by the DCPS model of the DDS standard. Publishers and Subscribers are common to all control agents, whereas Data Writers, Data Readers and Listeners are exclusive to each control agent.

Control algorithms are implemented by the components called Control Logical Sensors that provides the QoC parameters. The logical sensors are grouped hierarchically by means a component called Control Component. Thus the architecture provides the support to hierarchical control.

Manager agent provides support to all functions of the MAS: processes the request of the control agents, both from within and outside the control node, manage the ontology to connect successfully communications components and control components and mediates in the negotiation based on the QoS and QoC parameters.

4 Joining the QoS and the QoC

4.1 QoS and QoC within the Control Nodes

The events of the system arrive to agent by means of the Subscribers. Since the message arrives to agent, it generates a chain of messages that follow a path between

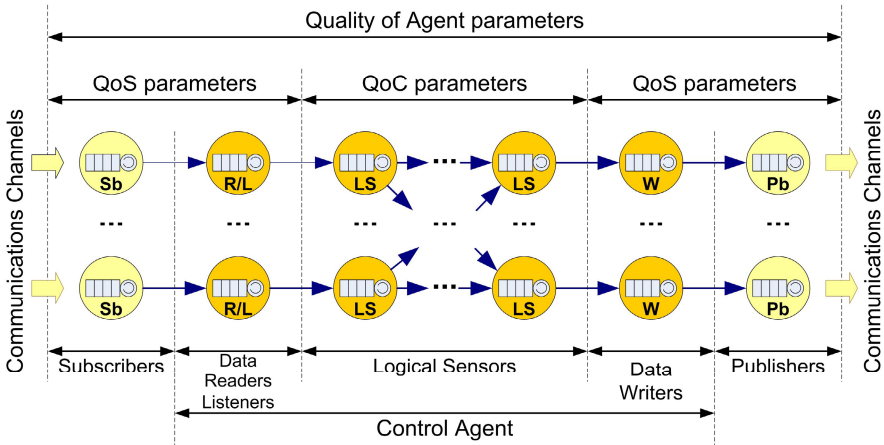


Fig. 2. QoS and QoC location within the FSACtrl components

components. Finally, if is necessary, the agent sends the action by means of the Publishers (figure 2). The QoS parameters are obtained from the connections between the communications elements of the control node (Publishers and Subscribers) and the communications elements from the agent (Data Writers, Data Readers and Listeners). The QoC parameters are obtained from the control components (Logical Sensors) of the agent and its connections. To avoid a deadlock, cycles are not allowed between Logical Sensors.

All communications and control components of the FSACtrl architecture have a unique message queue to manage the incoming messages. Besides, all components have a unique control thread. The control thread contains the control algorithm in the case of Logical Sensor or the communications code in the case of communications components. Incoming and outgoing connections must work according with the QoS and QoC values agreed with the others components. Wearing a unique queue for every atomic component can make the component a bottleneck. To avoid it, the manager agent controls the QoS parameters and can duplicate some components and propose the agent movement to other control node with better conditions.

Since each component can provide parameters of both types (QoS and QoC) combining single parameters is possible to obtain general parameters to measure the QoS or the QoC about the Control Agent or Control Node. These parameters are known as Quality of Agent (QoA) parameters.

4.2 The Quality Request-Offer Cycle

In a Multi-Agent System (MAS) when the communications between the agents is based on a middleware, every agent can works as a server, to offer their services, and as client, to use the services of other agents. Usually, in distributed systems when an agent needs an interchange of information with other agent, is necessary an initial phase to negotiate the values of QoS parameters. Additionally, in EBC systems when an agent needs to change the control action, is necessary a phase to negotiate the new values. Consequently, the values of QoC parameters need to be adjusted.

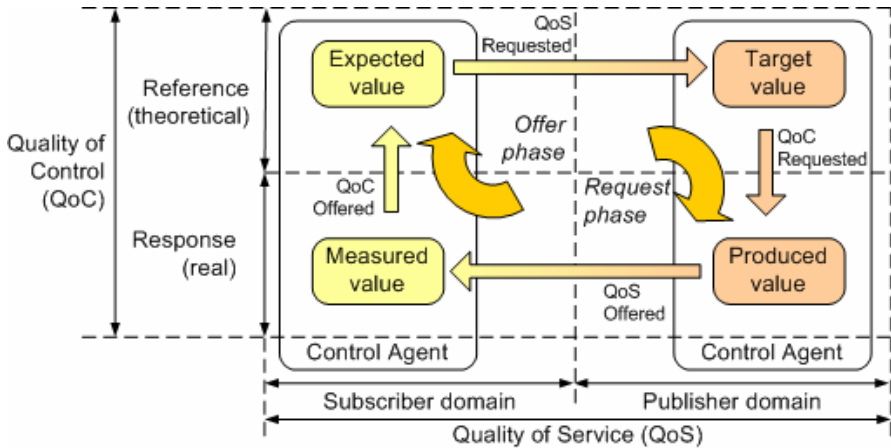


Fig. 3. The QoS and QoC cycle with the request phase and the offered phase

DDS offers a protocol for negotiating the QoS parameters by means the QoS policies. However, when the QoC is included in the negotiation process, it is necessary that the control agent can provide the appropriate protocol and parameters to measure the optimization level. In FSACtrl, the protocol is based in both types: QoS and QoC parameters.

Figure 3, shows the location of the parameters in the control loop. Based on the QoS cycle [12], FSACtrl architecture adds the QoC in the protocol. The QoS parameters are associated with the publish-subscriber communication and the QoC parameters characterize the relationship between the theoretical expectations and the real results of the control actions, both the client and the server side.

As seen in figure 3, the QoS and QoC have a request phase to communicate the value needed of the parameters and the next phase to communicate the values offered. Consequently the cycle consists in two cyclic phases: the QoS and QoC request, and the QoS and QoC offer. This cycle is known as “quality request-offer cycle”. Since a single component, for example a Logical Sensor, to an agent, the cycle runs as a service. The service allows the user to know the values of the formulas that estimates the relation between the QoS and the QoC. By correlating QoS parameters with QoC parameters, the agent is able to decide to prioritize the communications in front of the control or vice versa.

There are two ways to start the cycle: by manager agent’s initiative and by event. When the Manager Agent wants to monitor a specific situation, starts the service in the control agents.

Whenever an agent performs an operation over a specific Control Node (for example, move or clone), the Manager Agent verifies if there is any Control Node in the distributed system that provides the QoS and QoC requested. The quality request-offer cycle is used to determine which Control Nodes are suitable to locate an agent, in the case of insert and move operations.

Figure 4 shows the events that can trigger the quality request-offer cycle. There are two external events, one and four, and three internal events, the rest. The external

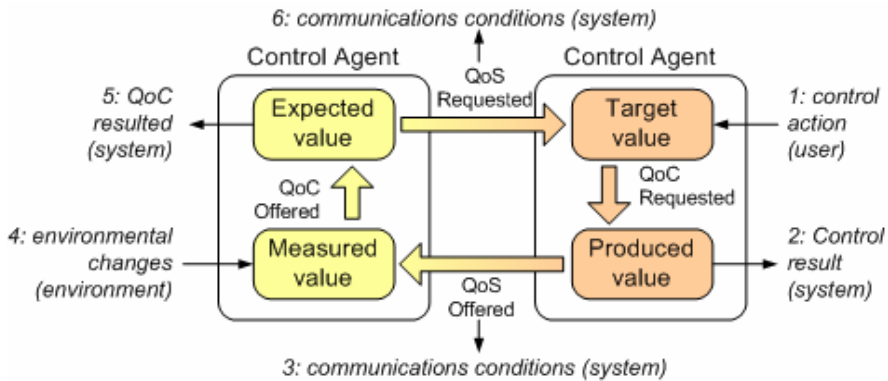


Fig. 4. Events that trigger the quality

triggering can be started by the user when changes the control action (Figure 4, case 1), or by the environmental (Figure 4, case 4), for example when a robot changes the path from the prefixed. The internal events can be located on the control action agent (Figure 4, case 2), on the agent controlled (Figure 4, case 5) and in the communications interchange (Figure 4, case 3 and 6). The cycle defines the actions to take in function of the event triggered. When an event occurs, the adjacent elements related with the event are altered. For example when the user changes the action control, the expected and the produced value changes, so that, the QoC requested and the QoS requested parameters are reviewed. The QoS and QoC changes due to the new control action can need other features of the same data, like an increase in the samples from a sensor, or new information from other control agents, for example new sensors samples.

5 Concluding Remarks and Future Work

This article has shown the need to have QoS and QoC parameters in distributed intelligent control systems. A control paradigm based on events needs the synergy between communications and control. FSACtrl architecture allows QoS and QoC parameters, whatever type of components that implements the agent. Architecture is suitable to implement a distributed intelligent control system based on events.

Currently, the software developed based on FSACtrl architecture is not suitable to implement hard real-time systems because it works in TCP/IP based networks. However, it is very suitable for simulate the behaviour of specific agents topologies in the distributed system in order to optimize global system.

Some research lines related with the FSACtrl architecture are the study of the relations of the QoS and QoC parameters. Other field to work is the development the QoC policies to manage the QoC parameters, in the same way that DDS implements QoS policies to manage the performance of the communications.

Acknowledgements. The architecture described in this article is a part of the coordinated project SIDIRELI: Distributed Systems with Limited Resources. Control Kernel and Coordination. Education and Science Department, Spanish Government and FEDER funds.

References

1. Lee, E.A.: Cyber Physical Systems: Design Challenges. In: 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing, pp. 363–369 (2008)
2. Siegel, J.: CORBA 3: Fundamentals and Programming. OMG (2000)
3. FIPA. FIPA-QoS (2002), <http://www.fipa.org/specs/fipa00094>
4. Object Management Group (OMG): Data Distribution Service for Real-Time Systems, v1.1. Document formal (2005-12-04)
5. Posadas, J.L., Poza, J.L., Simó, J.E., Benet, G., Blanes, F.: Agent Based Distributed Architecture for Mobile Robot Control. *Engineering Applications of Artificial Intelligence* 21(6), 805–823 (2008)
6. Aurecochea, C., Campbell, A.T., Hauw, L.: A Survey of QoS Architectures. *Multimedia Systems Journal, Special Issue on QoS Architecture* 6(3), 138–151 (1998)
7. Pardo-Castellote, G.: OMG Data-Distribution Service: architectural overview. In: *Proceedings of 23rd International Conference on Distributed Computing Systems Workshops*, Providence, USA, vol. 19(22), pp. 200–206 (2003)
8. International Telecommunication Union (ITU). *Terms and Definitions Related to Quality of Service and Network Performance Including Dependability*. ITU-T Recommendation E.800 (0894) (1994)
9. Sánchez, J., Guarnes, M.Á., Dormido, S.: On the Application of Different Event-Based Sampling Strategies to the Control of a Simple Industrial Process. *Sensors* 9, 6795–6818 (2009)
10. Dorf, R.C., Bishop, R.H.: *Modern Control Systems*, 11th edn. Prentice Hall, Englewood Cliffs (2008)
11. Poza, J.L., Posadas, J.L., Simó, J.E.: Middleware with QoS Support to Control Intelligent Systems. In: *2th International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP*, pp. 211–216 (2008)
12. Poza, J.L., Posadas, J.L., Simó, J.E.: From the Queue to the Quality of Service Policy: A Middleware Implementation. In: Omatu, S., Rocha, M.P., Bravo, J., Fernández, F., Corchado, E., Bustillo, A., Corchado, J.M. (eds.) *IWANN 2009*. LNCS, vol. 5518, pp. 432–437. Springer, Heidelberg (2009)

Robust 1-Norm Soft Margin Smooth Support Vector Machine

Li-Jen Chien, Yuh-Jye Lee, Zhi-Peng Kao, and Chih-Cheng Chang

Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, 106 Taiwan
{D8815002,yuh-jye,M9515040,M9415011}@mail.ntust.edu.tw

Abstract. Based on studies and experiments on the loss term of SVMs, we argue that 1-norm measurement is better than 2-norm measurement for outlier resistance. Thus, we modify the previous 2-norm soft margin smooth support vector machine (SSVM₂) to propose a new 1-norm soft margin smooth support vector machine (SSVM₁). Both SSVMs can be solved in primal form without a sophisticated optimization solver. We also propose a heuristic method for outlier filtering which costs little in training process and improves the ability of outlier resistance a lot. The experimental results show that SSVM₁ with outlier filtering heuristic performs well not only on the clean, but also the polluted synthetic and benchmark UCI datasets.

Keywords: classification, outlier resistance, robustness, smooth technique, support vector machine.

1 Introduction

Support vector machines (SVMs) have been proven to be one of the promising learning algorithms for classification [6]. The standard SVMs have *loss + penalty* terms measured by 1-norm or 2-norm measurements. The “loss” part measures the quality of model fitting and the “penalty” part controls the model complexity. In this study, our purpose is to improve original 2-norm soft margin smooth support vector machine (SSVM₂) [9] with robust strategies. First, we find out that the measurement of the 2-norm loss term will amplify the effect of outliers much more than the measurement of the 1-norm loss term in training process. We argue that the 1-norm loss term is better than the 2-norm loss term for outlier resistance. From this robustness point of view, we modify the previous framework in SSVM₂ to a new 1-norm soft margin smooth support vector machine (SSVM₁). We show that SSVM₁ can remedy the drawback of SSVM₂ for outlier effect and improve outlier resistance as well.

Although SVMs have the advantage of being robust for outlier effect [15], there are still some violent cases that will mislead SVM classifiers to lose their generalization ability for prediction. For example, the classification results will be very dissimilar if the difference between the total sum of the hinge losses and

the total sum of the misclassification losses is too large. Hence secondly in this study, based on the design of Newton-Armijo iterations in SSVMs, we propose a heuristic method to filter outliers among Newton-Armijo iterations of the training process and make SSVMs be more robust while encountering datasets with extreme outliers. Our method differs with other methods by truncating hinge loss [10]. It can directly and effectively drop the effect of the outliers.

The rest of the paper is organized as follows: In Section 2, we show how outliers have a great impact on SVMs. Following the idea of SSVM₂, we propose the SSVM₁ in Section 3. In Section 4, we describe how to design the heuristic method for outlier filtering. The numerical results and comparisons are presented in Section 5. Finally, we conclude the paper in Section 6.

2 Review on Soft Margin SVMs and Outlier Effect

We first introduce the standard 1-norm soft margin SVM (SVM₁) and the standard 2-norm soft margin SVM (SVM₂). Then, we argue that the SVM₁ is more robust than the SVM₂ in outlier resistance by observing their primal and Wolfe dual formulations.

Consider the binary problem of classifying m points in the n -dimensional real space R^n , represented by an $m \times n$ matrix A . According to membership of each point $A_i \in R^{n \times 1}$ in the classes +1 or -1, D is an $m \times m$ diagonal matrix with ones or minus ones along its diagonal. Sometimes, we will take the notation y_i as the class label of A_i and the notation x_i as A_i^\top for convenience. The standard 1-norm soft margin and 2-norm soft margin support vector machines are given by the following optimization problems.

1-norm soft margin SVM (SVM₁):

$$\begin{aligned} \min_{(w,b,\xi) \in R^{(n+1+m)}} & \frac{1}{2} \|w\|_2^2 + C \|\xi\|_1 \\ \text{subject to : } & D(Aw + \mathbf{1}b) + \xi \geq \mathbf{1} \\ & \xi \geq \mathbf{0}. \end{aligned} \tag{1}$$

2-norm soft margin SVM (SVM₂):

$$\begin{aligned} \min_{(w,b,\xi) \in R^{(n+1+m)}} & \frac{1}{2} \|w\|_2^2 + \frac{C}{2} \|\xi\|_2^2 \\ \text{subject to : } & D(Aw + \mathbf{1}b) + \xi \geq \mathbf{1} \\ & \xi \geq \mathbf{0}. \end{aligned} \tag{2}$$

The SVMs try to minimize not only the *penalty term* but also the *loss term* in the object function. In the SVM₂ (2), the 2-norm loss term will amplify the outlier effect much more as compared to the 1-norm loss term in the SVM₁ (1). The convex quadratic programs [3] of (1) and (2) can also be transformed into the following Wolfe dual problems by the Lagrangian theory [6].

The dual formulation of SVM₁:

$$\begin{aligned} \min_{\alpha \in R^m} & \frac{1}{2} \alpha^\top D A A^\top D \alpha - \mathbf{1}^\top \alpha \\ \text{subject to : } & \mathbf{1}^\top D \alpha = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, m. \end{aligned} \tag{3}$$

The dual formulation of SVM₂:

$$\begin{aligned} & \min_{\alpha \in R^m} \frac{1}{2} \alpha^\top D(AA^\top + \frac{I}{C})D\alpha - \mathbf{1}^\top \alpha \\ & \text{subject to : } \mathbf{1}^\top D\alpha = 0, \\ & \quad 0 \leq \alpha_i, \quad i = 1, 2, \dots, m. \end{aligned} \quad (4)$$

In the dual form of SVM₂ (4), the constraint, $0 \leq \alpha_i$, is a big cause of the outlier effect, where $\alpha_i = C\xi_i$ (by the optimality conditions). It means that the upper bound of α_i depending on the variable ξ_i is unlimited, and the normal vector, $w = A^\top D\alpha$, will be affected by the unrestricted α consecutively. In the SVM₁ (3), however, the maximum value of α_i could not exceed the constant value C due to the constraint, $0 \leq \alpha_i \leq C$. According to these observations, we argue that the SVM₁ is more robust than the SVM₂ in outlier resistance. Hence, we develop SSVM₁, which will be introduced in next section.

3 1-Norm Soft Margin Smooth SVM (SSVM₁)

Similar to the framework of SSVM₂ (9), the classification problem (1) is reformulated as follows:

$$\begin{aligned} & \min_{(w,b,\xi) \in R^{n+1+m}} \frac{1}{2}(\|w\|_2^2 + b^2) + C\|\xi\|_1 \\ & \text{subject to : } \quad D(Aw + \mathbf{1}b) + \xi \geq \mathbf{1} \\ & \quad \quad \quad \xi \geq \mathbf{0}. \end{aligned} \quad (5)$$

In the solution of problem (5), ξ is given by

$$\xi = (\mathbf{1} - D(Aw + \mathbf{1}b))_+, \quad (6)$$

where $(\cdot)_+$ is defined by $\max\{\cdot, 0\}$. Namely, if $1 - D_{ii}(A_i w + b) \leq 0$, then $\xi_i = 0$. Thus, this ξ in the objective function of problem (5) is replaced by $(\mathbf{1} - D(Aw + \mathbf{1}b))_+$ so that problem (5) can be converted into an unconstrained optimization problem as follows:

$$\min_{(w,b) \in R^{n+1}} \frac{1}{2}(\|w\|_2^2 + b^2) + C\|(\mathbf{1} - D(Aw + \mathbf{1}b))_+\|_1. \quad (7)$$

The problem is a strongly convex minimization problem without any constraint. Thus, problem (7) has a unique solution. Obviously, the objective function in problem (7) is not twice differentiable, so the Newton method can not be applied to solve this problem. Therefore, SSVM₂ employs a smoothing function (5) to replace the original plus function. The smoothing function is given by $p(x, \alpha)$, the integral of the sigmoid function $\frac{1}{1+e^{-\alpha x}}$ of neural networks (11), that is,

$$p(x, \alpha) = x + \frac{1}{\alpha} \log(1 + e^{-\alpha x}) \text{ for } \alpha > 0, \quad (8)$$

where α is a smoothing parameter to adjust the degree of approximation. Note that if the value of α increases, the $p(x, \alpha)$ will approximate the plus function

more accurately. Next, the $p(x, \alpha)$ is taken into problem (7) to replace the plus function as follows:

$$\min_{(w,b) \in R^{n+1}} \frac{1}{2}(\|w\|_2^2 + b^2) + C \|p(\mathbf{1} - D(Aw + \mathbf{1}b), \alpha)\|_1. \quad (9)$$

By taking the advantage of the twice differentiability of the objective functions on problem (9), a prescribed quadratically convergent Newton-Armijo algorithm [3] can be introduced to solve this problem. Hence, the smoothing problem can be solved without a sophisticated optimization solver.

Moreover, we can obtain the unconstrained nonlinear smooth SVM₁ by applying the kernel trick [12] on problem (9) as follows:

$$\min_{(u,b) \in R^{m+1}} \frac{1}{2}(\|u\|_2^2 + b^2) + C \|p(\mathbf{1} - D(K(A, A^\top)u + \mathbf{1}b), \alpha)\|_1. \quad (10)$$

The nonlinear separating surface is defined by the optimal solution u and b of (10):

$$K(A, x)u + b = 0. \quad (11)$$

The computational complexity for solving (10) is $\mathcal{O}((m+1)^3)$. To conquer the computation difficulty caused by a big full kernel matrix $K(A, A^\top)$, we introduce the reduced kernel technique [8] to replace it by $K(A, \bar{A}^\top)$. The key idea of the reduced kernel technique is to randomly select a small portion of data and to generate a thin rectangular kernel matrix to replace the full kernel matrix. The reduced kernel method constructs a compressed model and cuts down the computational cost from $\mathcal{O}(m^3)$ to $\mathcal{O}(\bar{m}^3)$. It has been shown that the solution of reduced kernel matrix approximates the solution of full kernel matrix well.

4 A Heuristic Method for Outlier Filtering

So far, SSVM₁ has been developed for better outlier resistance, but there are some violent cases that are still easy to mislead either 1-norm soft margin SVMs or 2-norm soft margin SVMs to lose their generalization ability. We present a violent case in Fig. 1. It shows that no matter the 1-norm soft margin SVMs (SSVM₁ and LIBSVM [4]) or the 2-norm soft margin SVM (SSVM₂), all of them cannot separate the major parts of positive and negative examples. Why all of the SVMs lose their generalization ability in this case is that they pay too much effort to minimize the *loss term* and sacrifice for minimizing the *penalty term* because of these extreme outliers.

To rescue the SVMs from such the violent case, we prescribe a heuristic method to filter out the extreme outliers, which makes SVMs be more balanced to minimize both *penalty term* and *loss term* at the same time. Our strategy is to continue removing the training input with a large ξ_i in each Newton iteration and make sure that the removed number is still smaller than the outlier ratio, which is given by the intuition of users or data collectors. In implementation, the removal is arranged to distribute fairly in several iterations according to the setting outlier ratio.

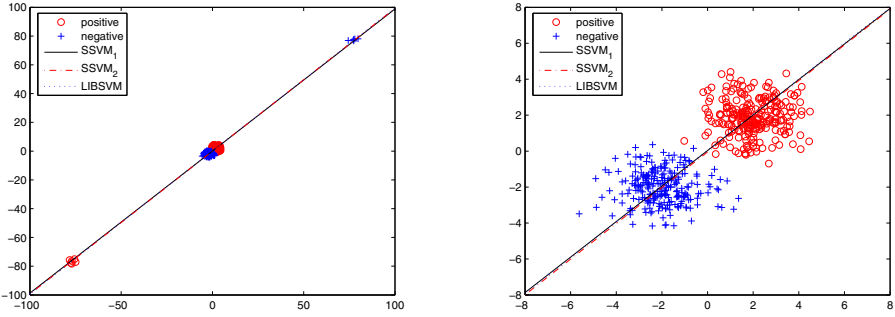


Fig. 1. (Synthetic Dataset: a normal distribution, mean = 2 and -2, the standard deviation = 1) The outlier ratio is 0.025 (outliers are on the upper-right and lower-left corners in (a)). For the outliers, the outlier difference from the mean of their groups is set to be 75 times the standard deviation. All classifiers are seriously affected by these outliers.

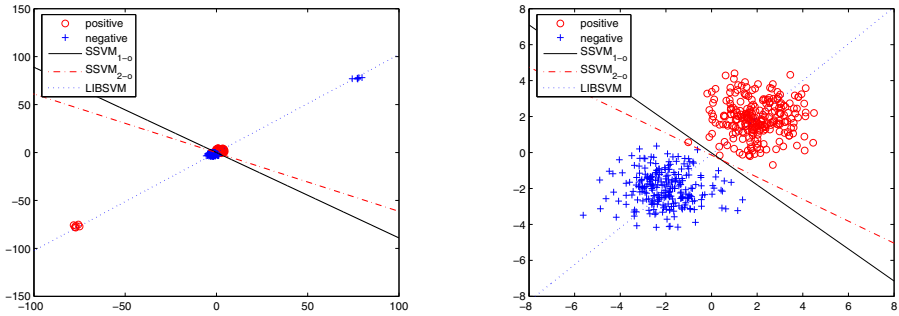


Fig. 2. $SSVM_{1-o}$ and $SSVM_{2-o}$ have successfully remedied the classification results of $SSVM_1$ and $SSVM_2$ in Fig. 1. LIBSVM is still affected by the outliers a lot.

Note that the outlier filtering process is also embedded in $SSVM_2$ to compare with $SSVM_1$ in experiments. We denote $SSVM_{1-o}$ and $SSVM_{2-o}$ to represent the $SSVM_1$ and $SSVM_2$ with filtering strategy. In order to see the power of the heuristic filtering method, we test $SSVM_{1-o}$ and $SSVM_{2-o}$ on the identical synthetic dataset in Fig.1 again. Fig. 2 shows that $SSVM_{1-o}$ and $SSVM_{2-o}$ indeed remedy the previous classification results of $SSVM_1$ and $SSVM_2$ in Fig. 1, and they are superior to LIBSVM without outlier filtering mechanism.

5 Numerical Results

All codes of SSVMs are written in Matlab [14]. In experiments, we test the $SSVM_2$, $SSVM_1$, LIBSVM [4], $SSVM_{2-o}$ and $SSVM_{1-o}$ on ten publicly available binary class datasets from the UCI Machine Learning Repository [2] and CBCL datasets: Wisconsin Prognostic Breast Cancer Database [13], Ionosphere,

Table 1. Numerical comparisons of nonlinear SVMs on the original and polluted data problems

		10-fold training correctness, % 10-fold testing correctness, %				
		Method				
Dataset size (reduced ratio)						
m x n		SSVM ₂	SSVM ₁	LIBSVM	SSVM _{2-o}	SSVM _{1-o}
WPBC	194 × 34	88.69	86.02	85.91	78.13	82.10
		81.67	80.00	80.00	79.44	79.44
Ionosphere	351 × 34	96.78	99.43	99.43	96.66	98.45
		96.18	96.47	96.47	95.59	95.88
BUPA	345 × 6	76.21	76.4	75.88	76.50	76.3
		74.41	75.29	74.71	75.59	74.71
Pima Indians	768 × 8	77.95	77.62	77.88	82.34	77.76
		78.82	78.82	78.42	78.29	78.55
Cleveland	296 × 13	86.67	85.54	84.53	84.34	85.47
		84.14	85.17	84.48	84.14	84.48
WDBC	569 × 30	99.14	99.24	99.06	96.78	98.81
		98.21	98.21	98.21	96.96	98.04
Face (r=0.01)	6977 × 361	98.76	98.82	98.68	97.90	98.28
		98.29	98.51	98.38	97.84	98.05
Image (r=0.01)	2310 × 18	92.39	91.52	91.67	90.49	90.54
		92.16	91.17	91.26	89.91	90.04
Singleton (r=0.01)	3175 × 60	79.58	80.56	81.32	81.98	81.41
		79.11	79.68	81.30	81.17	79.87
Waveform (r=0.01)	5000 × 21	91.52	91.86	91.47	91.94	92.34
		91.08	91.38	91.04	91.28	91.00

(a) The results on original data problems and the best values are emphasized in boldface. The outlier ratio parameters of SSVM_{2-o} and SSVM_{1-o} are set to 5%.

		10-fold training correctness, % 10-fold testing correctness, %				
		Method				
Dataset size (reduced ratio)						
m x n		SSVM ₂	SSVM ₁	LIBSVM	SSVM _{2-o}	SSVM _{1-o}
WPBC	194 × 34	72.84	71.02	71.02	80.23	81.93
		78.33	77.78	77.78	79.44	80.00
Ionosphere	351 × 34	87.03	88.58	85.49	84.42	88.74
		92.35	93.24	92.94	92.06	93.24
BUPA	345 × 6	73.05	72.80	72.38	71.03	72.48
		72.06	72.06	72.65	72.65	73.82
Pima Indians	768 × 8	69.93	72.30	72.47	73.71	72.88
		75.00	76.71	76.84	77.89	77.24
Cleveland	296 × 13	79.25	80.67	80.04	78.50	80.79
		84.83	84.83	84.14	84.83	85.17
WDBC	569 × 30	87.80	88.79	88.69	89.24	89.55
		97.32	97.68	97.32	97.32	97.14
Face (r=0.01)	6977 × 361	91.33	90.89	89.06	90.49	90.20
		93.39	93.74	93.29	93.97	94.90
Image (r=0.01)	2310 × 18	82.02	81.99	81.30	82.71	84.25
		89.65	89.91	89.26	90.52	91.95
Singleton (r=0.01)	3175 × 60	73.61	76.12	74.21	74.10	77.82
		78.45	80.89	78.45	78.58	82.59
Waveform (r=0.01)	5000 × 21	83.19	83.36	83.21	83.67	83.61
		90.84	90.86	91.14	91.18	91.20

(b) The results on the data problems with 10% outlier pollution and the best values are emphasized in boldface. The outlier ratio parameters of SSVM_{2-o} and SSVM_{1-o} are set to 10%.

BUPA Liver, Pima Indians, Cleveland Heart Problem, WDBC, Image, Singleton, Waveform and CBCL Face Database [1]. We perform 10-fold cross-validation on each dataset in order to evaluate how well each SVM generalizes to future data.

We train all of the classifiers by Gaussian (RBF) kernel, which is defined as $K(A, A^T)_{ij} = e^{-\gamma \|A_i - A_j\|_2^2}$, $i, j = 1, 2, 3 \dots m$. To build up a satisfied SVM model, we need to search a good pair of *Gaussian kernel width parameter* γ and *regularization parameter* C . A well developed model selection method is nested uniform designs (UDs) [7], which is applied in experiments. In [7], the results by using the nested-UDs are usually good enough with much less computational cost as compared to the grid search for parameters tuning. For the large-scale datasets (CBCL Face Database, Image, Singleton and Waveform), we apply the reduced kernel technique (1% from the columns of the full kernel) to the SSVMs except for LIBSVM.

Since the specificity and the sensitivity of the tests are not unusual for all the methods, on the limit of space we just report the average training and testing correctness of 10-fold cross-validation in Table 1. In the part (b) of Table 1, we try to pollute the datasets by replacing 10% outlier training samples into each dataset. The experiments show that SSVM_{1-o} performs very well in dealing with the problems with outliers.

6 Conclusions

In this paper, we argue that 1-norm soft margin SVMs have better outlier resistance than 2-norm soft margin SVMs, so we develop SSVM₁ by modifying the previous framework in SSVM₂. To strengthen the robustness of SSVM₁ in some violent cases, we also propose the heuristic method for outlier filtering. From experiments, we see that the 1-norm soft margin SVMs do have better robustness, and the heuristic filtering method, which costs little in training process, improves the outlier resistance a lot.

References

1. CBCL Face Database #1. MIT Center For Biological and Computation Learning, <http://cbcl.mit.edu/software-datasets/FaceData2.html>
2. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
3. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Software available at, <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
5. Chen, C., Mangasarian, O.L.: A class of smoothing functions for nonlinear and mixed complementarity problems. Computational Optimization and Applications 5(2), 97–138 (1996)

6. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
7. Huang, C.M., Lee, Y.J., Lin, D.K.J., Huang, S.Y.: Model selection for support vector machines via uniform design. A special issue on Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis 52, 335–346 (2007)
8. Lee, Y.J., Huang, S.Y.: Reduced support vector machines: a statistical theory. IEEE Transactions on Neural Networks 18, 1–13 (2007)
9. Lee, Y.J., Mangasarian, O.L.: SSVM: A smooth support vector machine. Computational Optimization and Applications 20, 5–22 (2001)
10. Liu, Y., Wu, Y.: Robust truncated-hinge-loss support vector machines. Journal of the American Statistical Association 102, 974–983 (2007)
11. Mangasarian, O.L.: Mathematical Programming in Neural Networks. ORSA Journal on Computing 5(4), 349–360 (1993)
12. Mangasarian, O.L.: Generalized support vector machines. In: Smola, A., Bartlett, P., Schölkopf, B., Shuurmans, D. (eds.) Advance in Large Margin Classifiers, pp. 135–146. MIT Press, Cambridge (2000)
13. Mangasarian, O.L., Street, W.N., Wolberg, W.H.: Breast cancer diagnosis and prognosis via linear programming. Operations Research 43(4), 570–577 (1995)
14. MATLAB: User's Guide. The MathWorks, Inc., Natick (1994) 01760
15. Xu, H., Caramanis, C., Mannor, S.: Robustness and regularization of support vector machines. Journal of Machine Learning Research 10, 1485–1510 (2009)

A Generalization of Independence in Naive Bayes Model

Yu Fujimoto¹ and Noboru Murata²

¹ Aoyama Gakuin University,
5-10-1 Fuchinobe, Chuo, Sagami-hara, Kanagawa 252-5258, Japan

² Waseda University,
3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555, Japan

Abstract. In this paper, generalized statistical independence is proposed from the viewpoint of generalized multiplication characterized by a monotonically increasing function and its inverse function, and it is implemented in naive Bayes models. This paper also proposes an idea of their estimation method which directly uses empirical marginal distributions to retain simplicity of calculation. Our method is interpreted as an optimization of a rough approximation of the Bregman divergence so that it is expected to have a kind of robust property. Effectiveness of our proposed models is shown by numerical experiments on some benchmark data sets.

Keywords: naive Bayes, generalized independence, nonloglinear marginal model, copula, Bregman divergence.

1 Introduction

Statistical models based on some kind of independence, such as naive Bayes (NB) models, Bayesian networks [1] or pLSA [2], are broadly used in various situations; the assumption of independence is attractive in modeling between categorical variables with a lot of categories because the composed model may have a significantly smaller number of parameters than the model denoting dependence. Technically, the assumption of independence in these models should be introduced by analyzing a data set. However, in practical scenes, the models are casually used without rigorous analysis; e.g., in classification problems, it is known that the NB model shows good performance even if the assumption is violated [3]. In this paper, we introduce a generalization of independence and propose an extension of the NB model to express weak special dependence with the small number of parameters.

In the statistical inference, we naturally use arithmetic operators, such as multiplication or division, for probability values. For instance, statistical independence can be defined with multiplication of marginal probabilities. We can generalize these operators with an appropriate monotonically increasing function $u(\cdot)$ and its inverse function $\xi(\cdot)$. For example, multiplication between two positive values a and b are generalized as follows.

$$a \times b = \exp(\log(a) + \log(b)) \xrightarrow{\text{generalization}} u(\xi(a) + \xi(b)).$$

This type of generalization has been proposed in several contexts (for example, discussions from a perspective of density integration are given in [4,5]), and is closely related to nonloglinear marginal models [6] or the Archimedean copula [7]. In this paper, we generalize conditional independence in NB models by using generalized multiplication, and experimentally show the effectiveness of our proposed model.

This paper is composed as follows. In Section 2, we introduce an idea of generalized independence defined with a monotonically increasing function. Some properties of the generalization are also discussed in this section. In Section 3, the NB model is extended by implementing generalized independence in several ways. In Section 4, we numerically evaluate extended NB models by using benchmark data sets. Lastly, in Section 5, concluding remarks are given.

2 Generalized Independence

In this section, we introduce generalized multiplication denoted with a monotonically increasing function $u(\cdot)$ [8].

2.1 U-multiplication and Generalized Independence

Let $\mathbf{X} = \{X^1, \dots, X^M\}$ be a set of M categorical variables where X^m has a domain $\mathcal{X}^m = \{x_i^m\}_{i=1}^{I_m}$, and $p_{\mathbf{X}}(\mathbf{x})$ be a joint probability of $\mathbf{x} \in \mathbf{X}$. Given marginal probability density functions (pdfs), defined as

$$p_{\mathcal{X}^m}(x^m) = \sum_{l \neq m} \sum_{x^l \in \mathcal{X}^l} p_{\mathbf{X}}(x^1, \dots, x^M), \quad (1)$$

then statistical independence is defined as follows.

Definition 1 (Independence). Let p_{\times} be the joint pdf defined with marginal pdfs $p_{\mathcal{X}^1}, \dots, p_{\mathcal{X}^M}$ as

$$p_{\times}(\mathbf{X}; p_{\mathcal{X}^1}, \dots, p_{\mathcal{X}^M}) = \prod_{m=1}^M p_{\mathcal{X}^m}(X^m) = \exp\left(\sum_{m=1}^M \log(p_{\mathcal{X}^m}(X^m))\right). \quad (2)$$

Variables X^1, \dots, X^M are mutually independent if their joint pdf $p_{\mathbf{X}}$ has the following property,

$$p_{\mathbf{X}}(\mathbf{X}) = p_{\times}(\mathbf{X}). \quad (3)$$

Equations (3) and (2) indicate that the sum of logarithmic marginal probabilities in the function $\exp(\cdot)$ defines statistical independence. By introducing a monotonically increasing function $u(\cdot)$, we can generalize Definition 1, as follows.

Table 1. Examples of functions $u(\cdot)$ and $\xi(\cdot)$

	$u(z)$	$\text{dom}(u)$	$\text{range}(u)$	$\xi(z) = u^{-1}(z)$	$\text{dom}(\xi)$	$\text{range}(\xi)$	$\text{dom}(\pi)$
	$\exp(z)$	$(-\infty, \infty)$	$(0, \infty)$	$\log(z)$	$(0, \infty)$	$(-\infty, \infty)$	-
Ex.1	$(\pi z + 1)^{\frac{1}{\pi}}$	$[-\frac{1}{\pi}, \infty)$	$[0, \infty)$	$\frac{z^{\pi}-1}{\pi}$	$[0, \infty)$	$[-\frac{1}{\pi}, \infty)$	$(0, \infty)$
Ex.2	$\exp(z) + \pi$	$(-\infty, -\frac{1}{\pi})$	$(0, \infty)$	$\log(z - \pi)$	(π, ∞)	$(-\infty, -\frac{1}{\pi})$	$(-\infty, 0)$
Ex.3	$\exp\left(\text{sgn}(z) z ^{\frac{1}{\pi}}\right)$	$(-\infty, \infty)$	$(0, \infty)$	$\text{sgn}(\log(z)) \log(z) ^{\pi}$	$(0, \infty)$	$(-\infty, \infty)$	$(-\infty, \inf(z))$
Ex.4	$\exp\left(\frac{1-\exp(-z)}{\pi}\right)$	$(-\infty, \infty)$	$(0, \exp(\frac{1}{\pi}))$	$-\log(1 - \pi \log(z))$	$(0, \exp(\frac{1}{\pi}))$	$(-\infty, \infty)$	$(0, \frac{1}{\log(\sup(z))})$

Definition 2 (*U-independence* [8]). Let $u(\cdot)$ be a monotonically increasing function, $\xi(\cdot) = u^{-1}(\cdot)$ be its inverse function and p_{\otimes} be the joint pdf defined by using $u(\cdot)$ and $\xi(\cdot)$ as

$$\begin{aligned}
 p_{\otimes}(\mathbf{X}; p_{\mathcal{X}^1}, \dots, p_{\mathcal{X}^M}, u) &= u\left(\sum_{m=1}^M \xi(p_{\mathcal{X}^m}(X^m)) - c\right) \tag{4} \\
 &= p_{\mathcal{X}^1}(X^1) \otimes p_{\mathcal{X}^2}(X^2) \otimes \dots \otimes p_{\mathcal{X}^M}(X^M) \\
 &= \bigotimes_{m=1}^M p_{\mathcal{X}^m}(X^m), \tag{5}
 \end{aligned}$$

where c is a constant to satisfy $\sum_{\mathbf{x} \in \mathbf{X}} p_{\otimes}(\mathbf{x}) = 1$. Variables X^1, \dots, X^M are called mutually *U-independent* if their joint pdf $p_{\mathcal{X}}$ has the following property,

$$p_{\mathcal{X}}(\mathbf{X}) = p_{\otimes}(\mathbf{X}). \tag{6}$$

Note that we assume that $\text{range}(\sum_{m=1}^M \xi(p_{\mathcal{X}^m}) - c) \subseteq \text{dom}(u)$ and $\text{range}(u) \supseteq \text{range}(p_{\otimes})$ hold.

We use the operator \otimes to derive a *U-independent* pdf, and the operation is called *U-multiplication* in this paper[1]. In Table 1, ranges and domains of several one-parameter family functions for p_{\otimes} are shown. The *U-independent* model is a kind of nonloglinear marginal model [6] and naturally interpreted as a generalization of the independence expression in the loglinear model [9] by using $\xi(\cdot)$ instead of $\log(\cdot)$. Intuitively speaking, *U-independence* indicates that a sample set is observed in non-independent way and shows kind of weak dependence in the conventional term.

Figure 1 shows intuitive differences between conventional independence and *U-independence*. As shown in the figure, changing $u(\cdot)$ and $\xi(\cdot)$ in *U-multiplication*

¹ With Definition 2, we also obtain a marginal pdf from the given *U-independent* joint pdf as follows,

$$p_{\mathcal{X}^m}(X^m) = u\left(\xi(p_{\otimes}(\mathbf{X})) - \sum_{l \neq m} \xi(p_{\mathcal{X}^l}(X^l)) - c'\right),$$

where c' is a constant to satisfy $\sum_{x^m \in X^m} p_{\mathcal{X}^m}(x^m) = 1$.

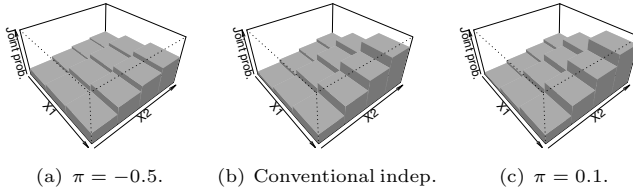


Fig. 1. Examples of $p_{\otimes} = p_{\mathcal{X}^1} \otimes p_{\mathcal{X}^2}$ constructed with Ex.1 in Table 1, where $p_{\mathcal{X}^1} = \{0.167, 0.333, 0.5\}$ and $p_{\mathcal{X}^2} = \{0.1, 0.2, 0.3, 0.4\}$. Note that conventional independence indicates $\pi \rightarrow 0.0$ in this case.

overstates, or understates, probabilities in marginal distributions. Therefore U -independence is interpreted as an expression of weak special dependence between variables. From another perspective, we can say that U -independence constructs a kind of copula [7] based on pdfs instead of cumulative distribution functions (cdf). When the variables are continuous, this difference might be a drawback for U -independence because it is impossible to control dependency of upper and lower tails of marginals separately. However, this property is convenient to control these tails simultaneously, and it is handy to control categorical distributions.

2.2 Empirical Marginals

Consider a set of two discrete variables $\mathbf{X} = \{X^1, X^2\}$. Let \mathcal{P}_{\times} and \mathcal{P}_{\otimes} be sets of independent and U -independent distributions, given as

$$\begin{aligned} \mathcal{P}_{\times} &= \{p_{\times}(\mathbf{X}; p_{\mathcal{X}^1}, p_{\mathcal{X}^2})\} \\ \mathcal{P}_{\otimes}(\mathcal{U}) &= \{p_{\otimes}(\mathbf{X}; p_{\mathcal{X}^1}, p_{\mathcal{X}^2}, u) \mid u \in \mathcal{U}\}, \end{aligned}$$

where \mathcal{U} is a set of monotonically increasing functions. Let $\tilde{p}_{\mathbf{X}}$ and $\tilde{p}_{\mathcal{X}^m}$ be empirical distributions, given as

$$\tilde{p}_{\mathbf{X}}(\mathbf{x}) = \frac{n_{\mathbf{x}}}{\sum_{\mathbf{x}' \in \mathbf{X}} n_{\mathbf{x}'}} \quad \text{and} \quad \tilde{p}_{\mathcal{X}^m}(x^m) = \frac{n_{x^m}}{\sum_{x'^m \in \mathcal{X}^m} n_{x'^m}},$$

where $n_{\mathbf{x}}$ is the frequency of the observed event $\mathbf{x} \in \mathbf{X}$ and n_{x^m} is that of $x^m \in \mathcal{X}^m$. The maximum likelihood (ML) estimate of the conventional independent model is given with empirical marginals, as follows,

$$\begin{aligned} \operatorname{argmin}_{q \in \mathcal{P}_{\times}} D_{\text{KL}}(\tilde{p}_{\mathbf{X}}, q) &= \operatorname{argmin}_{q \in \mathcal{P}_{\times}} \sum_{\mathbf{x} \in \mathbf{X}} \tilde{p}_{\mathbf{X}}(\mathbf{x}) \log \frac{\tilde{p}_{\mathbf{X}}(\mathbf{x})}{q(\mathbf{x})} \\ &= \tilde{p}_{\mathcal{X}^1} \times \tilde{p}_{\mathcal{X}^2}, \end{aligned} \tag{7}$$

where $D_{\text{KL}}(\tilde{p}_{\mathbf{X}}, q)$ is the KL divergence between $\tilde{p}_{\mathbf{X}}$ and q . On the other hand, to obtain the ML estimate of the U -independent model, we need to solve a non-linear optimization problem, that is

$$\operatorname{argmin}_{q \in \mathcal{P}_{\otimes}(\mathcal{U})} D_{\text{KL}}(\tilde{p}_{\mathbf{X}}, q) = \hat{u}(\hat{\xi}(\hat{p}_{\mathcal{X}^1}) + \hat{\xi}(\hat{p}_{\mathcal{X}^2}) - c), \tag{8}$$

with respect to marginals $\hat{p}_{\mathcal{X}^1}, \hat{p}_{\mathcal{X}^2}$ and the function \hat{u} for multiplication operator. In this paper, we only focus on searching a function u from a one parameter family $\mathcal{U} = \{u(z; \pi)\}$ which is given as an example in Table 1 and use empirical marginals; i.e., we find the estimate

$$\operatorname{argmin}_{q \in \tilde{\mathcal{P}}_{\otimes}(\mathcal{U})} D_{\text{KL}}(\tilde{p}_{\mathbf{X}}, q) = \hat{u}(\hat{\xi}(\tilde{p}_{\mathcal{X}^1}) + \hat{\xi}(\tilde{p}_{\mathcal{X}^2}) - c) \tag{9}$$

where

$$\tilde{\mathcal{P}}_{\otimes}(\mathcal{U}) = \{p(\mathbf{X}; u) = \tilde{p}_{\mathcal{X}^1}(X^1) \otimes \tilde{p}_{\mathcal{X}^2}(X^2) \mid u \in \mathcal{U}\}.$$

Solving Eq. (9) with respect to \hat{u} is much simpler than solving Eq. (8) with respect to $\hat{u}, \hat{p}_{\mathcal{X}^1}$ and $\hat{p}_{\mathcal{X}^2}$. We call the solution of Eq. (9) *empirical U-independent model*.

Note that the set $\mathcal{P}_{\otimes}(u)$ defined with any u functions have the following property. Let us define uniform distributions on respective domains as

$$\bar{p}_{\mathcal{X}^1}(X^1) = \frac{1}{I^1}, \quad \bar{p}_{\mathcal{X}^2}(X^2) = \frac{1}{I^2}, \quad \bar{p}_{\mathbf{X}}(\mathbf{X}) = \frac{1}{I^1 \times I^2},$$

where I^1 and I^2 are the number of elements in \mathcal{X}^1 and \mathcal{X}^2 . Then, the following property holds with any types of $u(\cdot)$,

$$\bar{p}_{\mathbf{X}}(\mathbf{X}) = \bar{p}_{\mathcal{X}^1}(X^1) \otimes \bar{p}_{\mathcal{X}^2}(X^2). \tag{10}$$

As denoted in the previous subsection, the joint expression of the U -independent distribution is affected by the form of $u(\cdot)$, however it is reduced to $\bar{p}_{\mathbf{X}}$ for any $u(\cdot)$ in the case that all the marginals are uniform distributions. This fact indicates that the space of empirical U -independence $\tilde{\mathcal{P}}_{\otimes}$ is not a rich subspace in $\mathcal{P}_{\mathbf{X}}$ if the empirical marginals are close to uniform. On the other hand, when the marginals are far from uniform and have extremely high (or low) probabilities because of small sample sets or outliers, empirical U -independent models can be flexible and convenient candidates.

The ML estimation of the empirical U -independent model is interpreted from the viewpoint of an approximated Bregman divergence [8]. Therefore, with an appropriate function u , the empirical U -independent model enjoys robustness which the Bregman divergence essentially has.

3 Extension of Naive Bayes Model

Let Y be a categorical variable, and $\mathbf{X} = \{X^m, \dots, X^M\}$ be a set of categorical variables. Then, the naive Bayes (NB) model is defined as follows,

$$\begin{aligned} p_{\text{NB}}(\mathbf{X}, Y) &= p(Y)p(\mathbf{X}|Y) \\ &= p(Y) \prod_{m=1}^M p(X^m|Y). \end{aligned} \tag{11}$$

The NB has some convenient properties, such as simple structure, easy estimation and scalability. And it is also known as a simple but robust classification tool [3]. With the empirical joint distribution $\tilde{p}(\mathbf{X}, Y)$, the ML estimate is given by

$$\hat{p}_{\text{NB}} = \underset{p_{\text{NB}}}{\operatorname{argmin}} \operatorname{D}_{\text{KL}}(\tilde{p}, p_{\text{NB}}). \tag{12}$$

The concrete form of $\hat{p}_{\text{NB}}(\mathbf{X}, Y) = \hat{p}(Y) \prod_{m=1}^M \hat{p}(X^m|Y)$ is composed of

$$\hat{p}(y) = \frac{n(y)}{\sum_{y' \in Y} n(y')} \tag{13}$$

$$p(x^m|y) = \frac{n(x^m, y) + \alpha}{\sum_{x'^m \in X^m} (n(x'^m, y) + \alpha)}, \tag{14}$$

where $n(y)$ and $n(x^m, y)$ are the numbers of observations of events y and (x^m, y) respectively, and $\alpha \in [0, 1]$ is a Laplace smoother for estimation of $p(X^m|Y)$.

Now, we consider an extension of the NB model by using U -independence. For example, assume that all the elements in the variable set \mathbf{X} are mutually conditional U -independent given Y , we can derive the following expression,

$$p_U(\mathbf{X}, Y) = p(Y) \left(\bigotimes_{m=1}^M p(X^m|Y) \right). \tag{15}$$

For another example, assume that only some of the elements in \mathbf{X} are conditionally U -independent, given as

$$p_U(\mathbf{X}, Y) = p(Y) \left(\bigotimes_{m \in \bar{S}_I} p(X^m|Y) \right) \left(\prod_{m' \in S_I} p(X^{m'}|Y) \right), \tag{16}$$

where \bar{S}_I is an index set of weakly dependent variables in X^1, \dots, X^M and S_I is an index set of the rest.

Given \tilde{p} , the ML estimates of Eqs. (15) and (16) are derived by

$$\hat{p}_U = \underset{p_U}{\operatorname{argmin}} \operatorname{D}_{\text{KL}}(\tilde{p}, p_U). \tag{17}$$

Exact solution of Eq. (17) is derived by solving nonlinear optimization problem. Alternatively, we directly use empirical distributions Eqs. (13) and (14) and select an appropriate function u in an analogous way as the discussion of the empirical U -independent model introduced in the previous section. The extended NB models with empirical marginals are called empirical U -NB models in this paper.

4 Numerical Experiments

In this section, we experimentally evaluate the empirical U -NB by using some benchmark data sets. As given in Eqs. (15) and (16), there are some implementation manners of U -independence for the NB model. Therefore, in the first

Table 2. Data sets used in experiments

data set	M	# train data	# test data	note
MONKS1	6	124	432	$Y = 1$ when $(X^1 = X^2)$ or $X^5 = 1$
MONKS2	6	169	432	$Y = 1$ when exactly two of $\{X^m\}$ are 1
CAR	6	300	1728	
NUR	8	300	12960	

experiment, we compare some U -NBs extended with different manners. In addition, as denoted at the end of Section 2, the empirical U -independent model is expected to be a good candidate in the case that the sample set is small. In the second experiment, we try to tune the function u in the U -NB by using a small data set. In both experiments, we find an optimal u with respect to π in a one-parameter family Ex.1 in Table 1.

Here, we compare some extended NBs by using data sets “MONK1” and “MONK2” distributed in UCI ML repository [10]. These data sets are composed of a binary class variable Y and a discrete variable set $\mathbf{X} = \{X^1, \dots, X^6\}$. And there is kind of dependence in \mathbf{X} to determine Y as shown in Table 2. We compared three models given by Eq. (15) (model 1), Eq. (16) with $\bar{S}_I = \{1, 2\}$ (model 2) and Eq. (16) with $\bar{S}_I = \{3, 4, 5, 6\}$ (model 3); all the models are reduced to Eq. (11) at $\pi = 1$.

At first, we obtained the estimate \hat{p}_U by using an empirical distribution \tilde{p} of the training data set. Then we evaluated it by the KL divergence $D_{\text{KL}}(p^*, \hat{p}_U)$ where p^* is an empirical distribution of the test data set. In this experiment, we set $\alpha = 0$ in Eq. (14). Figure 2(a) shows the estimation result of “MONK1” data set for various π values. The figure indicates that models 1 and 2 which use U -multiplication for representation of relation between X^1 and X^2 show improvement. On the other hand, model 3 does not improve the result. In a similar way, Figure 2(b) shows the result of “MONK2” which has dependence in all the elements in \mathbf{X} . In this case, model 2 which uses U -multiplication only between X^1 and X^2 does not show improvement. However, model 3 (which applies U -multiplication for over half of the variables) and model 1 (which applies U -multiplication for all the variables in \mathbf{X}) show improvement. The results indicate that if there is no knowledge about dependence between variables, then a generalization like model 1 has a possibility to improve the NB model.

Next, we compared the U -NB (model 1) with the conventional NB with an appropriately tuned Laplace smoother. In addition to previously denoted data sets, we also use “car evaluation” (CAR) and “nursery” (NUR) from UCI ML repository in this experiment. At first, we find the optimal $\hat{\alpha}$ for \hat{p}_{NB} by using training data sets with 10-fold cross validation (CV). Secondly, we find the optimal $\hat{\pi}$ for \hat{p}_U under $\hat{\alpha}$ with CV. Then, we evaluate $D_{\text{KL}}(p^*, \hat{p}_{\text{NB}})$ and $D_{\text{KL}}(p^*, \hat{p}_U)$. The experimental results shown in Table 3 indicate that U -NBs outperform conventional NBs even if NBs have appropriately tuned Laplace smoothers. Thus, we see that the estimation of an empirical U -NB has a robust property even when the data set is not composed of a large number of samples.

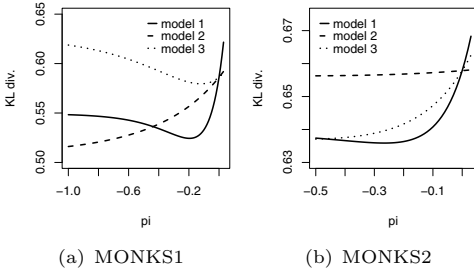


Fig. 2. Results of experiment 1

5 Conclusion

In this paper, we introduced U -independence and proposed an extension of the NB model. To reduce computational cost, we also proposed the empirical U -NB model which has robust property attributable to the approximated Bregman divergence. In the same manner with the U -NB model, we can extend loglinear models and Bayesian networks. Compared with cross terms in loglinear models or link expression in Bayesian network, U -independence has some advantages in the number of parameters and in robustness; e.g., we expect simple description of graphical models by omitting some links with weak dependence by using U -independence.

It is interesting to use the U -NB as a classifier. Especially, we can combine it with the useful classification method, such as complement naive Bayes [11]. The selection of an appropriate u is another interesting topic though it remains as the future work.

Acknowledgment. This work was partially supported by JSPS, Grant-in-Aid for Young Scientists(B), 22700292.

References

1. Jensen, F.V.: Bayesian Networks and Decision Graphs. In: Statistics for Engineering and Information Science. Springer, Heidelberg (2001)
2. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning 42, 177–196 (2001)
3. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–130 (1997)
4. Amari, S.: Integration of stochastic models by minimizing α -divergence. Neural Computation 19, 2780–2796 (2007)
5. Murata, N., Fujimoto, Y.: Bregman divergence and density integration. Journal of Math-for-Industry 1, 97–104 (2009)
6. Bergsma, W., Croon, M., Hagenaars, J.A.: Marginal Models for Dependent, Clustered, and Longitudinal Categorical Data. Springer, Heidelberg (2009)

Table 3. Results of experiment 2

data set	$D_{\text{KL}}(p^*, \hat{p}_U)$	
	NB	U-NB
MONKS1	0.5796	0.5518 ($\hat{\pi} = -0.12$)
MONKS2	0.6520	0.6385 ($\hat{\pi} = -1.05$)
CAR	0.6307	0.6251 ($\hat{\pi} = -0.07$)
NUR	0.4321	0.4204 ($\hat{\pi} = -0.03$)

7. Nelsen, R.B.: An Introduction to Copulas, 2nd edn. Springer Series in Statistics. Springer, Heidelberg (2006)
8. Fujimoto, Y., Murata, N.: A generalized product rule and weak independence based on Bregman divergence. In: Proc. 12th World Multi-Conference on Systemics, Cybernetics and Informatics (2008)
9. Agresti, A.: Categorical Data Analysis, 2nd edn. Wiley Inc., Chichester (2002)
10. Asuncion, A., Newman, D.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2007)
11. Rennie, J.D., Shih, L., Teevan, J., Karger, D.: Tackling the poor assumptions of naive Bayes text classifiers. In: Proceedings of the Twentieth International Conference on Machine Learning, ICML 2003 (2003)

Interval Filter: A Locality-Aware Alternative to Bloom Filters for Hardware Membership Queries by Interval Classification

Ricardo Quislan, Eladio Gutierrez, Oscar Plata, and Emilio L. Zapata

Department of Computer Architecture, University of Málaga,
ETSI Informática, Campus Teatinos, Málaga, E 29071, Spain
{quislant,eladio,oplata,zapata}@uma.es

Abstract. Bloom filters are data structures that can efficiently represent a set of elements providing operations of insertion and membership testing. Nevertheless, these filters may yield false positive results when testing for elements that have not been previously inserted. In general, higher false positive rates are expected for sets with larger cardinality with constant filter size. This paper shows that for sets where a distance metric can be defined, reducing the false positive rate is possible if elements to be inserted exhibit locality according to this metric. In this way, a hardware alternative to Bloom filters able to extract spatial locality features is proposed and analyzed.

1 Introduction

Bloom filters [1] were devised to test whether an element is a member of a set in a time and space-efficient way. They allow insertions of an unbounded number of elements at the cost of false positives, but not false negatives (elements can be added to the set, but not removed). A Bloom filter comprises a bit array and k different hash functions that map elements into k randomly distributed bits of the array. At first, all the array bits are set to 0. Inserting an element into the Bloom filter consists in setting to 1 the k bits given by the hash functions. Test for membership consists in checking that those k bits are asserted.

Bloom filters has been used in a wide range of applications in the domain of networks [2], file searching [6], and more recently in the domain of an emerging and promising field, Transactional Memory (TM) [5]. In this case, elements inserted into the Bloom filter correspond to memory address locations issued by a running program.

TM arises as an alternative to the conventional multithreaded programming to ease the writing of concurrent programs in multicore processors. TM introduces the concept of transaction that allows semantics to be separated from implementation. A transaction is a block of computations that appears to be executed with atomicity and isolation. Thus, transactions replace a pessimistic lock-based model by an optimistic one and solve the abstraction and composition problems.

Hardware Transactional Memory (HTM) implements most of the required mechanisms of TM at the core level. HTM systems execute transactions in parallel, committing non-conflicting ones. A conflict occurs when a memory location is concurrently accessed by several transactions and at least one access is a write. Thus, HTM systems must record all memory reads and writes during the execution of transactions in order to detect conflicts. Bloom filter signatures have been recently proposed to store the addresses of such memory reads and writes [9]. However, these signatures may exhibit high rates of false conflicts when transactions are long-running and large. False conflicts may slow down the execution significantly.

The main contribution of this paper is the design and analysis of a hardware alternative to Bloom filters that has been called the Interval Filter. Compared to a classical Bloom filter, the Interval Filter may show a lower false positive rate for those inserted elements that exhibit spatial locality according to some metric.

Hereinafter HTM is adopted in order to evaluate the proposed filter. However, results could extrapolate to other similar domains. Locality of reference will be exploited to store the locations read and written in an alternative way to Bloom filters, aiming to reduce false conflicts and enhance the execution of large transactions.

The rest of the paper is organized as follows: Section 2 defines the filter and explains how it operates. Section 3 places the Interval Filter within the Transactional Memory domain. Section 4 describes the simulation environment and evaluates the filter. Finally, conclusions are drawn in Section 5.

2 Interval Filter

The Interval Filter (IF) is proposed to reduce false positives in the presence of locality according to some metric. Without loss of generality, in the rest of the paper memory addresses will be considered as the elements to be inserted in the filter. Thus, intervals are defined as chunks of consecutive addresses that can be extracted out of a memory reference trace. Fig. 1 shows the design of the filter. IF comprises n intervals that are recorded as a pair of two full addresses, one representing the lower bound of the interval and the other one representing the upper bound. A valid bit per interval is also needed, V_0, \dots, V_{n-1} . Each interval bound has two bit lines. Lower bounds are compared with the incoming address incremented by one and upper bounds are compared with the address decremented by one. Hence, $=_0^l, \dots, =_{n-1}^l$ return true if the incremented address is equal to the corresponding lower bound of the interval. On the other hand, $>_0, \dots, >_{n-1}$ return true if the address is greater than the lower bound. Likewise, $=_0^u, \dots, =_{n-1}^u$ and $<_0, \dots, <_{n-1}$ are the bit lines for the upper bounds of the intervals. The filter can be thought of as an extended full-associative cache.

Same primitive operations than Bloom filters can be performed with the interval filter. Fig. 1 shows, within dash-line boxes, how test for membership and insertions can be implemented. Test for membership consists in checking the

Match line to be true. This output line is computed by checking that the incoming address is within an interval. To do so, $>_0, \dots, >_n$ and $<_0, \dots, <_n$ bit lines can be used in the way shown in Fig. 1. Thus, lookups are relatively fast but insertions are slower and more complicate, as in Cuckoo-Bloom filters [9]. Actually, three cases come up on inserting an address into the interval filter, given that the address is not a member yet. Fig. 2 depicts the insertion algorithm flow chart:

Case 1. If every $=_0^l, \dots, =_{n-1}^l$ and $=_0^u, \dots, =_{n-1}^u$ bit lines are zero it means that neither valid intervals can be expanded so the incoming address must form a new interval in the filter. Thus, if the filter is not *Full* then the address is inserted into a non-valid interval by storing the original address (neither incremented nor decremented) in both bounds, lower and upper. Conversely, if the filter is *Full* then a valid interval is widen introducing false positives. In order to minimize the number of false positives due to widening, the closest interval bound is chosen. To do so, the address is XORed with the bounds whose $>$ or $<$ bit lines are set to 0. Then, the lower one is chosen as the candidate to store the address in an iterative manner.

Case 2. If either only one lower bound or only one upper bound is equal to the incremented/decremented address then an existing interval is to be widen. This can be done in a straightforward way by only storing the original address into the matched bound.

Case 3. If one lower bound and one upper bound are matched at the same time it means that the incoming address is the one who left to merge two existing intervals. Therefore, one of the two matched intervals is invalidated by clearing its V bit and the remaining interval is widen by setting its lower/upper bound to the lower/upper bound of the invalidated interval. In Fig. 2 the invalidated interval is i and it has been matched in the upper bound so the lower bound of the interval k is set to the lower bound of i . If k is chosen to be invalidated then the upper bound of i is set to the upper bound of k .

3 Application to Transactional Memory

A TM system must record the trace of memory references read and written within each transaction to detect conflicts among transactions. These addresses are classified into two separate sets, the Read Set (RS) and the Write Set (WS), which are also known as the signature of the transaction.

In the beginning, such signature was implemented by adding transactional Read/Write bits to each block in the cache memory. However, modifying caches to track transactional information have been proved to pose major constraints into TM virtualization since transactions are limited to cache sizes, scheduling time-slice (quantum), migration problems,... Also, cache memories are critical fine-tuned structures that should not be modified by including additional hardware. Therefore, Ceze et al. [4] proposed a signature implementation with per-thread Bloom filters fulfilling next requirements: signatures must not tolerate false negatives (undetected true conflicts) but may assume false positives

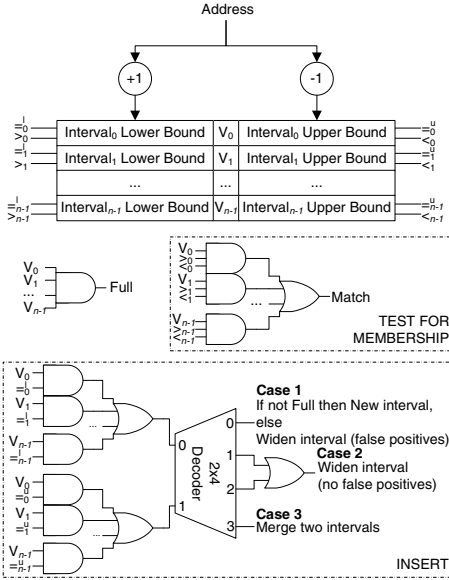


Fig. 1. Interval Filter Design

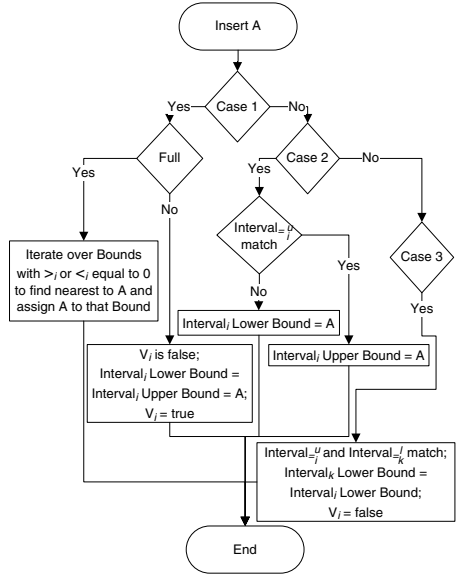


Fig. 2. Insertions Flow Chart

(false conflicts); RS and WS sizes are unknown in advance, therefore, signatures should not limit the number of addresses to be stored; and test and insertion of an address should be fast operations.

Interval filters also fulfill the cited requirements. Moreover, they classified the trace of memory references into intervals to extract spatial locality features and to try to keep read and write sets as concise as possible. Therefore, the IF can be used instead of a Bloom filter to record the RS and WS of large transactions exhibiting locality.

4 Experimental Evaluation

This section is devoted to the experimental evaluation of the IF comparing its performance with that of a Bloom filter of similar hardware complexity. The simulation environment used to evaluate the IF comprises the Simics [7] full system execution-driven simulator and the GEMS's Ruby module [8] that implements the Transactional Memory system. Simics simulates the SPARC architecture and it is able to run an unmodified copy of a Solaris operating system. Solaris 10 has been installed on the simulated machine and all workloads run on top of it. Ruby has been modified to include the IF. The base CMP system consists of 16 in-order, single-issue cores. Each core has a 32KB split, 4-way associative, 64B block private L1 cache. L2 cache is unified, 8MB capacity, 16-bank, 8-way associative, and 64B block size. Cache-coherence implements the MESI protocol and maintains an on-chip directory which holds a bit vector of sharers.

All workloads used are part of the Stanford’s STAMP suite [3]. This suite is designed for Transactional Memory research and includes a wide range of applications laying emphasis on those with long-running transactions and large read and write sets. Such benchmarks are of special interest for signature evaluation since they put the most pressure on signatures.

Synopsys and CACTI 5.3 [10] were used to estimate the area of the Interval Filter and the Bloom filter involved in the evaluation. A SRAM memory with 8-byte words and four separate read/write ports was modeled with CACTI to estimate the Bloom filter area. CACTI was also used to model a full-associative SRAM memory with 32-bit words and 2 banks for the IF. Additional control logic and extra comparators and incrementers used by the IF were modeled with Synopsys and have been proven to have a small impact on the total area. Given an IF with $n = 10$ (i.e. ten intervals), the hardware-equivalent Bloom filter has 4 hash functions of the class H3 (H3 has proven better than others like Bit Selection [9]) and 2048 bits length. Both filters take about $0.09mm^2$ of die area each, using 65nm technology node. Hereinafter, results will be shown for an $n = 10$ interval filter compared to a 2048 bits, 4 hash function Bloom filter.

Experiments were carried out with 4 benchmarks from the STAMP suite: Bayes, Kmeans, Labyrinth and Yada. Such benchmarks exhibit the largest transaction data sets that cause Bloom filters to slowdown the execution because of false conflicts. Table 1 summarizes input parameters and the maximum and average RS/WS size in cache lines for those benchmarks.

Table 1. Parameters and data set maximum and average sizes

Bench	Input	# xact	$\overline{max}_{ RS }$	$\overline{max}_{ WS }$	$\overline{ RS }$	$\overline{ WS }$
Bayes	-v32 -r4096 -n2 -p20 -s0 -i2 -e2	536	2171	1631	81.8	45.1
Kmeans	-m40 -n40 -t0.05 -i random-n1024-d1024-c16	1380	134	65	99.7	48.5
Labyrinth	-i random-x48-y48-z3-n64	158	529	510	128.7	120.7
Yada	-a20 -i dots.2	1338	578	405	60.5	37.5

The motivation behind the Interval Filter comes from the Fig. 3. This figure shows the interval histograms for the four benchmarks. It is a classification of the intervals by width, both in read sets and write sets of transactions, and it does not necessarily mean that every transaction in the system has intervals of every width. All the benchmarks show some amount of single addresses, i.e. width-1 intervals, but the most of the addresses can be classified into intervals wider than 1 by extracting spatial locality features. In fact, the number of single addresses in the benchmarks is between 2% and 22% as shown in Table 2. To keep track of address sets as intervals instead of doing so as single addresses could save in space and performance.

Table 2. Percentage of single addresses

Bench	Number of single addresses	
	Read Set	Write Set
Bayes	13.1%	2.7%
Kmeans	2.7%	2.5%
Labyrinth	4.2%	3.6%
Yada	22.0%	16.4%

Fig. 5 shows the execution time of the Bloom filter versus the IF normalized to the perfect filter (i.e. infinite length, no false positives). Two cases can be observed:

- *Bayes and Yada*: Interval filter performs similar or slightly worse than Bloom filters concerning these benchmarks. Two things cause such slowdown: (i) the high percentage of single addresses, see Table 2, and (ii) transactions are made up of small-mid size intervals, as can be inferred from Fig. 3, since the largest interval in Bayes is about 100 addresses long in the figure, while the largest transaction is 2171 addresses long (see Table 1). Also, for Yada, the largest interval is 11 addresses, while the largest transaction is 578 addresses long. Therefore, having an interval filter with $n = 10$ intervals and a great amount of intervals to be stored in it, then “Case 1” (see Fig. 1) will be the most frequent hence introducing lots of false positives.

Another important fact to consider is the creation of the intervals. Fig. 4 shows the interval creation in the write set of the largest transaction in each benchmark, i.e. it shows the result of having an infinite size interval filter in which addresses are inserted in order of appearance and the number of valid intervals are checked out after each insertion and then plotted. Notice that Bayes and Yada would need between 200 and 350 intervals to keep track of the whole set without false positives, however the interval filter used is 10 intervals. Flat parts in the Bayes curve corresponds to “Case 2” in Fig. 1.

- *Labyrinth and Kmeans*: Interval filter performs equal or better than Bloom filters for these benchmarks. Now the number of single addresses is lower than Bayes and Yada (Table 2) and large transactions are made up of a few large-size intervals, since Fig. 3 shows intervals greater than 400 addresses for Labyrinth while Table 1 shows maximum transactions about 500 and, 70 addresses intervals for Kmeans and maximum transactions of 70 and 130. Therefore, the interval filter does not get full immediately introducing few false positives. Fig. 4 shows a flat creation of intervals for Kmeans while Labyrinth shows a rise and fall that corresponds to interval merging, “Case 3” in Fig. 1.

The behavior of these benchmarks is due to the data types they manage. Labyrinth makes a copy of a global multidimensional mesh inside a transaction which is represented as a multidimensional array. Kmeans keep a table of objects and attributes within an array. Conversely, Bayes and Yada use more complex and memory-scattered data structures as trees and lists.

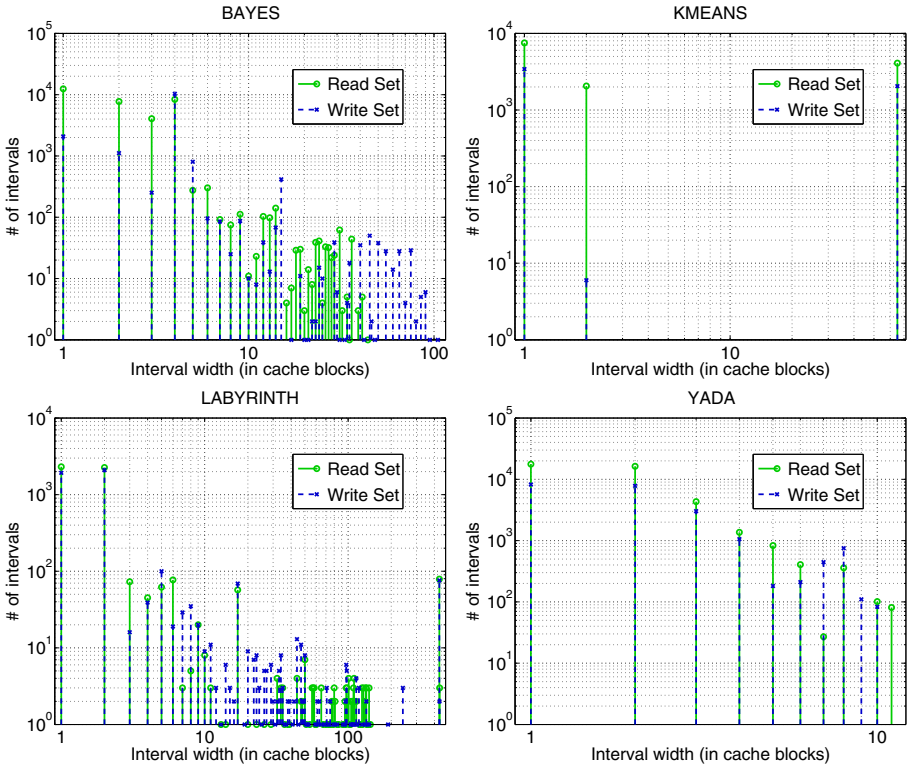


Fig. 3. Number of intervals of different widths for Bayes, Kmeans, Labyrinth and Yada, both RS and WS (log scale)

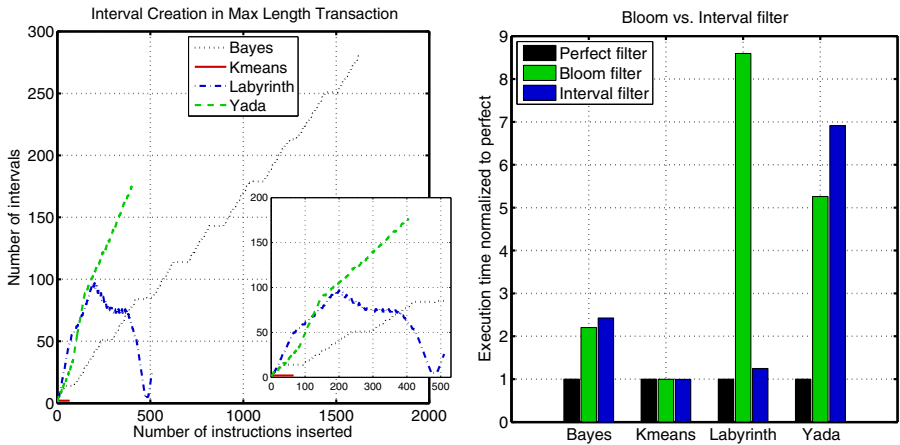


Fig. 4. Write set interval creation for maximum length transactions of Bayes, Kmeans, Labyrinth and Yada

Fig. 5. Execution time of Bayes, Kmeans, Labyrinth and Yada normalized to the Perfect filter

5 Conclusions

This paper proposes the classification of elements in a metric space into intervals to store them in a concise manner. For that purpose, a new Interval Filter hardware architecture has been developed that reduces false positives in the presence of locality as an alternative to Bloom filters. As case of study, the Interval Filter is evaluated in the context of Transactional Memory. The proposed filter is able to record contiguous addresses without introducing false positives. The Interval Filter presents excellent results when there exist few large intervals although it may perform similar or worse than regular Bloom filters for data streams with poor locality features.

Acknowledgment

This work has been supported by the Ministry of Education of Spain with project CICYT TIN2006-01078.

References

1. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM* 13(7), 422–426 (1970)
2. Broder, A., Mitzenmacher, M.: Network applications of bloom filters: A survey. *Internet Mathematics* 1(4), 485–509 (2004)
3. Cao Minh, C., Chung, J., Kozyrakis, C., Olukotun, K.: STAMP: Stanford Transactional Applications for Multi-Processing. In: *IEEE Int'l Symp. on Workload Characterization, IISWC'08* (2008)
4. Ceze, L., Tuck, J., Torrellas, J., Cascaval, C.: Bulk disambiguation of speculative threads in multiprocessors. In: *33th Ann. Int'l. Symp. on Computer Architecture (ISCA'06)*, pp. 227–238 (2006)
5. Herlihy, M., Moss, J.E.B.: Transactional memory: Architectural support for lock-free data structures. In: *20th Ann. Int'l. Symp. on Computer Architecture (ISCA'93)*, pp. 289–300 (1993)
6. Jimeno, M., Christensen, K.J., Roginsky, A.: Two-tier bloom filter to achieve faster membership testing. *Electronics Letters* 44(7), 503–504 (2008)
7. Magnusson, P.S., Christensson, M., Eskilson, J., Forsgren, D., Hallberg, G., Hogberg, J., Larsson, F., Moestedt, A., Werner, B., Werner, B.: Simics: A full system simulation platform. *IEEE Computer* 35(2), 50–58 (2002)
8. Martin, M.M.K., Sorin, D.J., Beckmann, B.M., Marty, M.R., Xu, M., Alameldeen, A.R., Moore, K.E., Hill, M.D., Wood, D.A.: Multifacet's general execution-driven multiprocessor simulator GEMS toolset. *ACM SIGARCH Comput. Archit. News* 33(4), 92–99 (2005)
9. Sanchez, D., Yen, L., Hill, M.D., Sankaralingam, K.: Implementing signatures for transactional memory. In: *40th Ann. IEEE/ACM Int'l Symp. on Microarchitecture (MICRO'07)*, pp. 123–133 (2007)
10. Thoziyoor, S., Muralimanohar, N., Ho Ahn, J., Jouppi, N.P.: CACTI 5.1. Technical Report HPL-2008-20, HP Labs (2008)

Histogram Distance for Similarity Search in Large Time Series Database

Yicun Ouyang¹ and Feng Zhang²

¹ Software School of Sun Yat-sen University
Guangzhou, 510006, China
ouyangyicun@163.com

² Software School of Sun Yat-sen University
Guangzhou, 510006, China
Jeff.F.Zhang@gmail.com

Abstract. Dynamic Time Warping (DTW) has been widely used for measuring the distance between the two time series, but its computational complexity is too high to be directly applied to similarity search in large databases. In this paper, we propose a new approach to deal with this problem. It builds the filtering process based on histogram distance, using mean value to mark the trend of points in every segment and counting different binary bits to select the candidate sequences. Therefore, it produces a more appropriate collection of candidates than original binary histograms in less time, guaranteeing no false dismissals. The results of simulation experiments prove us that the new method exceeds the original one.

1 Introduction

Nowadays time series is playing a more and more critical role in many different databases for financial and medical purposes. The experts interested in dealing with it focus their work on similarity search.

To compute the distance between two sequences of different lengths, which is critical for similarity search, DTW was first introduced by Berndt and Clifford [1] to database community. It had been widely used by many researchers since then because of its convenience for similarity search in the real databases. Whereas, there are still some confusion in DTW [2]. Furthermore, as the size of databases dramatically increases, the effectiveness of DTW is much lower than ever before.

In this paper, we introduce a simple but efficient approach, which can be applied to dealing with time series of large databases. It is based on the original binary histograms and has a better performance. Considering the points' orders and high-density, we divide all time series into segments, using mean value to mark the trend of points in each segment and counting different binary bits to select the candidate sequences. It improves the time complexity and precision, so we will get more appropriate candidate set in less time, making similarity search more rapid and convenient. Through our several experiments, the new algorithm exceeds the original one in large extent. The structure of this paper is

described as follows: In Section 2, we introduce the background and related work in details. In section 3, we propose our new approach called histogram distance. Section 4 shows the whole process of similarity search. Section 5 consists of our experiments and the results described by the figures. Finally, in section 6, we draw some important conclusions and give suggestions for the future work.

2 Background and Related Work

The symbols we use in this paper have been defined in the table 1.

Table 1. List of symbols

Symbol	Definition
S	a time series of database
Q	a query time series
S_i^{seg}	the i -th segment of time series S
$S_i^{\text{seg}} \cdot \text{Val}$	the value of the i -th segment of time series S
S_{new}	the new time series
H	the binary histograms
H_i	the i -th element of binary histograms
HD	the histogram distance method
BH	the binary histograms method
$range$	the subinterval width of range

There are some main methods for similarity search supported by dynamic time warping. The first one Native-Scan has the time complexity $O(|S| * |Q|)$ as it scans all the sequences of the database.

Constructing the index structure by FastMap [3] is the second way. It maps a sequence of length n into a k -dimensional point. Nevertheless, this would probably produce some false dismissals.

To overcome the drawback of FastMap, namely avoiding any dismissals, Park et al. [4] proposed ST-Filter to use the suffix tree as an index structure. At present, the index problem of time series still attracts much attention of many experts [5][6][7].

Yi et al. [3] combines a lower bounding function LB-Scan with the FastMap method to balance the efficiency and accuracy of the similarity search process. Fu et al. [8] also coordinate the lower bounding technique with multi-dimensional indexing to increase the efficiency of searching.

To avoid the false dismissals and improve the efficiency of computing distances, Some experts [9][10] proposed their own lower bounding functions. Kim et al. [11] respectively distract the first, last, greatest and smallest element from each sequence, defining the distance of sequences as the maximum of difference of each pair. Except for lower bounding functions, Gu and Jin [12] introduce a variation of the traditional histograms for consulting the distance of different time series before computing the real DTW distance.

3 Filter Process

In this section, we will select all the candidate time series which are within the threshold of histogram distance from the whole database.

3.1 New Time Series

Assuming that the average length of the time series of the database is n and the number is m , then the complexity of constructing the corresponding binary histograms of all these time series is $O(2mn)$. We can use the method time series segment to get the new time series, decreasing the complexity of binary histograms constructing process. Specifically, according to the Time Interval (TI) by the user, we divide the original time series into several segments with the same length, and the i -th part is called time series segment S_i^{seg} . It is defined by two parts: TI and $S_i^{\text{seg}} \cdot Val$. Assuming the time interval is 5 ($TI = 5$), we only map $S_i^{\text{seg}} \cdot Val$ rather than all five points to the according subinterval. The complexity of constructing binary histograms is clearly reduced to $O(mn + \frac{mn}{TI})$. Since the new time series uses the mean value of the original ones, it's reasonable that the precision of calculating process will drop. However, this problem can be well solved by defining a new criterion for evaluating similarity, and that has been proved by our experiments.

Definition 1. *The value of the i -th segment of time series is:*

$$S_i^{\text{seg}} \cdot Val = \frac{Val_i^1 + Val_i^2 + \dots + Val_i^{TI}}{TI} \quad (1)$$

In (1), Val_i^k denotes the value of the k -th element in S_i^{seg} .

Definition 2. *The new time series is:*

$$S_{\text{new}} = \{S_1^{\text{seg}} \cdot Val, S_2^{\text{seg}} \cdot Val, \dots, S_i^{\text{seg}} \cdot Val, \dots\} \quad (2)$$

Equation(2) suggests us the dimension of new time series is much less than the original one.

3.2 Histogram Distance

J. Gu and X. Jin [12] proposed a method called binary histograms to adapt traditional histograms to DTW. In our paper, the binary histograms approach is abbreviated to BH. Assuming that a time series $x = \{x_1, x_2, \dots, x_n\}$ has the maximum x_{max} and minimum value x_{min} , and the range $[x_{\text{min}}, x_{\text{max}}]$ could be divided into several subintervals with the same length. We will easily get the binary histograms H through mapping all elements of time series to the subintervals. In $H = [H_1, H_2, \dots, H_m]$, H_i denotes that whether there is at least one element of x appears in the i -th subinterval.

```

Algorithm calDistance (s_New, query_New, m)
//s_New=the new database time series, query_New=the new query sequence,
m=the number of subintervals
For all elements of time series s_New and query_New
//get the maximum and minimum value of all new time series
    max=Max(ts[i]);
    min=Min(ts[i]);
subintervals=Divide([min,max],m); //divide the range into m segments with the
same length
For all elements in subintervals
    if(s_New[index] ∈ subintervals[i]) bh_s[i]=1;
    if(query_New[index] ∈ subintervals[i]) bh_query[i]=1;
distance=Compare(bh_s,bh_query); //compare binary bits of bh_s and bh_query
return distance;

```

Fig. 1. Calculating Histogram Distance algorithm

In their paper, they pointed out that whether the element exists should be focused on, rather than the exact number of elements presenting in subintervals. If there is at least one element in some certain subinterval, the value of it will be given 1, or else given 0.

It's proved that BH will greatly improve the efficiency of similarity search of time series in large databases. However, there are still some flaws we can not neglect in the later work. Firstly, due to the dramatically increasing size of large databases, we have to map every points of time series to the according subinterval once. Furthermore, the filter process is too coarsen, because they consider the overall number of 1 in binary histograms as the criterion for evaluating the similarity of different time series.

Actually, the first problem can be solved by the new time series, because every segment including TI points of the original time series only needs mapping to the according subinterval once. As for the second one, we should define a new parameter as a criterion, to evaluate the differences between sequences of the database and the query one. Therefore, in this paper, we define the histogram distance, to reflect the differences. Specifically, comparing the binary histograms of all time series of database with that of the query one, we get the number of different binary bits, which is defined as the histogram distance. If some time series' distances are within the threshold given by the user, these time series will be added into the candidate set.

Figure 1 describes the algorithm for calculating histogram distance between the new database time series and the new query one.

3.3 Candidate Set

After constructing the new time series and calculating the histogram distances of them, we could select the appropriate sequences from the whole database

according to the threshold given by the user. If the distances of some time series are within the threshold given by the user, these time series should be added into the candidate set. The size of candidate set with the selected time series, can be much smaller than the database.

4 Similarity Search

While constructing the new time series, the effects on its binary histograms by noisy data can be alleviated, because we use the mean value to replace the original points. If the histogram distance between some database time series and the query one is within the threshold, this database time series should be considered as the candidate for computing the real DTW distance in the next step. After calculating the real DTW distance between all sequences in the candidates set and the query one, we can easily find the nearest time series to the query one by searching for the minimum value of the DTW distance.

4.1 Calculate Real DTW Distance

In this paper, we will not focus on improving efficiency of DTW, because the filtering procedure will not be affected by that in any case. In a general way, we adopt the method which can be considered as the original DTW distance calculating way, meaning no development will be provided while calculating the real DTW distance.

Definition 3. *DTW distance (between S and Q) is:*

$$DTW(S, Q) = DTW_{\text{base}}(F(S), F(Q)) + \min \begin{cases} DTW(S, R(Q)) \\ DTW(R(S), Q) \\ DTW(R(S), R(Q)) \end{cases} \quad (3)$$

In (3), $F(S)$ is the first element of the time series S , $R(S)$ denotes the rest elements of the time series S except the first one.

Recursion method can not be applied to this process directly because of the large length of the time series coming from the real world. Therefore, we use the for-loop and dynamic programming to replace it while computing the DTW distance.

4.2 Nearest Search

After calculating the DTW distance between each time series in the candidate set and the query one, we will return the sequence with the smallest distance. It is actually using the K-Nearest Search method under the condition $K=1$. The procedure of similarity search, considering histogram distance as the selecting criterion and DTW distance as the distance function, is described as follows:

Step 1. Construct the new time series of each sequence in the database and the query one.

Step 2. Calculate the histogram distance between new time series of each sequence in the database and the query one.

Step 3. Construct the candidate set according to the threshold given by the user.

Step 4. Calculate the DTW distance between each time series in the candidate set and the query one.

Step 5. Search for the time series with the nearest DTW distance to the query one.

5 Experiment

In this section, we have done some experiments to record the precision and the consuming time of the new method proposed. The data used in these experiments consists of two parts: daily closing prices of 200 stocks in Shanghai stock market from 2003 to 2010 and the synthetic time series produced by the random program. The stock price data can be freely downloaded from this website: <http://table.finance.yahoo.com/table.csv?s=000000.ss>, in which the number 000000 denotes the stock code. The random data varies between 0 and 100 with the relatively low fluctuation.

5.1 Candidate Ratio

We select all the time series within the threshold of histogram distance from the whole dataset, considering them as the candidates for calculating the real DTW distance. Therefore, candidate ratio [5][6] is one of the most significant parameters for reflecting the performance of filtering. As shown in (4), this figure is the smaller, the better.

Definition 4. *The Candidate Ratio is:*

$$\text{candidate ratio} = \frac{\text{the number of the candidate time series}}{\text{the number of all time series}} \quad (4)$$

In Fig.2 and Fig.3 we can see the candidate ratio of two different method: binary histograms and histogram distance. The time series of these experiments respectively come from the stock data or the random program. The parameter TI is defined as 4 and the range as 0.2. Apparently, the candidate ratio is increasing with the climb of the threshold of histogram distance. Furthermore, the candidate ratios of HD are always lower than BH when threshold varies.

5.2 Cost Time

Consuming time is the basic index for assessing the performance of an algorithm. In this paper, it consists of the time for filtering the time series database and computing the real DTW distance. In Fig.4 and Fig.5, the goal of testing is converted from candidate ratio to the cost time of the whole program. The pictures show in detail that how long HD and BH will find the nearest time series to the query one. The result tells us that the time consumed by HD is always less than BH. The parameters we used in the experiments here can be described as follows: $TI=4$, $range=0.2$, $slide\ window\ width=20$.

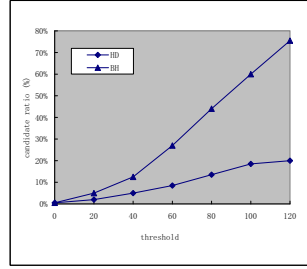
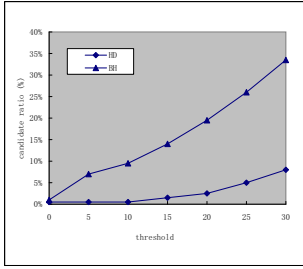


Fig. 2. Candidate Ratio on Stock Data

Fig. 3. Candidate Ratio on Synthetic Data

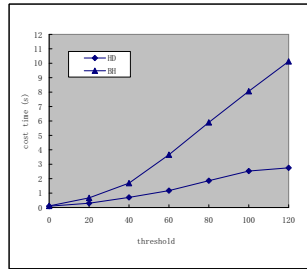
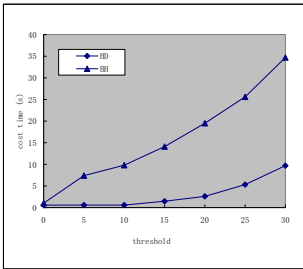


Fig. 4. Cost Time on Stock Data

Fig. 5. Cost Time on Synthetic Data

6 Conclusions

In this paper, we propose a new approach named histogram distance, and the results of experiments prove that our new method greatly increases the precision of filtering and lessens the consuming time of the whole program. In the future, we will also search for an appropriate method, such as the new lower bounding function, to replace the original DTW distance function in a fast and convenient way.

References

1. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: AAAI-94 Workshop on Knowledge Discovery in Databases (KDD), New York, USA, pp. 229–248 (1994)
2. Ratanamahatana, C.A., Keogh, E.: Three myths about dynamic time warping data mining. In: SIAM Conference on Data Mining (SDM), Newport Beach, USA, pp. 506–510 (2005)
3. Yi, B.K., Jagadish, H.V., Faloutsos, C.: Efficient retrieval of similar time sequences under time warping. In: 14th International Conference on Data Engineering (ICDE), Orlando, USA, pp. 201–208 (1998)
4. Park, S., Chu, W., Yoon, J., Hsu, C.: Efficient searches for similarity subsequences of different lengths in sequence databases. In: 16th International Conference on Data Engineering (ICDE), Los Angeles, USA, pp. 23–32 (2000)

5. An, J.Y., Chen, H.X., Furuse, K., Ohbo, N., Keogh, E.: Grid-based indexing for large time series databases. In: Liu, J., Cheung, Y.-m., Yin, H. (eds.) IDEAL 2003. LNCS, vol. 2690, pp. 614–621. Springer, Heidelberg (2003)
6. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. *Knowledge and Information Systems* 7, 358–386 (2005)
7. Ruengronghirunya, P., Niennattrakul, V., Ratanamahatana, C.: Speeding up similarity search on a large time series dataset under time warping distance. In: 13th Pacific-Asia Conference on Knowledge and Data Mining (PAKDD), Bangkok, Thailand, pp. 981–988 (2009)
8. Fu, A.W.C., Keogh, E., Lau, L.Y.H., Ratanamahatana, C.A., Wong, R.C.W.: Scaling and time warping in time series querying. *VLDB Journal* 17, 899–921 (2008)
9. Lemire, D.: Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recognition* 42, 2169–2180 (2009)
10. Aßfalg, J., Kriegel, H., Kröger, P., Renz, M.: Probabilistic similarity search for uncertain time series. In: 21st International Conference on Scientific and Statistical Database Management (SSDBM), New Orleans, USA, pp. 435–443 (2009)
11. Kim, S.W., Park, S., Chu, W.W.: An index-based approach for similarity search supporting time warping in large sequence databases. In: 17th International Conference on Data Engineering (ICDE), Heidelberg, Germany, pp. 607–614 (2001)
12. Gu, J., Jin, X.M.: A simple approximation for dynamic time warping search in large time series database. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 841–848. Springer, Heidelberg (2006)

The Penalty Avoiding Rational Policy Making Algorithm in Continuous Action Spaces

Kazuteru Miyazaki

National Institution for Academic Degrees and University Evaluation,
1-29-1, Gakuennichimachi, Kodaira-city, Tokyo 187-8587, Japan
teru@niad.ac.jp
<http://svrrd2.niad.ac.jp/faculty/teru/index.html>

Abstract. Reinforcement learning involves learning to adapt to environments through the presentation of rewards - special input - serving as clues. To obtain quick rational policies, profit sharing, rational policy making algorithm, penalty avoiding rational policy making algorithm (PARP), PS-r* and PS-r# are used. They are called *Exploitation-oriented Learning* (XoL). When applying reinforcement learning to actual problems, treatment of continuous-valued input and output are sometimes required. A method based on PARP is proposed as a XoL method corresponding to the continuous-valued input, but continuous-valued output cannot be treated. We study the treatment of continuous-valued output suitable for a XoL method in which the environment includes both a reward and a penalty. We extend PARP in the continuous-valued input to continuous-valued output. We apply our proposal to the pole-cart balancing problem and confirm its validity.

1 Introduction

Among machine-learning approaches, reinforcement learning (RL) focuses most on goal-directed learning from interaction [8]. It is very attractive because it uses dynamic programming (DP) to analyze behavior. RL generally uses rewards as teaching signals in learning. DP-based RL involves optimizing behavior under reward signals designed by RL users.

Despite important applications [6], RL is difficult to design to fit real-world problems because, first, interaction requires too many trial-and-error searches and, second, no guidelines exist on how to design reward signal values. While these are essentially neglected in theoretical papers, they become serious issues in real-world applications, e.g., unexpected results arise if inappropriate values are assigned to reward signals [3].

We are interested in approaches treating reward signals independently and not assigning them value. We also want to reduce the number of trial-and-error searches by strongly enhancing successful experience — a process known as exploitation-oriented learning (XoL) [5]. Examples of learning systems belonging to XoL are the rational policy making algorithm (RPM) [2], the penalty avoiding rational policy making algorithm (PARP) [3], PS-r# [5] and so on.

XoL features four factors: (1) With conventional DP-based methods, reward signal values must be suitably designed, while XoL treats them as independent signals, letting rewards be handled more intuitively and easily than concrete values [3]. (2) XoL learns more quickly by strongly tracing successful experiences. (3) XoL is effective in classes beyond MDPs because it is a Bellman-free [8] method. (4) XoL does not pursue optimality efficiently, which can be acquired by multistart [2] resetting all memory to get a better policy.

Apply RL to real environments, however, requires humongous information on states mostly taking continuous values. This can be handled by approximating states using function approximation and so on [7,8,11,9]. Especially, in RPM and PARP, a method that discretizes state using a basis functions is known [4]. However, continuous-valued output cannot be treated yet.

We study the treatment of continuous-valued output suitable for a XoL method in which the environment includes both a reward and a penalty. We extend PARP in the continuous-valued input [4] to continuous-valued output. We show the effectiveness of our proposal using the pole-cart balancing problem.

2 The Domain

2.1 Notation

After perceiving state input from the environment, an agent selects and executes an action. Time is discretized by one input-action cycle. Input from the environment is called a state. The pair consisting of a state and an action selected in the state is called a rule. Rewards and penalties based on a series of actions are received from the environment, and a reward is given to the state or action that caused the transition to the state in which our purpose is achieved and a penalty given to the state or corresponding action in which our purpose is not achieved. Rewards and penalties are treated independently, eliminating the need for sophisticated design of their values, although conventional RL system based on DP requires it.

A rule series that begins from a reward/penalty state or an initial state and ends with the next reward/penalty state is called an episode. The number of states in an episode is called an episode length. If an episode contains rules of the same state but paired with different actions, the partial series from one state to the next is called a detour. A rule always existing on a detour is called an irrational rule, and otherwise a rational rule. A rule that directly receives a penalty is called a penalty rule. If all selective rules for a state are penalty or irrational rules, the state is called a penalty state. If a destination resulting after selecting a rule enters a penalty state, the rule is also classified as a penalty rule. A function that maps states to actions is called a policy. A policy with a positive amount of reward acquisition expectations is called a rational policy, a rational policy receiving no penalty is called a penalty avoiding rational policy.

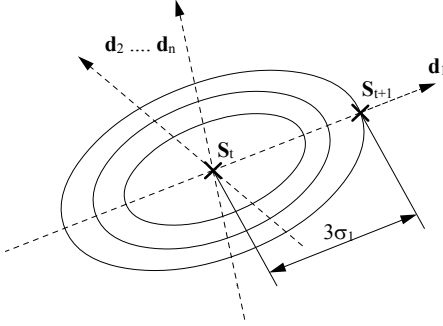


Fig. 1. Basis Function

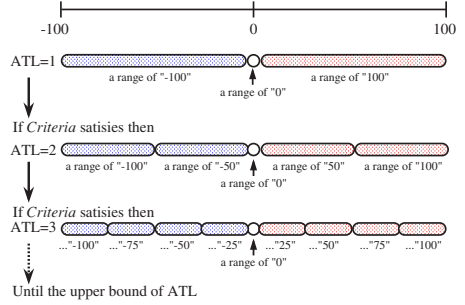


Fig. 2. An example of updating mechanism of the action types level (ATL)

2.2 Continuous State Space Discretization

To discretize continuous state spaces using basis functions [4], let state (n -dimensional vector) at time t be \mathbf{S}_t , a selected action by \mathbf{S}_t be a_t , and the resulting transition destination be \mathbf{S}_{t+1} , as shown in Fig. 1. A basis function is created by an n -dimensional normal distribution function with current state \mathbf{S}_t at the center. The principal axis direction of the function is defined by $\mathbf{S}_{t+1} - \mathbf{S}_t$, the d_1 axis in Fig. 1. Directions of other axes d_2, \dots, d_n are generated using Gram-Schmidt orthonormalization to mutually intersect at right angles, the d_2, \dots, d_n axes in Fig. 1.

Principal axis range extent σ_1 is given by $3\sigma_1 = |\mathbf{S}_{t+1} - \mathbf{S}_t|$, and extent of the range of other axes σ_i are given by $3\sigma_i = \frac{|\mathbf{S}_{t+1} - \mathbf{S}_t|}{\sqrt{n}}$ ($i = 2, 3, \dots, n$). Because a function biased along an experienced direction is obtained, the range extent of other than the principal axis is multiplied by $\frac{1}{\sqrt{n}}$. $3\sigma_i$ covers 99% of samples.

Center \mathbf{S}_t of the generated basis function is called the basis function state, or μ . The action that generates the basis function is memorized together with the basis function and is used when selecting an action. The basis function corresponds to a rule under discrete state spaces. The basis function corresponding to the penalty rule is called a penalty basis function, and the other basis function is called a nonpenalty basis function. Initially a basis function is generated as an unlabeled basis function. When a learning agent obtains a penalty, we can find a penalty basis function by the *Penalty Basis Decision* (PBD) procedure proposed in the paper [4]. On the other hand, when the agent obtains a reward, we can find a nonpenalty basis function using an episode memory [2].

If values of an observation at time t are given by \mathbf{y} , then the returned value is given as:

$$f(\mathbf{d}) = \exp \left\{ -\frac{1}{2}(\mu - \mathbf{y})^T T \Sigma T^T (\mu - \mathbf{y}) \right\}, \tag{1}$$

$T = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n]$, $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2)$, and \mathbf{t}_i is a unit vector along the d_i axis for $i = 1, 2, \dots, n$. Eq.(1) becomes 1.0 when \mathbf{y} coincides with μ , and the farther \mathbf{y} is away from μ , the smaller the value of Eq.(1). Calculating the value

of Eq(II) for each memorized basis function for given observation \mathbf{y} enables the closeness of the observation to each basis function to be compared.

Each basis function scope is controlled by a threshold. The control parameter is called f_para ($e^{-\frac{9}{2}} \leq f_para \leq 1.0$) and attached to each basis function. If f_para becomes large, the scope of the function is narrowed and if f_para becomes 1.0, it coincides with center μ . $f_para = e^{-\frac{9}{2}}$ means that the basis function range is expanded to transition destination \mathbf{S}_{t+1} .

If the value of Eq(II) for basis function k calculated for the state observed at one point in time exceeds the attached f_para , the state is said to “match” with basis function k and the value of Eq(II) at that time is called $match_value$.

3 Proposal of the Penalty Avoiding Rational Policy Making Algorithm in Continuous Action Spaces

In this section, we study the treatment of continuous-valued output suitable for a XoL method. It will be an uncommon case that we have to output from a fine-grained value, for example $F = 35.14$ [N] and so on, in the initial learning phase. Therefore we will output rough values at first, and the values are subdivided as needed. We introduce a parameter called *the action types level (ATL)* to control the grained level of the action. It is updated to output more grained value, if some criteria, that is designed based on our purpose, is satisfied.

We show an example of the updating mechanism using Fig. 2. In the first, the initial value of *ATL* is 1 where a learning agent can output 3 types action, that is $\{-100.0, 0.0, 100.0\}$. It is increased to 9,17,33,65,129,257 and 513 types action as updating *ATL* from 2 to 9. *ATL* is updated when the agent transits to the state where there are penalty basis functions only, or there is no reward between $ini_r_step \times MSN$ steps, where ini_r_step is the number of steps to obtain a reward at first time. The criteria should be designed more suitably based on an application area.

Fig. 3 is the Penalty Avoiding Rational Policy Making algorithm in continuous state and action spaces with the updating mechanism of *ATL*. We call it **pure-cPARP**. If our purpose of the learning has been achieved, pure-cPARP can return *Performance* that is the number of actions to achieve the purpose. The last two “else if” in Fig. 3 is used to restart the learning. Especially, the last one is prepared for learning a better set of basis functions efficiently.

Furthermore, we can update the *Performance* by multistart method [2],[5] as shown in Fig. 4. We call it **cPARP**. **cPARP** can find a known best *Performance* termed *BestPerformance* in Fig. 4.

4 Numerical Experiment

4.1 Pole-Cart Balancing Problem

We confirm the effectiveness of our proposal using the pole-cart system in Fig. 5, where m is the mass of a pendulum ($m = 0.1kg$), M the sum of pendulum and

```

procedure pure-cPARP(BestPerformance)
begin
  Define MAXB for controlling the number of bases functions (BFs)
  and MSN for updating ATL
  do
    Initialize all BF's
    Set Performance  $\leftarrow$  0, ini_r_step  $\leftarrow$  0, ATL  $\leftarrow$  1, UpperATL  $\leftarrow$  0
    do
      Perceiving an sensory input
      do
        Matching with the set of BF's
        if There is an action that does not belong in a penalty BF then
          if There is an action that belongs in a nonpenalty BF then
            Select the action, break
          else if There is an action that belongs in an unlabeled BF then
            Select the action, break
          else Select the action and Generate a new BF, break
        else
          if All actions are matching with a penalty BF in the center then
            Increase ATL
            if ATL reaches to the upper bound then UpperATL  $\leftarrow$  1
            else Increase f_para of the BF that has the least matching value
            break
          while
            if UpperATL == 1 then, break
          Update an episode by the sensory input and the action
          if Obtain a reward then
            Unlabeled BF's on the episode is registered to a nonpenalty BF
            if Obtain a reward at first time then
              Set ini_r_step  $\leftarrow$  the episode length required to obtain the reward
            Initialize the episode
          else if Obtain a penalty then
            Update the set of penalty BF's by PBD [4] and Initialize the episode
          if Our purpose has been achieved then
            Performance  $\leftarrow$  the number of steps to achieve the purpose, break
          else if There is no reward between ini_r_step  $\times$  MSN steps then
            Increase ATL
            if ATL reaches to the upper bound then, break
            else if The number of BF's is larger than MAXB then, break
            else if BestPerformance > 0 AND
              The number of steps based on nonpenalty BF's has been continued
              more than BestPerformance then, break
          while
            while(Performance == 0)
          return(Performance)
    end
  end

```

Fig. 3. Procedure pure-cPARP

```

procedure cPARP
begin
  Define MAXM for controlling the number of times of executing multistart
  Set BestPerformance  $\leftarrow$  0, i  $\leftarrow$  0
  do
    Performance  $\leftarrow$  pure-cPARP(BestPerformance)
    if Performance < BestPerformance OR BestPerformance == 0
      then BestPerformance  $\leftarrow$  Performance, i++, print(BestPerformance)
  while(i < MAXM)
end

```

Fig. 4. Procedure cPARP

cart mass ($M = 1.1kg$), $2L$ pendulum length ($2L = 1.0m$), and F force exerted on the cart. The initial location is the state in which the pendulum hangs straight down and the cart at the center. Four-dimensional continuous values are given for sensory input: { location of the cart (x), velocity of the cart (\dot{x}), angle of the pendulum (θ), and angular velocity of the pendulum ($\dot{\theta}$) }. The range of cart motion is $-2.4 < x < 2.4$. Beyond this range, the system returns to the initial location and the agent obtains a penalty. When the pendulum is raised and continuing within $-24^\circ < \theta < 24^\circ$, the agent obtains a reward. On the other hand, beyond this range after reaching the range, the agent obtains a penalty.

We compare our proposal with the method in the paper [4] that treats a continuous-valued input only. We prepare 3 types action { $-M, 0.0, M$ } (“3 actions”) and 5 types action { $-M, -M/2.0, 0.0, M/2.0, M$ } (“5 actions”) for the discrete action method [4], where M is 100, 200, 400 or 800. On the other hand, our proposal can output the range of $-M \leq 0.0 \leq M$ equally divided by ATL . As we can increase ATL from 1 to 9, we can use from 3 to 513 types action. It means that the action interval is $\frac{M}{256}$ in the most grained case.

Sarsa with random tiling [1], that is a DP-based method, cannot stabilize a pole, when we have designed a reward for 100.0 and a penalty for -100.0 . In general, a DP-method requires sophisticated design of these values.

4.2 Results

We have experimented 100 trials with different random seed. Table 1 shows “Performance” and “No. of Steps” by basis functions that have achieved the stabilization of a pole. “Performance” is the number of steps to raise a pole from the initial location when the stabilization has been achieved at first, and “No. of Steps” is the number of steps to learn the set of basis functions. “a3,a5,a7,a19” for our proposal is corresponding to $ATL=1,2,3,4$, respectively. The values at “a3,a5,a7,a19” are frequencies in 100 trials.

From Table 1, as M becomes larger, “3 actions” or “5 actions” have “IMPOSSIBLE” cases that fail to stabilize a pole. It means that the larger action value is difficult to realize the stabilization. It can understand with the frequency of ATL in our proposal. For example, since there is no success trial in the case of $M = 400$ and $ATL = 1$ (a3), we can understand that it is difficult to stable a pole by 400.0 or -400.0 value for action. Therefore “3 actions” on $M = 400$, whose output values are 400.0, 0.0, or -400.0 , shows “IMPOSSIBLE”.

Our proposal can attain the stabilization in any M and can find the appropriate range of action values by updating mechanism of ATL even if we set the larger M . In general, we cannot know the appropriate range of action values in advance. Therefore we can understand the importance of our proposal. Remark that “No. of Steps” will be larger than the method in the paper [4] except for the case of “5 actions” on $M = 100$, because search spaces of our method become larger than “3 actions” or “5 actions” in those cases. The search space in the case of “5 actions” on $M = 100$ will be larger than our proposal on $M = 100$ in many trials, since it can stable a pole by 3 types action { $-100.0, 0.0, 100.0$ } only at 85 trials in 100 trials.

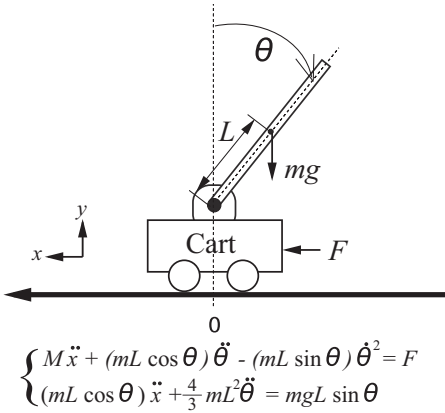


Fig. 5. Pole-cart system

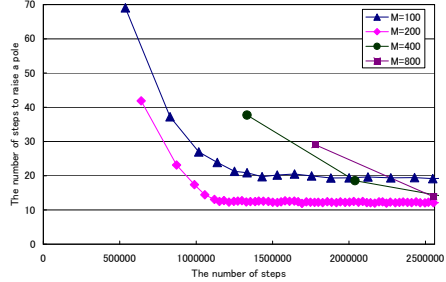


Fig. 6. Effect of the multistart method embedded in cPARP

Table 1. The results of the pole-cart balancing problem (Fig. 5)

M	Methods	Performance	No. of Steps	a3	a5	a9	a17
100	pure-cPARP	69.1	5.42×10^5	85	11	3	1
	5 actions	58.6	5.98×10^5	-	-	-	-
	3 actions	84.6	2.44×10^5	-	-	-	-
200	pure-cPARP	41.9	6.46×10^5	48	46	6	0
	5 actions	37.7	4.51×10^5	-	-	-	-
	3 actions	78.0	2.90×10^5	-	-	-	-
400	pure-cPARP	37.7	1.35×10^6	0	86	14	0
	5 actions	43.7	2.66×10^5	-	-	-	-
	3 actions	IMPOSSIBLE		-	-	-	-
800	pure-cPARP	29.1	1.80×10^6	0	0	85	15
	5 actions	IMPOSSIBLE		-	-	-	-
	3 actions	IMPOSSIBLE		-	-	-	-

Next, we try to confirm the effect of the multistart method embedded in cPARP. Fig. 6 is a result of it. It shows the number of steps to raise a pole plotted against the number of steps when the stabilization has been achieved. The number of steps to raise a pole is improved as taking a step. It means that we can improve the set of basis functions by the multistart method.

5 Conclusion

A method based on PARP is proposed as a XoL method corresponding to the continuous-valued input, but continuous-valued output could not be treated. We have studied the treatment of continuous-valued output suitable for a XoL method and proposed pure-cPARP and cPARP. We have applied our proposal to the pole-cart balancing problem and confirmed its validity.

We plan to extend our proposal to environments having several types of reward signals. We also plan to find an efficient application.

References

1. Kimura, H.: Reinforcement Learning in Multi-Dimensional State-Action Space Using Random Rectangular Coarse Coding and Gibbs Sampling. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 88–95 (2007)
2. Miyazaki, K., Kobayashi, S.: Learning Deterministic Policies in Partially Observable Markov Decision Processes. In: Proceedings of the 5th International Conference on Intelligent Autonomous System, pp. 250–257 (1998)
3. Miyazaki, K., Kobayashi, S.: Reinforcement Learning for Penalty Avoiding Policy Making. In: Proceedings of the 2000 IEEE International Conference on Systems, Man and Cybernetics, pp. 206–211 (2000)
4. Miyazaki, K., Kobayashi, S.: A Reinforcement Learning System for Penalty Avoiding in Continuous State Spaces. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 11(6), 668–676 (2007)
5. Miyazaki, K., Kobayashi, S.: Exploitation-oriented Learning PS-r#. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 13(6), 624–630 (2009)
6. Merrick, K., Maher, M.L.: Motivated Reinforcement Learning for Adaptive Characters in Open-Ended Simulation Games. In: Proceedings of the International Conference on Advanced in Computer Entertainment Technology, pp. 127–134 (2007)
7. Santamaria, J.C., Sutton, R.S., Ram, A.: Experiments with Reinforcement Learning in Problems with Continuous State and Action Spaces. *Adaptive Behavior* 6(2), 163–218 (1998)
8. Sutton, R.S., Barto, A.G.: “Reinforcement Learning: An Introduction. A Bradford Book. MIT Press, Cambridge (1998)
9. Tateyama, T., Kawata, S., Shimomura, Y.: A Reinforcement Learning Algorithm for Continuous State Spaces using Multiple Fuzzy-ART Networks. In: Proceedings of SICE-ICCAS 2006, pp. 2445–2450 (2006)

Applying Clustering Techniques to Reduce Complexity in Automated Planning Domains

Luke Dicken and John Levine

Strathclyde AI and Games Group,
Department of Computer and Information Science
University of Strathclyde, Glasgow, Scotland
{luke, johnl}@cis.strath.ac.uk

Abstract. Automated Planning is a very active area of research within Artificial Intelligence. Broadly this discipline deals with the methods by which an agent can independently determine the action sequence required to successfully achieve a set of objectives. In this paper, we will present initial work outlining a new approach to planning based on Clustering techniques, in order to group states of the world together and use the fundamental structure of the world to lift out more abstract representations. We will show that this approach can limit the combinatorial explosion of a typical planning problem in a way that is much more intuitive and reusable than has previously been possible, and outline ways that this approach can be developed further.

Keywords: Clustering, Automated Planning, Reducing Complexity.

1 Introduction

1.1 Motivation

The problem of complexity in the area of Automated Planning is a major stumbling block towards finding good and, importantly, efficient algorithms that can efficiently take a description of a problem and the goals to be achieved and return a list of the actions that must be performed. Part of the issue lies in the richness of the languages used to describe planning problems. The Planning Domain Description Language (PDDL) was originally introduced to describe purely propositional STRIPS-style worlds and problems within these worlds [15], but has since been extended to provide a much richer selection of tools to a domain modeller such as numeric effects, duration of actions [9], independent changes in the world at specific times [6] and continuous change of numerics over time [8]. This richness of language brings with it a host of problems to an already complex task, with PDDL now able to represent domains with a PSPACE-Complete complexity class (although it is worth noting that as most benchmark problems are designed by humans, and indeed most problems encountered tend to be at some level human-solvable, the actual complexity class utilised tends towards NP-Hard) [11].

Even at its most basic, planning suffers from combinatorial explosion. From the initial state, a number of actions may be applicable which each lead to further states and from these, actions are also applicable, meaning that as the expansion progresses, the number of states will explode as n^k where n represents the current depth into the algorithm and k represents the average number of actions applicable in a state. As a result, work has focused on trying to reduce the explosion, either by finding ways to limit k or to divide the problem into a sequential solving of x sub-problems each of size approx. $\frac{n}{x}$. This will be discussed below.

Our approach is based on also reducing the size of n , but where previous attempts have been problem-specific at run-time, we have taken the view that a domain-specific technique that can be executed in advance will be a much more efficient approach. To this end, we have applied a clustering technique over a representation of the domain in order to group particular states together. This gives several advantages, as it turns a single monolithic planning task into two distinct problems, firstly one of finding paths across the clusters, and secondly of finding a path from the initial state to a goal state across the clusters. This gives an abstracted problem to solve and then a sequence of smaller problem instances than can be solved relatively trivially.

1.2 Paper Outline

The remainder of this paper is structured as follows. In Section 2 we will give an overview of the background to the problem being tackled, and describe approaches that have attempted to solve this or similar problems. Section 3 will cover a detailed specification of how our approach can be implemented. We will present results and analysis in Section 4 from our experimentation, and discuss the implications of these. Finally, in Section 5 we will discuss how this technique can be expanded in the future, and draw conclusions based on what we have outlined.

2 Related Work

2.1 Automated Planning

Automated Planning is the name given to the sub-discipline of Artificial Intelligence concerning agents that can reason independently about their environment and create plans of actions that must be executed in order to achieve a given set of objectives. In order for the agent to perform this sort of reasoning, it requires three things to be supplied :

- A complete description of the manner in which the environment of the agent works, a description of the types of object within the environment and the actions that are possible in the environment, with associated conditions that must be true before an action can be applied (for example, it would not be possible to load a package onto a truck if both these objects were not at the same location) and the effects that the action will have in the world.

- A full description of the initial state of the world that the agent will be acting in.
- A partial description of the key facts that must hold in the end state, in other words, a list of the goals to be achieved.

The agent then reasons to attempt to build a list of actions that will allow it to reach one of potentially many states of the world in which the goals hold true. This process can be seen as a tree search with the initial state as the root node of the tree and the actions that can be applied in a given state (i.e. those whose preconditions have been met) forming the branches of the tree. Note that this is a slightly oversimplified representation as planning tasks can be cyclical (loading a parcel onto a truck then unloading it gives an identical state at depth 2 in the tree as was initially at depth 0), and are broadly non-terminating (few states in planning tasks exist for which no further actions can ever be applied).

As was noted previously, PDDL is the standard language in which the descriptions about the world are supplied. This formalism uses a predicate calculus to describe a propositional representation of the world, in which facts are made true or false. This gives a representation very well suited to a tree-search style approach, in that it has a high-dimensionality but each dimension is defined with just two values.

A full examination of work in this area is outwith the scope of this paper, however as a brief overview work in Automated Planning is primarily focused in three core areas. Some research aims to find better strategies to effectively traverse the search tree [17]. Other work has emphasised discovering new heuristics with which to evaluate nodes within the search tree in order to more accurately guide the search algorithm [3]. Finally, tree pruning has also been a major topic of research, in the belief that some branches within the search tree can be disregarded as irrelevant to the search, thus artificially constraining the growth of the tree [12].

Due to the computational complexity of this process, increasing emphasis is being placed on finding approaches that use alternative methods to make the problem easier to solve. One such method is the ordering of the goal conditions, with the theory being that the order in which goal resolution is tackled might give a varying hardness to the search [5]. As a brief example of this, consider a scenario in which the goal is to both to have purchased a car and to have driven to London. Here, purchasing a car and then using it to drive to London is clearly a more optimal approach than purchasing some other sort of vehicle, driving to London and then selling the current vehicle to purchase a car. This is a somewhat artificial example, but it serves to highlight the core concept.

2.2 Landmark Analysis

Another approach that has been utilised is the identification of what are known as “Landmarks” [16]. Landmarks in this context are states which are shown to be necessary in any plan which will satisfy the goals of the problem - more formally, in a temporal logic this can be described as $\diamond\{fact1 \wedge fact2 \wedge \dots \wedge factn\}$ or

at some point in the future, the facts that describe the landmark state must all be true, although it is not necessary for these facts to remain true subsequently. These landmarks are therefore waypoints that any plan must pass through on the route to a goal state. In this way, they frame the start and end points of smaller subproblems and break the task into a sequence of smaller chunks that can be solved independently. The issue with this approach is that this is a problem specific technique - in order to use landmarks to reduce the complexity of the planning problem, the landmarks must first be identified which in itself is a complex problem. This does not generalise, the landmarks found are instance specific and do not provide insight into the domain in general, although Porteous et al [16] demonstrated this can lead to significant increases in efficiency when computing specific plans.

3 Methodology

Our contention is that, rather than focusing on the problem, we can achieve simpler computation - and more generalisable results - by focusing on the domain itself. We assess the structure of the world without regard for the specific goals to be achieved in that world, meaning that the computation is reusable for any task being achieved within the same “world”, where the world is defined as being equivalent to the PDDL “domain” and that part of the problem file that describes what objects exist within the problem. Our approach is independent of both the initial state of the world and the goal state.

3.1 The Domain Transition Graph

As noted above, the PDDL formalism gives a high-dimensional formalism, with each dimension having exactly two possible values. SAS+ is an alternative formalism [1] based around the use of multi-valued variables. Where PDDL uses a set of (formally) unconnected true/false statements to represent different aspects of the world, SAS+ takes those propositions that represent related concepts (derived by analysis of mutual exclusions and potential interactions within the domain) and turns them into a single variable, with each value representing one of the previous facts that could be true under the propositional representation.

Consider the previously mentioned example involving packages and trucks. A single package can be at one of several locations or it could be in a truck. Under a PDDL representation, this would give a sequence of propositions that could be true or false :

$$at(package1, location1), \dots, at(package1, locationN), in(package1, truck1)$$

The beauty of SAS+ is that this high-dimensional data becomes vastly reduced. We can intuitively see that a package can only be in one place at a single time – whether *at* a location or *in* a truck makes no real difference – but PDDL does not adequately express the relationship these propositions have. However, automated analysis can reveal quickly that only one of these facts will ever be true at once, and so this can be represented simply by one variable :

`location_package1` \in $\{location1, \dots, locationN, truck1\}$

Moreover, if we further constrain the domain in the PDDL definition by defining three actions, one to load a package into the truck, one to unload and the third to move the truck between locations, we can capture this sequence – implicit in the PDDL – explicitly in the SAS+, again through automated translation [10]. This allows us to define a notional ordering of the values the SAS+ variables can take, and we can represent this relationship in the form of a graph, with each node representing a value the variable can take and the edges representing the actions that allow transition between these values. Each edge still has associated preconditions, now termed Causal Links, essentially defining what node must be active in each of the other graphs representing the full world in order for the edge to be able to be transitioned. This graph is called the Domain Transition Graph or DTG and is the core structure we use in our approach to complexity reduction.

3.2 Clustering the DTG

The DTG gives a representation of the structure of the world we are dealing with. Rather than looking at reducing complexity for a given problem, we use the DTG to allow us to decompose the world itself, by grouping together nodes within the graph that are within the same neighbourhood. We do this by using a Fuzzy Clustering algorithm, specifically the Fuzzy c -Means algorithm [2]. This algorithm assumes that datapoints to be clustered are defined within a coordinate system; however the DTG only reflects nodes and edges, it does not give any sense of the absolute position of the node. Because of this it was necessary to alter the implementation of the algorithm to cater to our representation in conceptual space. In particular, we define the centroid of a cluster to be the node for which the average distance to every node within the cluster, weighted proportional to that node’s strength of belonging to the cluster is minimised. The distance between nodes is calculated in advance as a look-up table using Dijkstra’s Algorithm [4] and stored as a matrix listing the shortest distance from each node to every other node in the graph, assuming that each edge connecting nodes has a notional “length” of one. Our implementation of the algorithm is given in the following equations. Eqn. 1 formally defines the centroid of each cluster k , whilst eqn. 2 defines the weight with which node x belongs to cluster k as proportional to the distance from x to the centroid of k compared against the other clusters. The value m in this equation gives the degree of “fuzziness” and must be greater than 1 - for the results presented below a value of 2 has been used.

$$\forall k, \exists C_k := C_k = \operatorname{argmin}_a (\sum_x (\operatorname{distance}(x, a) * w_{xk}) / (\sum_x (w_{xk}))) \quad (1)$$

$$\forall x, k := w_{xk} = \Sigma_j (\operatorname{distance}(C_k, x) / \operatorname{distance}(C_j, x))^{-2/(m-1)} \quad (2)$$

Initially the weights are chosen randomly, with the only restriction being that the weights of a particular node must sum to a total of 1. The centroids are updated in accordance with these weights using eqn. 1, and then the weights

are updated with their new values based on the new values for the centroids as defined in eqn. 2. These steps are repeated until the results stabilise, at which point clustering is complete.

The number of clusters that are identified by the algorithm, that is to say the maximum value k can take, is determined by eqn. 3, where n is the number of nodes within the graph. This is considered to be a “good rule of thumb” for clustering techniques [14].

$$k = \lceil \sqrt{(n/2)} \rceil \quad (3)$$

We use a fuzzy clustering technique because as much as we want to group nodes together into the clusters, we are also very interested in those nodes that lie at the interface. We describe these nodes as Focal Nodes, and define them to be those nodes which belong to two or more clusters with a weight between 0.3 and 0.7. These nodes form the entry and exit points of the clusters and are therefore important elements of our drive for complexity reduction.

3.3 Overcoming Non-determinism

The Fuzzy c -Means algorithm is a non-deterministic algorithm due to the random seed weights supplied to initialise the process. Because we are defining the centroid of a cluster to be a node, in order for the centroid to change from one node to another, the incremental difference between iterations needs to be sufficient to step across the discrete space – a drawback not encountered with this algorithm in a continuous coordinate space. As a consequence, the algorithm will not necessarily converge on the same solution each time, and some solutions will be inappropriate. Testing showed that in some cases, clusters were found to be exactly duplicated, such failure being immediately rejectable by mandating that all clusters must have distinct centroids. This did not overcome the problem completely though, and we were forced to fall back to a solution inspired by the Byzantine Generals problem [13]. As the failure rate was around 10% on the problems we tested, we were able to implement a system that executed the clustering a number of times and chose the result that the majority of samples agreed with.

4 Results

We have tested our approach on a number of different DTGs, both “homogeneous” and “heterogeneous”. Homogeneous DTGs are those which clearly have a clustering in which each cluster is equivalent, whilst heterogeneous DTGs have an inherent inequality or imbalance in their structure, making the clusters derived intuitively be non-equivalent. Results from testing are shown in Table 1. In this table a “Successful” clustering is one in which the intuitive answer has been derived. A “Satisfactory” clustering will be fit for purpose but is not the intuitive result – in some ways this may be deceptive, as our intuition as to what is and is not an ideal result is based on a knowledge of the domain that generated the DTG, and may not be mathematically valid, especially in the heterogeneous

Table 1. Results of clustering on homogeneous and heterogeneous DTGs

	Successful %	Satisfactory %	Failed %
Homogeneous	81	15	4
Heterogeneous	0	84	16
Total	40.5	49.5	10

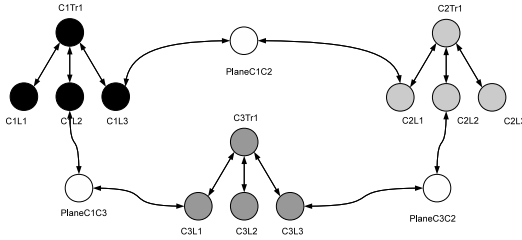


Fig. 1. Example of successful clustering across a homogeneous DTG

cases, where it was found that no clustering matched our preconceived notions, although 84% of results were found to be usable classifications. As can be seen, using the approach outlined in this paper, we are able to generate an effective clustering in 90% of cases, which can easily be compensated for in practice using the majority voting technique mentioned above. An example of a successful clustering on a homogeneous graph is shown in Figure 1, which is a small extension of the domain described previously in which Locations are grouped into Cities, Trucks can only travel between Locations within a single City and Planes allow for travel between two specific locations in different Cities. This is a standard benchmark world within Automated Planning literature known as “Logistics”.

5 Conclusions

In this paper we have presented exploratory work into developing a new approach to reducing complexity in Automated Planning through the use of problem-independent clustering. Using this technique, we have shown that the world can be decomposed into regions in a similar manner to that seen in Hierarchical Task Networks [7], but generated dynamically from the structure of the world, rather than by crafting a domain-specific library of decompositions of actions. This decomposition remains relevant for any combination of initial state and goals within the world, making it a highly reusable technique and calculable *a priori*. Additionally, it should be noted that despite being intended to be calculated offline, the technique is highly efficient being merely a number of calculations. This grouping of nodes within the world gives us a natural approach to subdividing a large and complex problem, and shows great promise as a technique for reducing complexity in solving Automated Planning problems. Future work will investigate this further, as well as assessing what other uses the identified “Focal Nodes” could have in related areas.

References

1. Bäckström, C., Nebel, B.: Complexity results for SAS+ planning. *Computational Intelligence* (1995)
2. Bezdek, J.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Dordrecht (1981)
3. Coles, A., Fox, M., Long, D., Smith, A.: Additive-disjunctive heuristics for optimal planning. In: *Proceedings of the International Conference on Automated Planning and Scheduling* (2008)
4. Dijkstra, E.: A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 279–271 (1959)
5. Drummond, M., Currie, K.: Goal ordering in partially ordered plans. In: *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (1989)
6. Edelkamp, S., Hoffman, J.: PDDL 2.2: the language for the classical part of IPC-04. In: *Proceedings of the International Conference on Automated Planning and Scheduling* (2004)
7. Erol, K., Hendler, J., Nau, D.: HTN planning: Complexity and expressivity. In: *Proceedings of the National Conference on Artificial Intelligence* (1994)
8. Fox, M., Long, D.: PDDL+ level 5: An extension to PDDL2. 1 for modelling planning domains with continuous time-dependent effects. Technical Report, U. of Durham (2001)
9. Fox, M., Long, D.: PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research* (2003)
10. Helmert, M.: A Planning Heuristic Based on Causal Graph Analysis. In: *Proceedings of the International Conference on Automated Planning and Scheduling* (2004)
11. Helmert, M.: *Understanding Planning Tasks: Domain Complexity and Heuristic Decomposition*. LNCS (LNAI), vol. 4929. Springer, Heidelberg (2008)
12. Hoffmann, J., Nebel, B.: The FF Planning System: Fast plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research* (2001)
13. Lamport, L., Shostak, R., Pease, M.: The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems* (1982)
14. Mardia, K., Kent, J.: *Multivariate Analysis* (1979)
15. McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., Wilkins, D.: PDDL The Planning Domain Definition Language. Technical Report, Yale Center for Computational Vision and Control (1998)
16. Porteous, J., Sebastia, L., Hoffmann, J.: On the extraction, ordering, and usage of landmarks in planning. In: *Proc. European Conf. on Planning* (2001)
17. Vidal, V.: A lookahead strategy for heuristic search planning. In: *Proceedings of the International Conference on Automated Planning and Scheduling* (2004)

The M-OLAP Cube Selection Problem: A Hyper-polymorphic Algorithm Approach

Jorge Loureiro¹ and Orlando Belo²

¹ Departamento de Informática, Escola Superior de Tecnologia e Gestão,
Instituto Politécnico de Viseu, Campus Politécnico de Repeses, 3505-510 Viseu, Portugal
jloureiro@di.estv.ipv.pt

² Departamento de Informática, Escola de Engenharia, Universidade do Minho
Campus de Gualtar, 4710-057 Braga, Portugal
obel@di.uminho.pt

Abstract. OLAP systems depend heavily on the materialization of multidimensional structures to speed-up queries, whose appropriate selection constitutes the cube selection problem. However, the recently proposed distribution of OLAP structures emerges to answer new globalization's requirements, capturing the known advantages of distributed databases. But this hardens the search for solutions, especially due to the inherent heterogeneity, imposing an extra characteristic of the algorithm that must be used: adaptability. Here the emerging concept known as hyper-heuristic can be a solution. In fact, having an algorithm where several (meta-)heuristics may be selected under the control of a heuristic has an intrinsic adaptive behavior. This paper presents a hyper-heuristic polymorphic algorithm used to solve the extended cube selection and allocation problem generated in M-OLAP architectures.

Keywords: Hyper-Heuristics; Distributed Data Cube Selection; Multi-Node OLAP Systems Optimization.

1 Introduction

Today, we know that the success of *On-Line Analytical Processing* (OLAP) systems depends largely on their multidimensional visions' mechanisms and on their support for fast query answering, independently of the aggregation level of the required information. Yet, they also carry some seeds of failure. In fact, answering an OLAP query may imply the scanning and aggregation of huge amounts of data, something that is often incompatible with the inherent on-line characteristic. The materialization of multidimensional structures, denoted as materialized views or subcubes, was devised as the answer to such problem, and has been a condition of performance. However, the increasing needs of OLAP users forced these structures to attain an inordinate size and complexity, implying new approaches to their optimization, beyond the classical cube selection solutions, as greedy heuristics [4], genetic approaches [8], or hill climbing with simulated annealing [6]. So, we think that distribution is the key.

Accompanying the trends of organizations' infrastructures, and given the principle of locality, the distribution of the materialized structures will increase the probability of local satisfaction of OLAP needs, having several advantages: a sustained growth of processing capacity without an exponential increase of costs, an increased availability of the system, and the avoidance of bottlenecks. Here, we focus in only one distribution type, something that we denoted as a *Multi-node OLAP* (M-OLAP) architecture, generated by the distribution of the OLAP cube by several storage and processing nodes, named *OLAP Server Nodes* (OSN), with a known processing power and storage space, inhabiting in close or remote sites, interconnected by communication facilities, available to the (possibly spreaded) users' community. Conventional cube selection problem deals only with the appropriate selection of the materialized structures, attending to a given query profile. Now, a new factor is included into the optimizing equation: space. In fact, it is not enough to select the most beneficial subcubes; they also have to be conveniently located.

The optimization of the distributed OLAP approach needs an extended cost model, which was proposed in [1], as the distributed aggregation lattice. This work also proposed a distributed node set greedy algorithm that addressed the distributed view selection problem, being shown that this algorithm has a superior performance than the corresponding standard greedy algorithm, using a benefit per unit space metric. In [10], this distributed lattice framework is used, but extended, to include real communication cost parameters and processing node power, which led to heterogeneity in the nodes and the network. This model was used as a framework to build a simulated OLAP environment, which disposes a set of estimation cost algorithms [10] that used the intrinsic parallel nature of the distributed OLAP architecture and time as the cost unit. Framed on this simulated environment, several metaheuristics have been applied to the selection and allocation of cubes in M-OLAP systems: genetic co-evolutionary [11] and *Discrete Particle Swarm Optimization* (Di-PSO) [9]. It was also used a simulated annealing metaheuristic, but using a more comprehensive cost model framing the simulated OLAP environment. This work pursued the research that has been conducted, but trying another approach based on two evidences: 1) it is known that each optimizing heuristic (or meta-heuristic) has virtues (and some limitations); one side of its approach was the design of a huge amount of hybrids, which try to collect the better of two worlds: compensate the weaknesses of a given algorithm with the strengths of another; in this approach we have two (meta-) heuristics which are applied in a pre-defined sequence; 2) but the concept may be extended by using a bundle of (meta-) heuristics, an emerging concept known as hyper-heuristic, building an algorithm where several (meta-) heuristics may be selected under the control of a heuristic. In this paper, we'll try to solve a general optimizing problem, as we have a complex and heterogeneous environment, the M-OLAP architecture.

2 Hyper-heuristic Approach

According to Burke et al. [2], hyper-heuristics are defined as a procedure of "using (meta-)heuristics to choose (meta-)heuristics to solve the problem in hand". At first sight, this emerging approach operates on macro-level. But, when we are facing very

complex and heterogeneous problems, even for the same kind of problem, the differences between two specific cases can be huge, and, as said previously, given its specificities, a new approach can be tried, operating at a lower granularity. But the scheme can be even further refined: the referred micro-level acting scheme is not simply to use one of the available meta-heuristics in different algorithm's running-time phases, controlled by the hyper-heuristic. Nor, like hybrids and memetic algorithms [12], which use search strategies simultaneously (in the same main iteration) operating at population level. All *Search Agents* (SA) use one strategy and then the other, a sequence that is repeated over and over again. Now, the hyper-heuristic controls the mix and balancing of metaheuristics, managing the transmutation process. In this paper, we are interested in the referred micro-level acting of the hyper-heuristic. We deal with a specific kind of problem, the cube selection problem. But, as it was said, we intend to work with a generalized OLAP architecture, where we can have several OLAP nodes, interconnected with a possibly heterogeneous communication network. The workbench we developed was designed as a simulated environment where we can perform the experimental study of the performance of several metaheuristics. This way, we have transposed all previously developed metaheuristics to this workbench. But our aim was, not only to have a tool which allows us to study the behavior of each metaheuristic by himself, but, and referring the discussion above, perform new studies in the field of the hyper-heuristics. In fact, it is expected that the transposition of the referred transmutation strategy to optimizing algorithms will create a self-adaptive search heuristic in which each search agent (having the candidate solution) can assume one of the different algorithms' shapes. In this algorithm, each search agent can be a swarm particle of a *Particle Swarm Optimization* (PSO) algorithm, a genetic individual of a population of a *Genetic Algorithm* (GA), or a *Hill Climber* (HC) whose movement is managed by a simulated annealing accepting approach.

3 Hyper-polymorphic M-OLAP Algorithm

The algorithm we have designed (Fig. 1) has two main sections: an external shell, which corresponds to the controlling hyperheuristic and several subsidiary services and an inner loop, which consists of the several metaheuristics that run and transmute. The controlling hyper-heuristic is in charge of triggering the transmutation of the avatar of each search agent. On this version, this triggering occurs simply when after a given number of iterations a search agent hasn't improved the quality of its solution. Moreover, the transmutation process controlled by the hyperheuristic is deterministic: the sequence of avatars is always the same (as shown in Fig. 1 as circular arrows). The others tasks performed by the outer shell correspond to a set of services needed by the controlling center of the algorithm or the hyperheuristic itself.

3.1 M-OLAP Cube Selection Problem Coding into Controlled Metaheuristics

The problem that we intend to solve may be defined like this:

Definition 1: *M-OLAP cube selection problem.* Let $Q=\{q_1,\dots,q_n\}$ be a set of queries with access frequencies $\{f_{q_1},\dots,f_{q_n}\}$, query extension $\{qe_1,\dots,qe_n\}$; let update frequency and extension be $\{fu_1, \dots, fu_n\}$ and $\{ue_1,\dots,ue_n\}$, respectively, and let S_{Ni} be

the amount of materializing space by OLAP node i . A solution to the selection and allocation problem is a set of subcubes $M=\{s_j, \dots, s_n\}$ with a constrain $\sum_j |s_{jN_i}| \leq S_{N_i}$, where $\sum_j |s_{jN_i}|$ is the materializing space of all subcubes S_j in node N_i , so that the total costs of answering all queries Q and maintaining M , $Cq(Q, M)+Cm(M)$ are minimal.

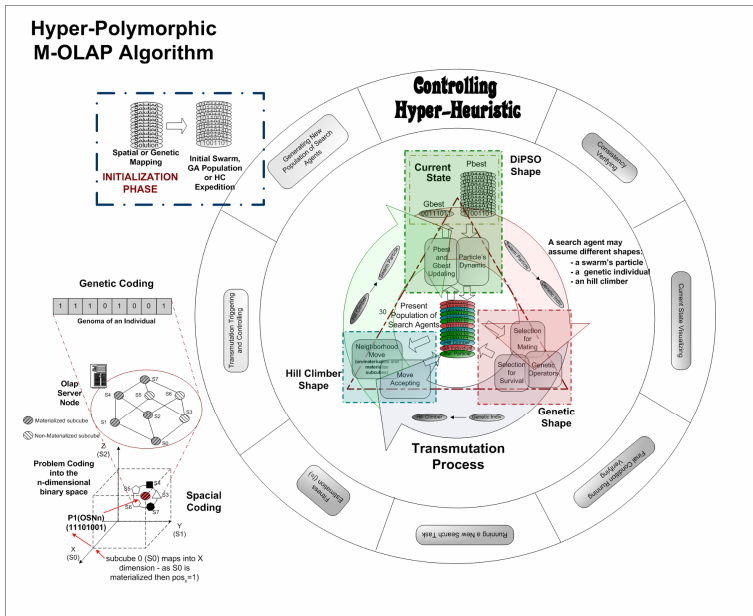


Fig. 1. Functional architectural scheme of Hyper-Polymorphic M-OLAP algorithm

Summing up, the algorithm we propose has to find the optimal set of sub-cubes to materialize in each of the nodes, which forms the M-OLAP architecture. In this case, each sub-cube is materialized or not: we assume, for this problem, that the low level granularity of the materialization is the sub-cube (we do not consider a partial materialization). We can see in Fig. 1 that the inner loop of the algorithm has three component meta-heuristics: a particle swarm, a genetic and a simulated annealing approach. Two (the first and the third) problem solvers have a natural space paradigm problem's coding: we must map each possible M (subset of the materialized sub-cubes) into the position of a *Hill Climber* (HC) or *Swarm Particle* (SP). Each possible sub-cube may or may not be materialized. Mapping this into the space paradigm, if we have a binary space, with a number of dimensions equal to the maximal number of possible sub-cubes in M ($nS=n.Ls$, where n is the number of OSNs and Ls is the number of sub-cubes into the lattice), each position of a hill climber represents a distribution M of sub-cubes (a solution to the M-OLAP cube selection problem). This is represented in the left lower corner of Fig. 1, where a multidimensional space for an OLAP node with 8 possible sub-cubes is shown. The representation must be replicated for each node of the M-OLAP architecture. A position=1 for a dimension d_i , means that the corresponding sub-cube in M is materialized and conversely. E.g., in Fig. 1,

sub-cube S_0 is mapped into the X dimension: as S_0 is materialized, the HC or the particle is at a 1 position. Summarizing, as the search space has d dimensions, the position of the hill climber is coded by a binary string where each bit is then mapped to a sub-cube that may (or not) be materialized into each node. Concerning to the other meta-heuristic (genetic algorithm) we have a binary string, that represents the genoma, which must be mapped to the solution. A one in the genoma means that the corresponding sub-cube is materialized; a zero means the opposite. The coding is also shown in the left side of Fig. 1, for the same sub-cubes' node distribution. Further details of each of the meta-heuristics, problem coding, controlling parameters and also related to the participant algorithms' architecture can be found in [5], [7], and [9].

3.2 Algorithm's Architecture

Given the limitation of space we can't show the diagram with the static structure of the Hyper-Polymorphic M-OLAP algorithm, which was developed in Java. Summing up, as main classes, we have: 1) the class set corresponding to the populations of each SA's shape and the methods that will generate the respective behavior (particle_swarm, individuals, and expedition); 2) the class Search_Agent, a super-class, related to the search agents, responsible for the global state and consequently for the control of the triggering of the transmutation process and the verifying of its consistency (includes the hyperheuristic); 3) the class M_OLAP-State, which shows the present state and the better solution already achieved; and 4) M class, used to estimate a given solution (distribution of subcubes) proposed by any of the SAs. Many other classes were grouped in subsystems as: base parameters of the several algorithms and also the ones related to the environment, the lattice's structure, and data output services, among others.

4 Experimental Simulated Test

4.1 Working Platform

We designed a suitable set of tests in order to evaluate the viability of applying the algorithm that is proposed to the distributed OLAP cube selection problem, especially to estimate its performance and scalability. Experiments were performed using the test set of Benchmark's TPC-R [13], selecting the smallest database (1 GB), from which we used 3 dimensions (customer, product and supplier). To broaden the variety of subcubes, we added some other attributes to each dimension, generating hierarchies, as follows: customer (c-n-r-all); product (p-t-all) and (p-s-all); supplier (s-n-r-all). With this, we generated a cube, whose subcubes may be found in [9]. Whenever the virtual subcube (base relations) is scanned, this has a cost three times the subcube of lower granularity. As we know, this is a simple OLAP scheme, but a higher complexity case will be left to future work. We generated several query sets to simulate the query profile. Given the stochastic nature of the algorithm, each test was repeated 10 times, being taken the average. The three component algorithms have a variety of parameters that must be tuned. We used the values we've already used in past experiments, when each of the algorithms was used for the same problem. Some of these parameters were changed later in particular tests; each situation and the new parameters setting are referred, when appropriate.

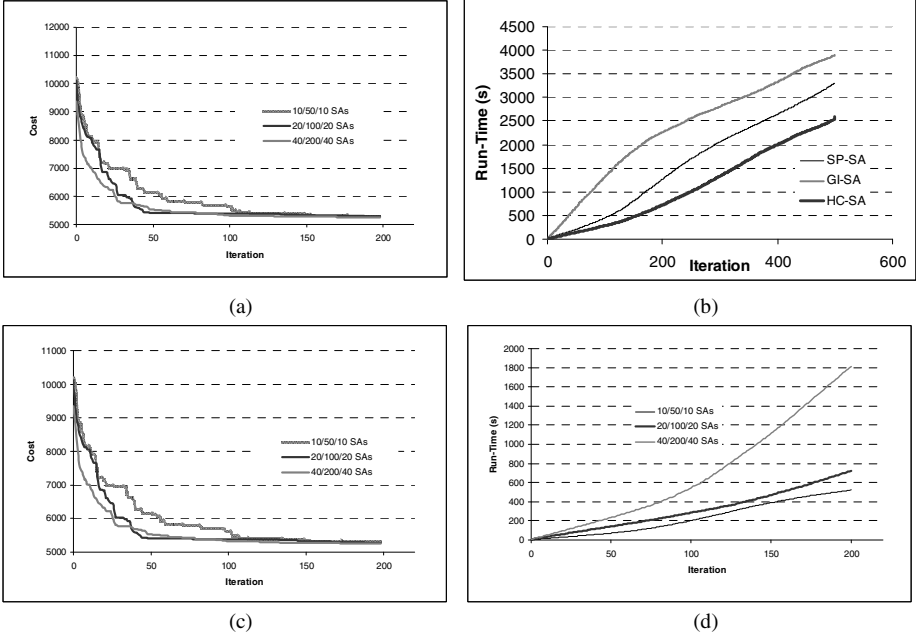


Fig. 2. Plots that show the results of the performed tests

4.2 Test Set and Results

The test set was designed intending to analyze the impact of some control parameters (of the hyper-heuristic) over the performance of Hyper-Polymorphic M-OLAP algorithm: a) the initial shape of the SAs population (which of them is the most beneficial one); b) the number of iterations without fitness improvement of the SA that triggers the transmutation process (I); c) the multiplicity (number of genetic individuals generated when a particle suffers its transmutation); d) the number of SAs; and e) the scalability, concerning to the queries number and M-OLAP OSNs number. For the first test we used a query set with 90 queries, randomly generated, and the GA standard version component. We ran the algorithm with the three possible initial shapes of SA’s population, using a multiplicity of 5 and allowing 500 iterations. The first two plots (Fig. 2) evaluate the impact of the initial shape of the SA’s population on the performance of the algorithm: the plot a) measures the quality of the achieved solutions and the plot b) the impact over the run-time of the algorithm. As we can see, the initial SA’s shape has a reduced impact on the quality of the solutions. However, a HC-SA initial shape seems to be the most beneficial, once it achieves better solutions faster. It was also observed that the HC-SA initial shape has also a positive impact over the run-time of the algorithm – Fig. 2 b). It is also evident that, for this particular problem and environmental conditions, 200 iterations are enough for the tests: beyond that iteration number, the algorithm has not achieved any significant further benefits. The second test tries to evaluate the impact of the number of SAs onto the performance of the algorithm. We used maximal populations of 50, 100 and 200 SAs, what means that, for

a maximal population of 100 SAs, we will have an initial population of 20 HC-SAs, and we may have a maximal population of 20 SP-SAs, 100 GA-SAs or 20 HC-SAs (denoted as 20/100/20) – see plot c) and d) in Fig. 2. As we can see, the number of the population of search agents has a limited impact on the quality of the solutions, although a larger population achieves good solutions faster. This behavior is possibly explained due to the positive impact that the forms SP-SA and HC-SA have, in this particular issue, since their quality of achieved solutions is relatively independent of the population size. The second plot (d) also shows that the run-time of the algorithm grows at a lower rate than the ratio of the population’s number for low populations (between 10/50/10 and 20/100/20); but the run-time has a high increase for the 40/200/40 population. This behavior may be explained by the better efficacy of GA-SA for higher individual populations, increasing the predominance of GA-SAs, and the genetic algorithm is also the most “expensive” of them all. Anyway, the quality *versus* population relation seems to bend to small populations, because the higher run-time (when using high populations) doesn’t return an important improvement on the quality of the solutions. Populations of 20/100/20 seem to be a good trade-off to use for the rest of the tests. The third test aims to evaluate the impact of I (the number of iterations without fitness improvement that triggers the metamorphosis of the SA) and m (multiplicity onto the performance of the algorithm). We used the same parameters as in the last test and a population of 20/100/20. For I , we used the values 15, 30 and 60; for m we used 1, 2 and 5. Results showed that I seems to have a reduced impact on the algorithm’s performance. However, a posterior detailed analysis showed that a value of $I=30$ or 60 seems to be a good choice for I . m seems to have a reduced impact too, although a value of $m=2$ seems to ensure the better performance. This behavior is somewhat unexpected, but, looking for possible causes, we may think that, in fact, only a part of the population is of *Genetic Individual* (GI) type, and its impact onto the solutions is limited; also, a high multiplicity disfavors the diversity, with negative impact onto the performance. We also performed a scalability test for the Hyper-Polymorphic M-OLAP concerning to the number of queries and the number of OSNs. We used 3 queries’ set (with 30, 60 and 90 queries) and two OLAP architectures: the one with 3 OSNs and another with 6 OSNs. Based on this last test, we can conclude that the number of queries has a reduced impact (only a 16% increase when the queries number increases 3 times).

5 Conclusions and Future Work

This paper proposes the Hyper-Polymorphic M-OLAP, a proposal for a new algorithm oriented for the selection and allocation of M-OLAP cubes. The algorithm also constitutes a repository of meta-heuristics, as each algorithm may be used by itself. It is important to refer that the described architecture is also a framework for M-OLAP cube selection metaheuristics. This is enforced by the design of the algorithm, which allows the easy inclusion of new heuristics. The results seem to point out that Hyper-Polymorphic M-OLAP can be a good choice for this problem. It reveals a great independence for the majority of the control parameters, showing a great adaptability. In fact, the population of each avatar of SA seems to fluctuate ensuring the adaptability of the algorithm in face of the diversity of the particular case that it has to solve. In a near future, we intend to pursue several research directions, as

the scheme is rigid: the transmutation cycle and the straightforward controlling hyper-heuristic. Some improvements can be intended towards an enhanced adaptability: 1) to include diversification mechanisms, using a probabilistic way of transferring the solution from the SA present form to the next; 2) to change the transmutation cycle, substituting the inflexible cycle of Fig. 1 by an adaptive mechanism; 3) to insert new metaheuristics into the hyper-polymorphic M-OLAP algorithm, as tabu search [3] and possibly ant colony optimization, another life inspired algorithm, which is turning out to be a good optimization method; and, finally 4) getting the best improvement could be the use of a meta-heuristic as the controlling hyper-heuristic. This strategy can implement all adaptive mechanisms: every controlling parameter can be dynamically adjusted, according to the instantaneous real-time performance of the algorithm and each one of the compound meta-heuristics.

References

1. Bauer, A., Lehner, W.: On Solving the View Selection Problem in Distributed Data Warehouse Architectures. In: Proceedings of the 15th International Conference on Scientific and Statistical Database Management (SSDBM'03), pp. 43–51. IEEE, Los Alamitos (2003)
2. Burke, E., Hart, E., Kendall, G., Newall, J., Ross, P., Schulenburg, S.: Hyper-Heuristics: An Emerging Direction in Modern Search Technology. In: Glover, F., Kochenberger, G. (eds.) Handbook of Meta-Heuristics, pp. 457–474. Kluwer, Dordrecht (2003)
3. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Dordrecht (1997)
4. Harinarayan, V., Rajaraman, A., Ullman, J.: Implementing Data Cubes Efficiently. In: Proceedings of ACM SIGMOD, Montreal, Canada, pp. 205–216 (1996)
5. Holland, J.H.: Adaptation in Natural and Artificial Systems, 2nd edn. MIT Press, Cambridge (1992)
6. Kalnis, P., Mamoulis, N., Papadias, D.: View Selection Using Randomized Search. Data Knowledge Engineering 42(1), 89–111 (2002)
7. Kennedy, J., Eberhart, R.C.: A Discrete Binary Version of the Particle Swarm Optimization Algorithm. In: Proc. of the 1997 Conference on Systems, Man and Cybernetics (SMC'97), pp. 4104–4109 (1997)
8. Lin, W.-Y., Kuo, I.-C.: Genetic Selection Algorithm for OLAP Data Cubes. Knowledge and Information Systems 6(1), 83–102 (2004)
9. Loureiro, J., Belo, O.: A Discrete Particle Swarm Algorithm for OLAP Data Cube Selection. In: Proc. of the 8th International Conference on Enterprise Information Systems (ICEIS 2006), Paphos – Cyprus, May 23-27, pp. 46–53 (2006)
10. Loureiro, J., Belo, O.: Evaluating Maintenance Cost Computing Algorithms for Multi-Node OLAP Systems. In: Proceedings of the XI Conference on Software Engineering and Databases (JISBD 2006), Sitges, Barcelona, October 3-6, pp. 241–250 (2006)
11. Loureiro, J., Belo, O.: An Evolutionary Approach to the Selection and Allocation of Distributed Cubes. In: Proceedings of 2006 International Database Engineering & Applications Symposium (IDEAS 2006), Delhi, India, December 11-14, pp. 243–248 (2006)
12. Moscato, P.: Memetic Algorithms: A Short Introduction. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, ch.14, pp. 219–234. McGraw-Hill, London (1999)
13. Transaction Processing Performance Council (TPC): TPC Benchmark R (decision support) Standard Specification Revision 2.1.0. tpcr_2.1.0.pdf, <http://www.tpc.org>

Privacy Preserving Technique for Euclidean Distance Based Mining Algorithms Using a Wavelet Related Transform

Mohammad Ali Kadampur and Somayajulu D V L N

National Institute of Technology (NITW),
Warangal, A.P. India 506004
{kadampur,soma}@nitw.ac.in
<http://www.nitw.ac.in>

Abstract. Privacy preserving data mining is an art of knowledge discovery without revealing the sensitive data of the data set. In this paper a data transformation technique using wavelets is presented for privacy preserving data mining. Wavelets use well known energy compaction approach during data transformation and only the high energy coefficients are published to the public domain instead of the actual data proper. It is found that the transformed data preserves the Euclidean distances and the method can be used in privacy preserving clustering. Wavelets offer the inherent improved time complexity.

Keywords: Privacy, Data Mining, Wavelet Transforms.

1 Introduction

Data perturbation is one of the well known privacy preserving techniques [2]. It refers to a data transformation process typically performed by the data owners before publishing their data. Owners achieve two goals by transforming the data. First, the data gets disguised and sensitive information is not available to the public domain. Second, such transformations best preserve all those domain specific data properties that are critical for building meaningful data mining models [1,11]. These modified data sets or data models maintain task specific data utility of the published data. The tasks may vary from simple statistical analysis to the hidden knowledge discovery. The models built from data perturbation techniques are useful for applications where data owners want to participate in cooperative mining but at the same time want to prevent the leakage of privacy sensitive information in their published datasets [3,7,8,10]. In this paper we try to build such data models using a wavelet transform technique.

1.1 A Background of Wavelets

The wavelet transform of a wavelet $\psi(x)$ is mathematically defined as [6,12]:

$$W(a, b) = \int_x f(x) \frac{1}{\sqrt{a}} \psi \left(\frac{x - b}{a} \right) \quad (1)$$

which means for every (a, b) we will have a wavelet transform coefficient, representing how much the scaled wavelet is similar to the function at location $x = \frac{a}{b}$. Wavelet transform basically quantifies the local matching of the wavelet with the signal. If the wavelet matches the shape of the signal well at a specific scale and location then a large transform value is obtained otherwise if the wavelet and the signal do not correlate well, a low value of transform is obtained. Essentially application of any wavelet on a signal involves obtaining correlated coefficients as the wavelet slides along the signal [11]. The value of the coefficients depends on the wavelet chosen.

1.2 Wavelet Decomposition

The discrete wavelet transform (DWT) is an implementation of the wavelet transform using a discrete set of the wavelet scales and translations obeying some defined rules [7]. In other words, this transform decomposes the signal into mutually orthogonal set of wavelets, which is the main difference from the continuous wavelet transform (CWT), or its implementation for the discrete time series sometimes called discrete time continuous wavelet transform (DT-CWT). The key advantage of it is temporal resolution: it captures both frequency and location information (location in time).

In this paper we apply a variant of Haar [6] wavelet transformation. The Haar transform decomposes the input data into approximation coefficients and detailed coefficients. These coefficients in fact correspond to the low frequency and high frequency decompositions of the original samples respectively. The wavelet literature is rich with many interesting ways of such decomposition and reader is encouraged to refer [12].

2 Our Approach

In the proposed approach we treat entire data set as a centralized data set and design an algorithm that transforms the data set into a synthetic data set. The approach is based on the following observations

1. For most of the real data sets the energy of each transformed record is represented by very few coefficients.
2. High energy coefficient in one transformed record may have low energy coefficient in some others, but on an average energy tends to concentrate in a small set of transform coefficients. Therefore it is just sufficient if we identify and publish only such high energy coefficients.

The objective of the algorithm is to generate a set of coefficients for each record and select a set of high energy coefficients across a large number of transformed records. The method is illustrated with an example in the following section.

2.1 The Algorithm

The algorithm starts with reading the given data set D and determining its size. It adjusts the number of columns to be even so that pairing is possible while computing sum and difference coefficients. For each record algorithm finds pairwise sum and difference coefficients and stores two different intermediate matrices. The information contained in the sum coefficient matrix is usually dominant. The two coefficient matrices are concatenated after eliminating any padded zeros in the intermediate matrices.

Algorithm 1. The data perturbation algorithm

```

input : A data matrix  $D$ 
output: A perturbed matrix  $Q$ 

begin
 $[m, n] = size(D)$ ;
if ( $n \bmod 2 == 0$ ) then
     $c_n = n$ ;
    else
         $c_n = n + 1$ ;
    ss for  $i \leftarrow 1$  to  $m$  do
        for  $j \leftarrow 1$  to  $c_n$  do
            if ( $n < j < c_n$ ) then
                 $D_z(i, j) = D(i, j)$ ;
            else
                 $D_z(i, j) = 0$ ;
         $[mz, nz] = size(D_z)$ ;
        for  $i \leftarrow 1$  to  $mz$  do
            for  $j \leftarrow 1$  to  $\frac{mz}{2}$  do
                 $h_a(i, j) = \frac{D_z(i, 2j-1) + D_z(i, 2j)}{\sqrt{2}}$ ;
                 $h_d(i, j) = \frac{D_z(i, 2j-1) - D_z(i, 2j)}{\sqrt{2}}$ ;
             $D_{zhconc} = horzcat(h_a, h_d)$ ;
             $D_{zhhigh} = indexize(D_{zhconc})$ ;
             $Freq = Freqindex(D_{zhhigh})$ ;
            Select the top  $\Delta$  frequently occurring indices.
            Shuffle the order of the selected indices.
            For each row in  $D_{zhconc}$ 
             $Q =$ Select the coefficients corresponding to the shuffled order of indices.
            Publish  $Q$ .
    end

```

Where,

1. $horzcat(h_a, h_d) \Rightarrow$ concatenates the two matrices.
2. $indexize(D_{zhconc}) \Rightarrow$ Finds the indices of the coefficients
3. $Freqindex(D_{zhhigh}) \Rightarrow$ Finds the frequency of each index.

The algorithm then selects Δ coefficients with highest energy and stores their indices in a $m \times \Delta$ matrix, Where m is the number of rows. The μ indices that have occurred frequently are selected and order of them is permuted. For each row the

coefficients corresponding to these permuted indices are selected and published as perturbed matrix Q . If D is a classification data, the class variable column is left as it is while the process is applied on the other attributes and the class variable column is sent to the third party along with the selected coefficients.

2.2 Illustration

Suppose D is a matrix of size 4×5 . After padding zeros to make even number of columns, It becomes D_z .

$$D = \begin{bmatrix} 8 & 1 & 6 & 13 & 7 \\ 7 & 3 & 5 & 6 & 12 \\ 4 & 2 & 8 & 3 & 10 \\ 5 & 11 & 2 & 14 & 8 \end{bmatrix}, \quad D_z = \begin{bmatrix} 8 & 1 & 6 & 13 & 7 & 0 \\ 7 & 3 & 5 & 6 & 12 & 0 \\ 4 & 2 & 8 & 3 & 10 & 0 \\ 5 & 11 & 2 & 14 & 8 & 0 \end{bmatrix}$$

The coefficient matrix Dz_{hconc} is

$$Dz_{hconc} = \begin{bmatrix} 6.36 & 13.43 & 4.94 & 4.94 & -4.94 & 4.94 \\ 7.07 & 7.77 & 8.48 & 2.82 & -0.707 & 8.48 \\ 4.24 & 7.77 & 7.07 & 1.41 & 3.53 & 7.07 \\ 11.31 & 11.31 & 5.65 & -4.24 & -8.48 & 5.65 \end{bmatrix}$$

If we chose number of coefficients to be 4 ($\Delta = 4$), then from the first row the 4 highest coefficients in their descending order will be $\{13.43, 6.36, 4.94, 4.94\}$. From Dz_{hconc} the indices corresponding to these coefficients are $\{2, 1, 3, 4\}$. Similarly it for the second row $\{8.48, 8.48, 7.77, 7.07\}$ and the corresponding indices are $\{3, 6, 2, 1\}$. Therefore the complete index matrix corresponding to 4 highest coefficients will be,

$$Dz_{high} = \begin{bmatrix} 2 & 1 & 3 & 4 \\ 3 & 6 & 2 & 1 \\ 2 & 3 & 6 & 1 \\ 1 & 2 & 3 & 6 \end{bmatrix}$$

Frequency of the indices from the index matrix,

freq(1)=4; freq(2)=4; freq(3)=4; freq(4)=1; freq(6)=3;

If we choose $\mu = 4$ then the top 4 frequently occurring indices are $\{1, 2, 3, 6\}$. After shuffling suppose we get the order as say $\{2, 3, 6, 1\}$, then by selecting corresponding coefficients from each row of the coefficient matrix Dz_{hconc} we obtain the perturbed and transmittable matrix Q .

$$Q = \begin{bmatrix} 13.43 & 4.94 & 4.94 & 6.36 \\ 7.77 & 8.48 & 8.48 & 7.07 \\ 7.77 & 7.07 & 7.07 & 4.24 \\ 11.31 & 5.65 & 5.65 & 11.31 \end{bmatrix}$$

Our experiments show that selection of optimum number of highest energy coefficients is important in getting low average loss of distances between original

and perturbed tuples of the data sets. Figure 2 depicts change of average loss of distance for varying μ values in PNDg data set. Users can draw a curve of loss of distance against the number of coefficients μ for the data set in question and select an appropriate μ that gives minimum loss of distance.

3 Metrics and Measures

This section describes a set of metrics that reflect the utility and amount of privacy achieved in the perturbed data sets.

1. **Average Loss of Distance** : It measures the average loss of distance between the perturbed and unperturbed records. If N is the total number of records then,

$$ALD = \frac{\sum_{i=1}^i \sum_{j=1}^j (d_{i,j} - \overline{d_{i,j}})}{\text{Total Number of records compared}} \quad (2)$$

where, Total number of records compared is the value $\frac{N!}{2! \times (N-2)!}$. $d_{i,j}$ is the distance between the i^{th} and j^{th} record of original data set. $\overline{d_{i,j}}$ is the distance between the i^{th} and j^{th} record of perturbed data set.

2. **Classification Accuracy** : It is a measure of how well the classifier labels the class for the test inputs. Higher the accuracy better the classifier. It is defined as

$$Accuracy = \frac{\text{Number of samples correctly classified}}{\text{Total number of samples}} \quad (3)$$

3. **Fmeasure** : The quality of clustering is measured using this metric. It is defined as

$$F_{ij} = 2 \frac{P_{ij} R_{ij}}{P_{ij} + R_{ij}} \quad (4)$$

where, Precision

$$P_{ij} = \frac{C_i \cap \overline{C_j}}{\overline{C_j}} \quad (5)$$

and the Recall,

$$R_{ij} = \frac{C_i \cap \overline{C_j}}{C_j} \quad (6)$$

The F measure of a class C_i is $F_i = \max(F_{ij})$ and the over all Fmeasure is

$$F = \sum_{i=1}^n \frac{|C_i|}{N} F_i \quad (7)$$

4. **RP (Rank Position)** : After distortion the relative order of the value of data elements also changes. Metric RP denotes the average change of order for all attributes and is defined as

$$RP = \frac{\sum_{i=1}^m \sum_{j=1}^n |OrdA_j^i - \overline{OrdA_j^i}|}{m \times n} \quad (8)$$

Where, $OrdA_j^i$ is order of the i^{th} attribute in j^{th} row of original data and $\overline{OrdA_j^i}$ is order of the i^{th} attribute in j^{th} row of the perturbed data. Higher values of RP indicate higher perturbation and better privacy.

5. **RK (Rank maintenance)** : It represents the percentage of elements that keep their orders of value in each column after perturbation. RK is defined as below.

$$RK = \frac{\sum_{i=1}^m \sum_{j=1}^n R_{kj}^i}{m \times n} \quad (9)$$

where,

$$R_{kj}^i = \begin{cases} 1, & \text{if } OrdA_j^i = \overline{OrdA_j^i} \\ 0, & \text{otherwise.} \end{cases}$$

6. **CP (Change of rank Position of attributes)** : It defines the change of order of average value of the attributes.

$$CP = \frac{\sum_{i=1}^m |OrdAV_i - \overline{OrdAV_i}|}{m} \quad (10)$$

where, $OrdAV_i$ is the ascending order of average value of attribute i of the original data set and $\overline{OrdAV_i}$ is ascending order of average value of attribute i of the perturbed data set.

7. **CK(Change of Rank maintenance)** : It measures the percentage of attributes that keep the orders of their average value after perturbation.

$$CK = \frac{\sum_{i=1}^m C_k^i}{m} \quad (11)$$

where,

$$C_k^i = \begin{cases} 1, & \text{if } OrdA_j^i = \overline{OrdA_j^i} \\ 0, & \text{otherwise.} \end{cases}$$

In general higher, values of RP , CP and lower values of RK , CK indicate better privacy [7,8,9,10].

3.1 Experimental Evaluation

The experiments were conducted on a machine with Pentium 4, 3.4 GHz CPU, 4.0 GB of RAM, and running Windows XP Professional. Algorithms are implemented using Matlab 7.0. Initially we set up the experiments for testing kmeans

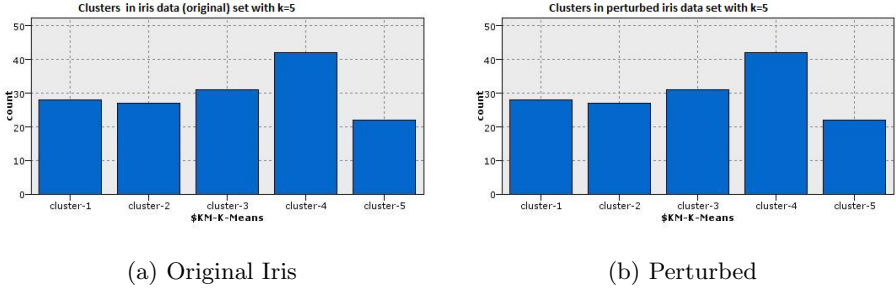


Fig. 1. Plots showing cluster statistics in the two instances of iris data set with $k=5$

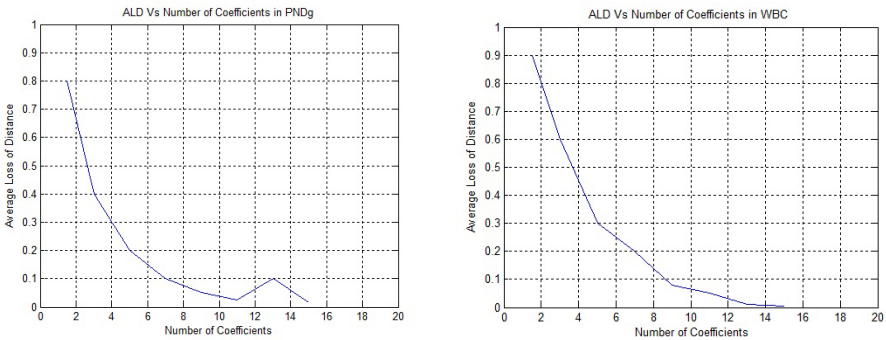
Table 1. Metric values for different data sets

Dataset	Accuracy	Fmeasure	ALD	RK	CK	RP	CP
WBC	0.70	0.72	0.10	0.0051	0.333	239.0	2.00
PNDg	0.72	0.48	0.06	0.0062	0.421	262.0	2.45
Iris	0.68	0.68	0.08	0.0042	0.235	242.0	1.88

clustering with $k = 5$ in two instances of iris data set. It is found that in the two instances of original and perturbed data sets the clustering results were identical. The following graphs in Fig 1 indicate the clustering results.

The RK, CK, RP, CP are computed by preserving the dimensions of the perturbed matrix.

The following graphs (Fig 2) show the bearing of Average loss of distance (ALD) on the selection of number of coefficients in (PNDg) data set. We have selected 10 as the optimum number of coefficients for the experiment.



(a) ALD vs Number of coefficients PNDg (b) ALD vs Number of coefficients WBC

Fig. 2. Plots showing importance of number of coefficient selection

4 Conclusion

This paper proposes a wavelet based approach using Haar like transforms to support Euclidean distance based privacy preserving data mining algorithms. The approach focuses on centralized data sets. The algorithm presented in this paper transforms the original data into a disguised equally valid synthetic data set. It is found that Euclidean based mining algorithms produce similar results on the synthetic data sets as they produce on the original data set. The number of coefficient selection has direct bearing on the analysis and optimum value need to be chosen depending upon the privacy level required.

References

1. Agrawal, R., Srikant, R.: Privacy preserving data mining. In: Proc. SIGMOD, pp. 439–450 (2000)
2. Aggarwal, C.C., Yu, P.S.: Privacy preserving Data Mining: Models and Algorithms. Springer, Heidelberg (2008)
3. Yu, H., Jiang, X., Vaidya, J.: Privacy-preserving svm using nonlinear kernels on horizontally partitioned data. In: SAC '06: Proceedings of the 2006 ACM Symposium on Applied Computing, New York, NY, USA, pp. 603–610 (2006)
4. Vaidya., J., Clifton, C.: Privacy preserving naive Bayes classifier for vertically partitioned data. In: Proc. of SIAM International Conference on Data Mining (ICDM'04), Lake Buena Vista, Florida, pp. 522–526 (April 2004)
5. Wang, J., Luo, Y., Zhao, Y., Le, J.: A survey on privacy preserving data mining. In: Proc. of First International Workshop on Database Technology and Applications, Impact on Knowledge Discovery in Databases. Journal of Database Management, vol. 14(2), pp. 14–26 (2003)
6. Soman., K.P., Ramachandran, K.I.: Insight into wavelets from theory to practice, 2nd edn. Prentice Hall, India (2005) ISBN-978-81-203-2902-7
7. L., Liu, L., Wang, J., Zhang, J.: Wavelet based data perturbation for simultaneous privacy preserving and statistics preserving. In: Proc. of IEEE International Conference on Data Mining Workshops (2008)
8. Wang, J., Zhong, W., Zhang, J.: NNMF-Based factorization techniques for high accuracy privacy protection on non-negative valued datasets. In: Proc. of Sixth IEEE International Conference on Data Mining Workshops, ICDM 2006 (2006)
9. Mukharjee, S., Banarjee, M., Chen, Z., Gangopadhyay, A.: A privacy preserving technique for distance based classification with worst case privacy guarantees. Data and Knowledge Engineering 66, 264–288 (2008)
10. Xu, S., Lai, S.: Fast Fourier transform based data perturbation method for privacy protection. In: Proc. of Intelligence and Security Informatics (2007)
11. Li, T., Li, Q., Zhu, S., Ogihara, M.: A survey on wavelet applications in data mining. SIGKDD Explorations 4(2), 49–68 (2002)
12. Li, X., Li, H., Wang, F., Ding, J.: A remark on Mallats pyramidal algorithm of wavelet analysis. Communications in Nonlinear Science & Numerical Simulation 2(4) (December 1997)
13. Young, R.: Wavelet theory and applications. Kluwer academic publishers, Boston (1993)

Extracting Features from an Electrical Signal of a Non-Intrusive Load Monitoring System

Marisa B. Figueiredo¹, Ana de Almeida¹,
Bernardete Ribeiro¹, and António Martins²

¹ CISUC - Center for Informatics and Systems, University of Coimbra,
Polo II, P-3030-290 Coimbra, Portugal

`mfig@dei.uc.pt`, `amca@mat.uc.pt`, `bribeiro@dei.uc.pt`

² ISA - Intelligent Sensing Anywhere, S.A., Rua D. Manuel I,
P- 3030-320 Coimbra, Portugal

`amartins@isa.pt`

Abstract. Improving energy efficiency by monitoring household electrical consumption is of significant importance with the present-day climate change concerns. A solution for the electrical consumption management problem is the use of a non-intrusive load monitoring system (NILM). This system captures the signals from the aggregate consumption, extracts the features from these signals and classifies the extracted features in order to identify the switched on appliances. An effective device identification (ID) requires a signature to be assigned for each appliance. Moreover, to specify an ID for each device, signal processing techniques are needed for extracting the relevant features. This paper describes a technique for the steady-states recognition in an electrical digital signal as the first stage for the implementation of an innovative NILM. Furthermore, the final goal is to develop an intelligent system for the identification of the appliances by automated learning. The proposed approach is based on the ratio value between rectangular areas defined by the signal samples. The computational experiments show the method effectiveness for the accurate steady-states identification in the electrical input signals.

Keywords: Automated learning and identification, feature extraction and classification, non-intrusive load monitoring.

1 Introduction

Several concepts that have recently arisen with the idea of Smart Environments ask for an application able to accurately identify and monitor electrical appliances consumptions, like Smart Grids or in-Home Activity Tracking. Furthermore, the monitoring systems must be inconspicuous. The use of ubiquitous computing to develop smart systems by designing a non-intrusive load monitoring system (NILM) satisfies all these requirements. Although the idea of a NILM system dates from the eighties¹, only today could it achieve its full

¹ The Electric Power Research Institute sponsored the research on NILM systems, which resulted in the American patent number 4858141, approved in 1989.

potential due to new sensing devices that render low-cost monitoring solutions. The main goal of these systems is to identify which are the appliances switched on at a certain moment in time. This allows the individual load monitoring of each identified device. After acquiring the aggregated load signal of a household network, electrical features are extracted and classified. For the correct identification of each appliance, an electrical ID is needed. The work here presented describes a technique for the recognition of steady-states in a digital signal as the first stage for the electrical devices identification needed for an innovative non-intrusive load monitoring implementation. This is a system leading to the intelligent identification of the appliances by automated learning. The approach proposed in this work is based on the ratio between rectangular areas defined by the signal samples. Computational experiments show the method effectiveness for steady-state identification in an electrical signal.

The market already offers some solutions that provide to the households the possibility of monitoring their electrical consumption: smart meters and individual meters. The first ones only measure aggregate consumptions but, until now, without the possibility of identifying which are the individual devices that are using energy. On the other hand, to monitor individual appliances it would be sufficient to use an individual meter for each one in the house. However, this would turn out to be an extremely costly solution for a household; therefore, the most viable solution for monitoring individual electrical loads would be the Non-Intrusive Load Monitoring (NILM).

The NILM concept and methodology were presented in the eighties and consist in a single device to monitor the electrical system and to identify the electric load related to each appliance without increasing the marginal cost of electricity and without the need for additional sub-measurements [1] [2]. A non-intrusive load monitoring system captures the signals from the aggregate consumption, extracts the features from these signals and classifies them in order to identify which electrical devices are switched on. To enable appliance identification is necessary to define an individual signature for each one. Thus the system for load monitoring requires the following initial steps: data acquisition (signal sampling), signal analysis and feature extraction from the acquired electrical signal. The main contribution to be presented here is the description of a new approach for the electrical signal analysis and feature extraction.

This paper is organized as follows: the next section contextualizes the NILM system concept and presents a brief related literature overview. In Section 3 a new method to analyze the step-changes in an electrical signal is proposed. Such step-changes can be used as features to identify the appliances. Finally, the experimental setup is described and initial results of signal analysis are presented in Section 4. Conclusions and future lines of work are addressed in Section 5.

2 Related Literature

Non-intrusive load monitoring uses only voltage and current signals to identify which appliances are being turned on and off. The concept was first presented by

Hart [1]. Almost simultaneously a similar idea was presented by Sultanem from Electricite de France [2]. In 1996 the first NILM system [3] was commercially produced by Enetics, Inc..

A NILM system acquires the signals, extracts events/characteristics and classifies them (see Figure 1). In order to identify the devices it is necessary to define the appliances electrical signatures which are the base of any NILM system [1]. An electrical signature for a device is defined as a set of parameters that can be measured from the total load. Hart divided signatures into two main groups: the non-intrusive and the intrusive ones. Our focus is in the non-intrusive signatures: either resulting from signal steady-states or obtained by transients.

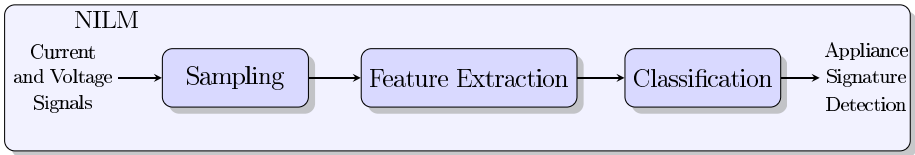


Fig. 1. A NILM high-level system

A steady-state signature is deduced from the difference between two consecutive states in a signal: one before turning on the appliance and other with the device in operation. Hart used this kind of signatures mainly because: are easier to detect (the processing requirements needed for their detection are less demanding than the ones for capture and analysis of the transients); have turning off signatures; are an additive ID (if two signatures occur at the same time, it is possible to analyze its sum). Since the seminal work in [1], the steady-state signature for residential electrical consumption management was used in [4,5], where the discrete changes on the active (and reactive) power are analyzed, or only on the active power as in [6]. However, this signature type has limitations. For instance, it is not possible to distinguish between two different appliances if they cause similar changes in the signal. Transients can be used to overcome this limitation. The transient is the noise in the electrical signal resulting from the switching on/off of an appliance. Nevertheless, this type of identification is more difficult to detect since a high sampling rate of the signals is needed.

Considering all the advantages and disadvantages between both signature types, an interesting study would be to create an appliance ID that would result from joining both types. The following section describes the first steps needed to achieve this kind of signature, namely, the signal analysis that allows for the steady-state identification.

3 Approach

Recalling the previous section, feature extraction and classification methods are applied, however, the appliances electrical signatures are the corner-stone of the

NILM system. As it was already highlighted in Hart’s work [1], steady-state signatures are simpler to detect than transients mainly because lower frequency sample rates can be used. Therefore this study starts with the signal analysis in order to identify steady-states. The data acquisition phase is possible due to the collaboration of ISA [7] company which provided a prototype device to collect the data for the initial tests.

According to Hart [1], a set of sequential samples represents a steady-state if the distance between any two samples of the set is less than a given value that will be here called tolerance. This threshold depends on the input signal that could be the active power (measured in Watts) or the reactive power (measured in VAR). Other variable is the minimum number of samples needed for a signal segment to be considered as a steady-state. The prototype proposed in [1] uses three samples of the input signal as the minimum length of a steady-state. Figure 2 presents an example where the steady-states are identified with two dashed line segments while the remaining samples belong to a change state. For the steady-states identification, methods such as filtering, differentiating and peak detection can be used [1].

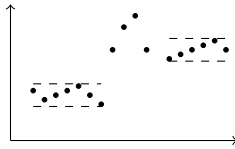


Fig. 2. Example with two steady-states and a change state

The method next proposed recognizes steady-states based on the following idea: given an initial time point x_{a-1} and an already identified steady-state, a new sampling point (x_i, y_i) is considered as part of the sequential steady-state if the rectangular area whose height equals to the sample value in that point (y_i) and whose width equals $x_i - x_{a-1}$ is similar to the sum of the sequential rectangular areas whose heights equal to the already included steady-state samples. For an illustration, assume that ϵ is the given tolerance and consider the discrete signal in Figure 3, where there is an already identified steady-state (consisting of a, b and c). Notice that, if $|k - s| < \epsilon$ with $k, s \in \{a, b, c, d\}$, then the sum of the areas A, B, C and D will be similar to the D' area, that is,

$$\left| \frac{\alpha_A + \alpha_B + \alpha_C + \alpha_D}{\alpha_{D'}} - 1 \right| \approx \gamma,$$

where $\gamma \leq 3\epsilon$ and $\alpha_A, \alpha_B, \alpha_C, \alpha_D$ and $\alpha_{D'}$ represent the areas of rectangles A, B, C, D and D' , respectively. Hence we will consider that, if

$$\left| \frac{\alpha_A + \alpha_B + \alpha_C + \alpha_D}{\alpha_{D'}} - 1 \right| \leq 3\epsilon$$

the sample point d belongs to the steady-state, otherwise it does not.

This reasoning can be generalized for a number n of $n - 1$ consecutive samples forming a steady-state and a new sample n . A sequence of null sample values is considered as a steady-state on its on.

The proposed method, hereinafter called *RA*, iteratively computes, for each new point, the ratio between the areas and verifies whether the areas are similar or not. If the new sample does not belong to the steady-state, the previous sample is considered as the end of the steady-state and the beginning of other steady-state is searched. Otherwise, the new sample is included. The algorithm ends when all the samples are tested. Accordingly to what is recommended by Hart in [1], three is the minimum number of samples needed to define a steady-state.

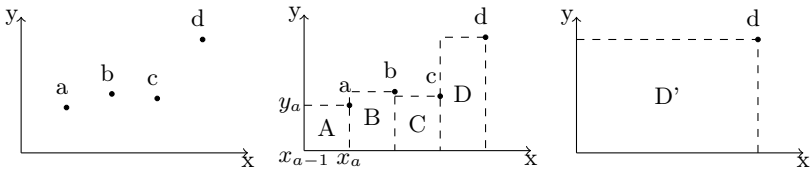


Fig. 3. Example of the rectangular areas used in the new method

Once all the steady-states have been defined, the changes in the signal are easily identified. These represent the switch on-off of an appliance. When a switch on has been identified and knowing what was the appliance, the difference between the average value of the steady-states can be used to represent the device’s signature.

4 Experiments

The sensing meter prototype provided by ISA makes possible to collect active power, voltage, current and power factor values. These values are processed to identify the steady-state signatures of the appliances. The first prototype that has been produced has some limitations like the fact that only one parameter value can be supplied at each point in time. This means that a delay between the values of the different data types will exist. This delay is, at least, 80 milliseconds, corresponding to the time that the device needs to deliver each of the several measurements. Other limitation is that errors in the measurements can eventually occur, resulting in the failure of deliverance of the expected value.

To evaluate the performance of the proposed method, data from different electrical appliances was collected using a delay of 100 milliseconds between the several data requests. Four different data types were requested. However, the following results regard to the active power signal. The frequency between each sample data type was of 400 milliseconds. The process of collecting data for each appliance can be divided in four steps: *a*) an amount of samples are collected without having the appliance plugged in the socket; *b*) the device is plugged in to the socket; *c*) the apparatus is switched on and it runs for a period of 10 to 15 seconds, and *c*) the appliance is switched off.

The described technique was implemented using Matlab as well as a version of the method proposed by Hart based on the description found in [1]. According to Section 3, we need to define two parameters: the tolerance and the minimum length for a steady-state. The latter was set to 3. The tolerance value for each appliance was found by trial and error. Based on the process used to collect the data for each appliance there are, at least, three recognizable steady-states: one before the appliance starts operating, the other obtained during the device operation and the steady-state corresponding to the period after switching off of the apparatus. The digital signal has been normalized so that the samples values belong to the limited range $[0, 1]$ and can be easily compared.

Figure 4 illustrates the active power signals (measured in watts) of the electrical appliances used in the experiment: a fan, a LCD TV, a microwave and a hairdryer. In Table 1 are presented the results for both methods: the indexes of the beginning and ending of identified steady-states and the active power mean values of each identified steady-state. In order to accurately identify the different steady-states produced by the fan’s operation, the tolerance value to be used must be, at least, 0.1. For the LCD we get a minimal value of 0.15. For the microwave we get 0.1 again, and for the hairdryer the tolerance must be set at 0.01. Note that this is a very important first processing step since these values would differ when using other sensor devices [8]. As shown in Table 1

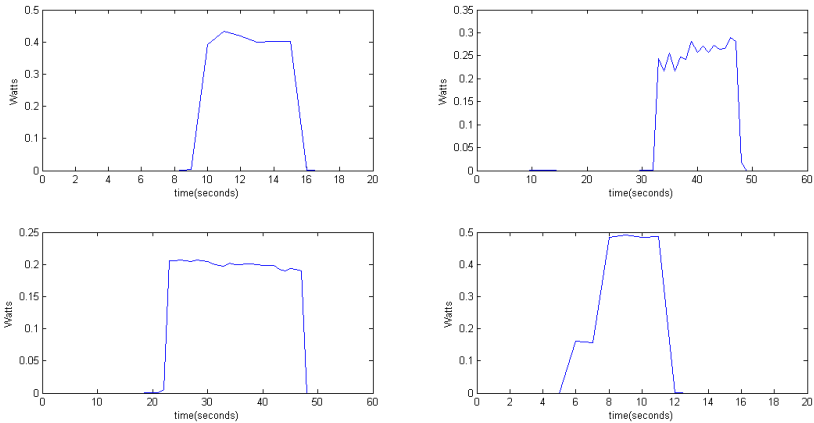


Fig. 4. Plots of the active power (Watts/s) for the fan, the LCD TV, the microwave and the hairdryer, from left to right, top to bottom

and, as expected, the two methods identified exactly the same steady-states, with the exception of the LCD data. For the latter the new technique identifies four steady-states instead of the expected three. This is due to the fact that it recognizes two different steady-states: one after connecting the LCD plug into the socket and another after the switch on of the appliance. This turns out to be a useful piece of information due to the fact that it indicates that the device has a stand-by mode that interferes with the signal characteristics. Hart’s method

Table 1. The beginning and ending of the steady-states identified by the new method, RA, and Hart’s method and the average values over each one

		Appliance												
		Fan			LCD TV			Microwave			Hairdryer			
RA method	Begin	1	10	17	1	15	33	49	1	23	48	1	8	13
	End	8	15	19	9	29	47	58	18	47	54	5	11	19
	Average value	0	0.4080	0	0	0	0.2573	0	0	0	0.1999	0	0	0.4871
Hart’s method	Begin	1	10	16	1	33	48	1	23	48	1	8	12	
	End	9	15	19	32	47	58	22	47	54	5	11	19	
	Average value	0.0003	0.4080	2.8728e-5	0.0	0.2573	0.0016	0.0002	0.1999	0	0	0.4871	8.5379e-9	

identifies correctly the beginning and the end of the steady-state corresponding to the apparatus operation mode but does not consider that there might be a significant difference in energy use at the beginning.

Another interesting fact is that, for practically all the steady-states corresponding to periods where no appliance was in operation, the beginning and ending sample indexes differ between the two methods. If we consider, for instance, the fan active power’s plot and corresponding results, Hart’s method identified sample 9 as the start of the steady-state representing the fan’s operation mode. By observing the fan’s active power chart, we can notice that the value of sample 9 is, in fact, not null and therefore considered as being a change of state by the new method. The same happens for the other devices, with the RA method always accounting for state-changes. Notice also that, for both methods, in the recognition of the hairdryer’s operation mode, the same graphic shows that samples 6 and 7, corresponding to the hairdryer medium level running, were not identified as being a steady-state on its own, which is the correct approach.

5 Conclusions and Future Work

The actual environmental problems are a serious issue that needs to be tackled urgently. The introduction of Smart Energy Grids needs automatic solutions able to identify which electrical appliances and devices are running and to characterize consumption patterns in order to promote an efficient energy management.

This paper focuses the basis for a non-intrusive load monitoring (NILM) system leading to the intelligent identification of appliances by automated learning. A simple hardware monitoring device, to acquire data from the household’s aggregate electrical signal, gathers the information. Such a framework uses feature extraction techniques and classification methods on the collected data. The system needs to be able to identify which appliances are operating and evaluate their consumptions. This can only be achieved through the definition of the electrical signature for each appliance. We propose a technique to analyze the step-changes in a signal for electrical signatures identification. The method is based on a relaxation and comparison of the areas on a step-wise electrical signal sampling. This new approach was tested for given appliances using a prototype for a sensing device. The new method was compared with the method proposed by Hart showing evidence that its results are quite promising.

This work reports an initial but meaningful study. More ambitious tests are being performed using a larger set of appliances and using several devices switched on at the same time. Also interesting is the study on how a potential method to recognize a stand-by operating mode may be incorporated into the system since it can bring an added-valued to the outputs of this framework. Further work will analyze the best method to incorporate the decision about the tolerance value for each appliance into the system itself and also the consumption variations due to of a device intermediate operation states will be taken in consideration. Finally, we intend to analyze the adequacy of the *Clustering by Compression* [9] as the classification method to be used in comparison with more traditional methods like Support Vector Machines.

Acknowledgments

The authors would like to thank ISA for the collaboration and to iTeam project for the grant support given to this project.

References

1. Hart, G.W.: Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 1870–1891 (1992)
2. Sultanem, F.: Using appliance signatures for monitoring residential loads at meter panel level. *IEEE Transactions on Power Delivery* 6, 1380–1385 (1991)
3. Drenker, S., Kader, A.: Nonintrusive monitoring of electric loads. *IEEE Computer Applications in Power* 12(4), 47–51 (1999)
4. Cole, A., Albicki, A.: Data extraction for effective non-intrusive identification of residential power loads. In: *Instrumentation and Measurement Technology Conference*, vol. 2, pp. 812–815 (1998)
5. Berges, M., Goldman, E., Matthews, H.S., Soibelman, L.: Learning systems for electric consumption of buildings. In: *ASCE International Workshop on Computing in Civil Engineering*, Austin, Texas (2009)
6. Bijker, A.J., Xia, X., Zhang, J.: Active power residential non-intrusive appliance load monitoring system. In: *IEEE AFRICON 2009* (2009)
7. ISA - Intelligent Sensing Anywhere, <http://www.isasensing.com/> (last accessed April 15, 2010)
8. Matthews, H.S., Soibelman, L., Berges, M., Goldman, E.: Automatically disaggregating the total electrical load in residential buildings: a profile of the required solution. In: *Intelligent Computing in Engineering (ICE'08) Proceedings*, Plymouth (2008)
9. Cilibrasi, R., Vitányi, P.M.B.: Clustering by compression. *IEEE Transactions on Information Theory* 51, 1523–1545 (2005)

Annotation and Retrieval of Cell Images

Maria F. O'Connor¹, Arthur Hughes^{1,*}, Chaoxin Zheng¹, Anthony Davies²,
Dermot Kelleher², and Khurshid Ahmad¹

¹ School of Computer Science & Statistics, Trinity College, Dublin, Ireland
oconnomf@tcd.ie, {Arthur.Hughes,Chaoxin.Zheng,Kahmad}@scss.tcd.ie

² Institute of Molecular Medicine, Trinity College, Dublin, Ireland
{Anthony.Davies,Dermot.Kelleher}@tcd.ie

Abstract. A multi-net neural computing system is described that can be used for classifying images based on intrinsic image features and extrinsic collateral linguistic description of the contents. A novel representation scheme based on wavelet analysis of images and a subsequent Zernike moment computation helps in a systematic extraction of image features; collateral linguistic description are obtained by the automatic extraction of single and compound keywords. We give a formal description of the system using the Z formal specification notation. An image data set comprising 480 fluorescent stained images of lymphocytes was used in the test of a 3-component unsupervised multi-net neural computing system. The classification accuracy of this system was found to be just over 85%.

Keywords: Neural networks, multi-net systems, keyword extraction, image segmentation, image annotation, cell imaging, wavelet analysis, Z-notation.

1 Foreword

Innovations in microscopy, including accessible confocal and electron microscopy, has allowed cell biologists/pathologists to observe events at sub-cellular levels. These events, ranging from a simple single cell count to the observation of cell mitosis, can now be captured as high quality images that was not hitherto possible. Cell images have substantial amounts of background noise and cannot easily be segmented; segmentation allows scientists to ‘see’ the cell at different levels of cell organisation. These images are stored by the typical end-user in a local file store and retrieved without the help of keywords or exemplar images; cell image storage and retrieval is basically a manual task. A computer-based image processing system capable of storing and retrieving images of cells can be put to three major uses. First, the system can be used as a device to control inputs to and outputs from (hospital/laboratory) data warehouse [1]; second, such a system can compensate for deficiencies related to variable illumination and/or staining [2]; and, third, the more ambitious use of such systems will be to simulate the ways in which the human cell experts examine and annotate an image.

* Corresponding author.

Such a program helps in the examination and annotation of cell images that are unknown to the program: Mango [3] has demonstrated how a neural network-based system can learn to distinguish between malignant and normal cells.

2 Literature Review

2.1 AI Based Cell Segmentation Systems

Zhou et al. [4] have developed a knowledge-base of heuristics for interpreting of images of cells (and tissues) that have been treated with organic dyes for staining various components of cells. The background appears grey, the cells appear red, green or yellow and hence segmentation from background is made possible. The authors use a variant of an edge detection algorithm and the output of the (Laplacian of Gaussian) edge detector is refined through the set heuristics mentioned above. The system was developed for a very specific set of cell images and its adaptation for other kinds of cellular images will require the construction of new knowledge bases.

2.2 Supervised Neural Network-Based Cell Segmentation Systems

PAPNET [3] is perhaps one of the most widely field-tested neural network based systems that can segment cell images. The field tests involved the use of over 200,000 images of cervical smear slides that were scanned for PAPNET in hospitals and laboratories in the USA, the UK, Italy and the Netherlands together with Australia. PAPNET was withdrawn in 2000 but large scale evaluations continued until 2006. Mangos PAPNET system can, in principle, be used in conjunction with an image capture system that then passes the image to be processed three times before generating a similarity score or classifying an image. PAPNET comprises neural networks for classifying the candidate cell images detected by colour thresholding and image morphological algorithms. Images that are identified as abnormal cells are marked by the PAPNET system as candidates of abnormal cells. These marked images are magnified into 400x and presented to the cytologistf for manual analysis.

Table 1. Confusion matrix for the performance of the PAPNET project: A total of 5170 images were used

		Expert			
		Negative	Abnormal	Unsatisfactory	
PAPNET	Negative	61.26%	4.41%	0.14%	65.80%
	Abnormal	10.50%	20.08%	0.15%	30.74%
	Unsatisfactory	0.56%	0.29%	2.61%	3.46%
		72.32%	24.78%	2.90%	100%

PAPNET evaluators found that the experts and PAPNET have a good agreement - just under 85% for negative results, 80% for abnormal images. According to the experts, 2.9% of all the 5170 images were neither negative and abnormal, and PAPNET suggests that there are 3.46% of such images.

2.3 Multi-net Unsupervised Learning and Cell Image Annotation

The variability of images obtained under in-vivo conditions, especially when the impact of drugs on cell mitosis is being studied, precludes the existence of ‘correct’ answers a priori: this justifies the use of unsupervised learning algorithms. Kohonens Self Organising Feature Maps (SOFM) are seldom used for cell segmentation; one exception is a system for segmenting organ-level images by using a two layer SOFM [5]. Multi-net systems, comprising two Kohonen Maps trained in the presence of a Hebbian network, have been used to store and retrieve images where each image in the training set had an associated vector of keywords collateral to the image content ([6], [7]); the Hebbian network affects the automatic annotation of the images. One network is trained on image features and the other on collateral keywords and the Hebbian network ensures relationship between the visual features and keywords. The multi-net approach has been used for storing and retrieving cell images, especially that of cell mitosis (informally cell division) and instances of cell adhesion [8].

3 A Multi-net Method for Annotating Cell Images

We have refined and formalised a method for automatic image annotation with specific reference to cell images through the use of a multi-net neural network. One of the networks is used for clustering images based on visual features and the other uses keywords collateral to the images [9]. There are two innovative aspects of the method: First, the two neural networks learn not only the visual features and collateral keywords of each of the images in a training, but learn to associate information in two modalities via Hebbian links. Second, we present a formal representation of the multi-net neural computing system, using the Z notation [10], to describe relationships between the constituent networks of the multi-net system. The training and testing of the networks requires a clear specification of how to extract and represent visual features, find and collate keywords from a text corpus and from experts Sections 3.1 and 3.2. The formal description of the behaviour of the algorithm on which the multi-net system is based is given in Section 3.3.

3.1 Image Representation

We have used wavelet analysis and moment invariants for representing the image features of an image. The motivation was that the use of any segmentation algorithm introduces its own idiosyncratic errors and almost all segmentation algorithms can be used only for a restricted class of images. The use of wavelet analysis is expected to help in removing the background. It has been argued that image features used for the automatic recognition of an object within an image, and by implication for image annotation, should be independent of the object’s position, size, and orientation [11]. A translation/rotation/size invariant well help in unambiguously representing an image [12]. This wavelet/invariant

moment approach will result in a multilevel representation for describing shapes of objects within an image together with having low noise sensitivity [13]. Essentially such a robust representation is built upon the projection of a digital image function, say $f(x, y)$ where x and y are coordinates, onto the monomial $x^p y^q$ (the so-called regular moments) or onto a set of orthogonal polynomials (the Zernike polynomials for example) defined over the polar co-ordinate transformation of x and y inside a unit circle - the so-called Zernike moments have the above mentioned invariance properties [14].

In order to extract image features, each image is decomposed using Daubechies D4 wavelet transform and both the wavelet and scaling functions have four coefficients each. We have used the (3,3) order Zernike function ($Z_{3,3}$). For cell images, colour is very important, as each channel in an image describes a different sub-cellular entity. The decomposition and computations for Zernike moments, therefore, are performed for each colour channel.

The computation complexity of Zernike moments depends on the number of pixels in an image of $R \times R$ dimensions and the order of the moment:

$$O(R^2 \times \frac{N \times M}{2} \times (\frac{N \times M}{2})!).$$

Where $Z_{N,M}$ is being computed over an image of dimension $R \times R$. The complexity of the function which calculates the image feature vector is then:

$$O(C \times \prod_{i=0}^{dLv-1} \frac{R^2}{2^i} \times R^2 \times \frac{N \times M}{2} \times (\frac{N \times M}{2})!).$$

Where C is the number of channels in the image, dLv is the decomposition level of the wavelet transformation, and $Z_{N,M}$ is being computed over an image of dimension $R \times R$.

The inputs to the multi-net system described above are shown in Figure 1 below. The resulting feature vector contains 72 elements; 3 Channels \times 3 Levels of Wavelet Decomposition \times 4 Signal Coefficients \times 2 Zernike moments. Two Zernike moments are computed for both the real and imaginary parts of the chosen polynomial.

3.2 Linguistic Representation

The process for computing linguistic feature vectors is as follows: from a corpus of collected domain papers, a wordlist is computed, and ordered by inverse document frequency. Collocations are then extracted from this corpus. Single keywords and collocations are then ranked by frequency and partitioned logarithmically; this method has been discussed in detailed in [6], [7], [8]. The expert input here is two fold: first, the experts suggest key journal papers and books which are used to extract the terms, and second, the experts verify and validate the keywords found automatically. The experts consulted in this study are specialists in cell mitosis.

3.3 Formalising the Annotation Algorithm

This section uses the Z formal specification language [10] to formally describe the process of annotation performed by the multi-net system. The subscripts I ,

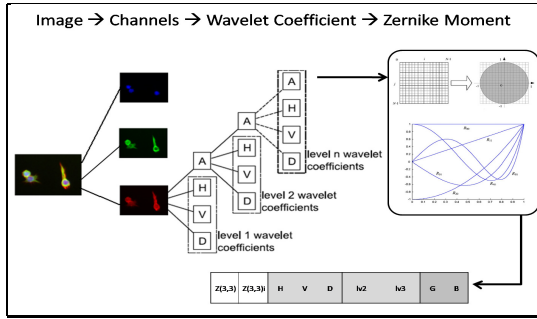


Fig. 1. Image Feature Vector Computation

T and H , in this section, will denote: the image network; the keyword network; and the Hebbian network respectively. What we ultimately wish to do is specify a relation *annotate* between image and keyword feature vectors sets:

$$| \textit{annotate} : FV_I \leftrightarrow FV_T$$

This will be achieved by factoring the *annotate* relation as a composition of three relations which will be subsequently defined. The following sets represent: the image and text feature vectors; the nodes in the image and keyword networks; the weight vectors of nodes in the image and keyword networks; and weight values in the Hebbian network:

$$[FV_I, FV_T, Node_I, Node_T, WV_I, WV_T, WV_H]$$

The weight functions which map network nodes and Hebbian nodes to their corresponding weight vector or weight value are given below. These weights are computed during training. Following this, the norms used for definition of a winning node are defined.

$$| \begin{array}{l} \textit{weight}_I : Node_I \rightarrow WV_I, \quad \textit{weight}_T : Node_T \rightarrow WV_T \\ \textit{weight}_H : Node_I \rightarrow Node_T \rightarrow WV_H \end{array}$$

$$| \begin{array}{l} \textit{norm}_I : FV_I \rightarrow Node_I \rightarrow \mathbb{R}_{\geq 0} \\ \textit{norm}_T : FV_T \rightarrow Node_T \rightarrow \mathbb{R}_{\geq 0} \\ \textit{norm}_I = \lambda x : FV_I; \lambda n : Node_I \bullet || x - \textit{weight}_I(n) || \\ \textit{norm}_T = \lambda y : FV_T; \lambda m : Node_T \bullet || y - \textit{weight}_T(m) || \end{array}$$

As the norms for each network are now defined and well-typed, the winning nodes, or most strongly linked nodes, of each network may be found using the relations below:

$$\begin{array}{|l}
 \hline
 win_I : FV_I \leftrightarrow Node_I, \quad win_T : FV_T \leftrightarrow Node_T, \quad link_H : Node_I \leftrightarrow Node_T \\
 \hline
 \forall x : FV_I; n : Node_I \bullet \\
 \quad x(win_I)n \Leftrightarrow norm_I x n = \min n' : Node_I \bullet norm_I x n' \\
 \\
 \forall y : FV_T; m : Node_T \bullet \\
 \quad y(win_T)m \Leftrightarrow norm_T y m = \min m' : Node_T \bullet norm_T y m' \\
 \\
 \forall n : Node_I; m : Node_T \bullet \\
 \quad n(link_H)m \Leftrightarrow weight_H n m = \max m' : Node_T \bullet weight_H n m' \\
 \hline
 \end{array}$$

Finally, we return to the original problem of finding a relation between image and keyword feature vectors. Using relational composition (§) and relational converse (\sim) we define *annotate* which relates two different modalities (image and text) via the winning and linking nodes in the described multi-net system:

$$\begin{array}{|l}
 \hline
 annotate : FV_I \leftrightarrow FV_T \\
 \hline
 annotate = win_I \text{ § } link_H \text{ § } (win_T) \sim \\
 \hline
 \end{array}$$

4 Case Study

4.1 Data Acquisition

We collected fluorescently-stained images which were taken in a series of experiments which investigate the effect of certain drugs on cell shape. For each fluorescent stain used, one greyscale image is taken. Up to six greyscale images may be used to produce one such “fused” image. In total, 480 images of lymphocytes were collected. The (training) images were split evenly into four classes of images; these were labelled *star*, *round*, *elongated*, and *stumpy* by our experts. Our experts chose 57 of the keywords extracted (Section 3.2) to label the training set (see Section 4.2). The rate of annotation was found to be about 8 images/hour.

4.2 System Design, Training and Evaluation

System Design. We have used CITU for computing feature vectors as well as testing and training self organising maps. This is based on our formalisation of the system (see section 3.3). CITU has been developed for automatic image annotation with specific reference to cell images and uses a multi-net neural system for the annotation exercise. The system offers novel input streams - one for archiving images and the other for either entering a textual description of the contents of an image or for providing an audio description of the contents. For image analysis, the system offers conventional image segmentation algorithms and programs for extracting colour and texture information. For a linguistic analysis, the system offers automatic extraction of single and compound keywords. The innovative aspect of CITU lies in the provision of Kohonen maps for clustering images based on visual features and similar provision for clustering of images based on keywords: a Hebbian network learns the association between simultaneous active nodes in both the network and affects annotation [9].

Training. The image network was trained using a 32×32 node map, which converged after 50 training cycles. We used 432 images for training from our 480 image database; similarly, 432 training vectors comprising keywords were used to train the collateral keyword network. Using these transforms decreases the quantification error drastically, however it also increases the error rate. The keyword feature map was trained using a 32×32 node map.

Evaluation. We tested our system with 48 of the images in our database that were not used in training. The test image (and the collateral keyword vector) was presented to the image (and keyword) networks: our system was deemed to have classified the test image (keyword) correctly if it excited a node in the network that had represented the same class of objects. Our image network classifies the four different key types of images in the testing set well. Best results were obtained for the *star*, *round*, and, *elongated* cells; the system misclassifies *stumpy* cells and only classifies 19% correctly. The confusion matrix for the (dis)agreement between our system and one domain expert shows an approx. 87% agreement (see Table 2).

The keyword vector classifies the collateral keywords for an image correctly whenever the image network classifies correctly. For instance when a *stumpy* cell was confused for being a *round* cell by the image network, the keyword network only managed to match fewer keywords related to the *stumpy* cell.

Table 2. Confusion Matrix for the CITU Zernike & Wavelet Feature Vector

		Expert				Total
		Star	Round	Elongated	Stumpy	
Multi-net	Star	22.71%	0.83%	1.67%	1.25%	26.46%
	Round	0.42%	22.08%	0.00%	1.88%	24.38%
	Elongated	1.25%	1.46%	22.92%	2.71%	28.33%
	Stumpy	0.63%	0.63%	0.42%	19.17%	20.83%
	Total	25.00%	25.00%	25.00%	25.00%	100.00%

5 Afterword

We have described how a self-organising multi-net system can be used in classifying complex images with substantial background noise. We described how the construction of image feature vector can be automated to a large extent by decomposing the images using a multi-scale analysis and using the results of the analysis to compute invariant moments for the components of the vector. The computational complexity of building an image feature vector was discussed and this suggests exercise of caution in choosing large images and/or higher moments. The accuracy of the collateral keyword network is gratifyingly high as well. We are currently finishing experiments on 1000 images and will be demonstrating the effectiveness of cross-modal retrieval.

References

1. Patel, A.A., et al.: A Novel Cross-Disciplinary Multi-Institute Approach to Translational Cancer Research: Lessons Learned from Pennsylvania Cancer Alliance Bioinformatics Consortium (PCABC). *Cancer Informatics* 3, 255–274 (2007)
2. Wooton, R., Springall, D., Polak, J.: *Image Analysis in Histology: Conventional and Confocal Microscopy*. Cambridge University Press, Cambridge (1995)
3. Mango, L.J.: Computer-assisted cervical cancer screening using neural networks. *Cancer Letters* 77(2-3), 155–162 (1994)
4. Zhou, X., Cao, X., Perlman, Z., Wong, S.T.C.: A computerized cellular imaging system for high content analysis in Monastrol suppressor screens. *Journal of Biomedical Informatics* 39, 115–125 (2006)
5. Chang, P.L., Teng, W.G.: Exploiting the Self-Organizing Map for Medical Image Segmentation. In: *Proceedings of the 20th IEEE International Symposium on Computer-based Medical Systems*, pp. 281–288 (2007)
6. Ahmad, K., Casey, M., Vrusias, B., Saragiotis, P.: Combining Multiple Modes of Information using Unsupervised Neural Classifiers. In: Windeatt, T., Roli, F. (eds.) *MCS 2003. LNCS*, vol. 2709, pp. 236–245. Springer, Heidelberg (2003)
7. Ahmad, K., Tariq, M., Vrusias, B., Handy, C.: Corpus-Based Thesaurus Construction for Image Retrieval in Specialist Domains. In: Sebastiani, F. (ed.) *ECIR 2003. LNCS*, vol. 2633, pp. 502–510. Springer, Heidelberg (2003)
8. Zheng, C., Ahmad, K., Long, A., Volkov, Y., Davies, A., Kelleher, D.: Hierarchical SOMs: segmentation of cell migration images. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) *ISNN 2007. LNCS*, vol. 4492, pp. 938–946. Springer, Heidelberg (2007)
9. Zheng, C., Ahmad, K., Long, A., Volkov, Y., Davies, A., Kelleher, D.: A Cross-Modal System for Cell Migration Image Annotation and Retrieval. In: *20th Int. Joint Conf. on Neural Networks*, Orlando, August 12-17, pp. 1738–1743. IEEE, Los Alamitos (2007)
10. Woodcock, J., Davies, J.: *Using Z Specification, Refinement, and Proof*. International Series in Computer Science. Prentice-Hall, Englewood Cliffs (1996)
11. Khotanzad, A., Hong, Y.H.: Invariant image recognition by Zernike moment. *IEEE Trans. Pattern Anal. Mach. Intell.* 12(5), 489–497 (1990)
12. Kim, W.Y., Kim, Y.S.: A region-based shape descriptor using Zernike moments. *Signal Process.-Image Commun.* 16(1-2), 95–102 (2000)
13. Kamila, N.K., Mahapatra, S., Nanda, S.: Invariance image analysis using modified Zernike moment. *Pattern Recognit. Lett.* 26(6) (2005)
14. Hwang, S.K., Kim, W.Y.: A novel approach to the fast computation of Zernike moments. *Pattern Recognition* 39(11), 2065–2076 (2006)

Adaptive Particle Swarm Optimizer for Feature Selection

M.A. Esseghir¹, Gilles Goncalves¹, and Yahya Slimani²

¹ University of Lille Nord de France, F-59000 Lille,
Artois University, LGI2A Laboratory,
Technoparc Futura, B ethune, 62400, France

² Tunis El-Manar University, Sciences Faculty of Tunis,
1060 Campus Universitaire

Abstract. The combinatorial nature of the Feature Selection problem has made the use of heuristic methods indispensable even for moderate dataset dimensions. Recently, several optimization paradigms emerged as attractive alternatives to classic heuristic based approaches.

In this paper, we propose a new an adapted Particle Swarm Optimization for the exploration of the feature selection problem search space. In spite of the combinatorial nature of the feature selection problem, the investigated approach is based on the original PSO formulation and integrates wrapper-filter methods within uniform framework. Empirical study compares and discusses the effectiveness of the devised methods on a set of featured benchmarks.

1 Introduction

Researchers in machine learning, data mining, combinatorial optimization and statistics have developed a number of methods for dimensionality reduction using usefulness and classification accuracy estimates for individual feature and subset assessment. In fact, feature selection (FS) tries to select the most relevant attributes from row data, and hence guide the construction of the final classification model or decision support system. Both feature selection and classification techniques struggle to gain the attended reliability, especially when they face high dimensional data [9]. Consequently, some trends in FS have attempted to tackle this challenge by proposing hybrid approaches or models based on multiple criteria [5]. On the other hand, recent approaches have focused on, new optimization paradigms, and specifically, approaches based on swarm intelligence (*i.e.* Ant colonies [1] and Particle swarm optimization [3]). Since, FS is combinatorial problem, a recent formulation proposed a Binary PSO variant to tackle the problem [3].

In this paper, we propose, an new alternative to the classic Particle swarm formulation of the feature selection problem. The main motivations for this proposal are three folds: (i) the use of the basic version of the PSO algorithm rather than using one of its variants; (ii) the complementary roles of wrappers and

filters; (iii) and the ability of PSO framework to integrate wrappers and filter strengths.

This paper is organized in six sections. Section 2 formalizes the feature selection problem and reviews representative approaches. Section 3 provides a brief introduction of Particle Swarm Optimization as well as its application to feature selection. Section 4 details the proposed approach. Section 5 compares and assesses the devised method. Finally, Section 6 concludes this paper and presents some directions of future research.

2 Feature Selection

Let D be a data set with F as a set of features such that $|N| = n$, and let X ($X \subseteq N$) be a subset of N . Let $J(X)$ the function that assesses the relevance of the features subset X . The problem of feature selection states the selection of a subset Z such that:

$$J(Z) = \max_{X \subseteq N} J(X) \quad (1)$$

In other words, the retained feature subset should be compact and representative of the dataset or the underlying context. This might be done by both removing redundant, noisy and irrelevant attributes by keeping the minimal information loss. For a given dataset of n features, the exhaustive exploration requires the examination of $2^n - 1$ possible subsets. Consequently, the search through the feasible solutions search space is a np -hard combinatorial problem [9]. Numerous reference approaches have been proposed for the identification of features having the highest predictive power for a given target [8]. The representative approaches could be categorized in two classes: *filters* and *wrappers* [5].

Filters. Considered as the earliest approach to feature selection, filter methods discard irrelevant features, without any reference to a data mining technique, by applying independent search which is mainly based on the assessment of intrinsic attribute properties and their relationship with the data set class (*i.e.* Relief, Symmetrical uncertainty, *etc*, Pearson correlation) [9]. The main advantage of filter methods is their reduced computational complexity which is due to the *simple* independent criterion used for feature evaluation. Nevertheless, considering one feature at a time cripple the filter to cope with either redundant or interacting features.

Wrappers. When feature selection is based on a wrapper, subsets of attributes are evaluated with a classification algorithm. The exploration of such feasible solutions requires a heuristic search strategy. The wrapper methods often provide better results than filter ones because they consider a classifier within the evaluation process. Kohavi *et al.* [8] were the first to advocate the wrapper as a general framework for feature selection in machine learning. Numerous studies have used the above framework with different combinations of evaluation and search schema. Featured search technique are ranging from greedy sequential attribute selection methods (*i.e.* SFS, SBE, Floating search) [5] to randomized and stochastic methods (*i.e.* Genetic Algorithms, GRASP, Ant colony, PSO, *etc*)

[5.11.1](#). Hybrid approaches based on filter-wrapper combination include memetic techniques, ensemble and embedded methods [5.9.11.1](#).

3 Particle Swarm Optimization (PSO)

Preliminaries

Swarm intelligence is an innovative distributed intelligent optimization paradigm that took its inspiration from biological examples by swarming, flocking and herding phenomena in vertebrates. Optimization paradigms, like Particle swarm optimization (PSO) [7](#) for continuous contexts, or Ant colonies optimization (ACO) [4](#) for discrete problems are based on a fundamental concept implemented in all swarm approaches, *the stigmergy* which is illustrated by collective behavior and implicit, or even explicit, communication of optimization agents (particles or ants) through environment. Indeed, Swarm approaches are considered as an attractive alternative to evolutionary optimization [4](#).

PSO is a population based search technique. The population is made of random solutions called *particles*. Each particle flies over the search space with specific *velocities* (one per problem dimension). When they move, respective positions change (solution). They try to improve their solutions by following leading particles and its own experience. To that end, particles should adjust their velocities according to their current and historical behavior for each one of the problem dimension (a particle velocity per dimension). The move of the *i*-th particle x_i for the dimension k (x_i^k) and velocity adjustment v_i^k are defined by Equations [3](#) and [2](#):

$$x_i^k \leftarrow x_i^k + v_i^k \quad (2)$$

$$v_i^k \leftarrow \omega * v_i^k + \psi_1 * r_1 * (p_i^k - x_i^k) + \psi_2 * r_2 * (p_g^k - x_i^k) \quad (3)$$

Here, velocity update depends on three factors: (i) the previous velocity value controlled by *the inertia factor* ω (ii) the adjustment toward the previous best position p_i^k which is scaled with random value r_1 and a weight ψ_1 (*individual factor*) (iii) and the influence of the global best solution p_g^k , that is also controlled by a weight ψ_2 (*social factor*) and a random value r_2 .

3.1 *PSO Application to Feature Selection

The PSO application to the FS problem requires the use of the Binary PSO variant (BPSO) [7](#) where particles positions are mapped into boolean values using a logistic regression function (*see eq. 4*) and a randomly generated threshold (*see eq. 5*)

$$S(v_i^k) = \frac{1}{1 + \exp(-v_i^k)} \quad (4)$$

$$if(rand() < S(v_i^k)) \quad x_i^k \leftarrow 1; else \quad x_i^k \leftarrow 0 \quad (5)$$

Equation 4 normalizes velocities into [0..1] range, whereas the Equation 5 replaces particle updating position formula (see eq. 2). Here, particle positions are coded with binary values corresponding to the state of the feature x_i in the k -th particle solution. A solution is obtained from the binary vector of particle positions by the selection of the set of features with position set to 1. The pseudo code of BPSO is detailed in Algorithm 1. The first stage (Lines 2-9) instantiates the swarm with particles and assigns to them random solutions. Next, the iterative process starts with particle position adjustment and velocities updates (Lines 13-22), followed by the evaluation of the new solutions and the update of both best particle and swarm solutions (Lines 23-27). The process restarts until a stopping condition is met or swarm converges (particle solution not improved).

An improved BPSO (IBPSO) [3] was applied to high dimensional gene expression data. The IBPSO tried to escape local minimas by resetting the values of best solution when it did not improve after a given number of iterations. By doing so, the velocity updates would, only, rely on particle experience. Then the collective behavior would generate another optimum.

4 Proposed PSO Scheme

This section details motivations, advantages, and design issues of the devised approach. Since the basic PSO was originally designed to cope with continuous problems, the adaptation to the context of feature selection did not necessarily require the systematic move to binary or discrete alternatives of basic PSO.

When selection of features are based on weights values, one can think that the more these values are representative of the feature dependence of the classification context, more the search process could be effective. In addition, filters [5], embedded methods, and weighting approaches [5,9] rely on feature scoring and ranking within the selection process.

In this paper, we devise a PSO based on feature weighting scheme. Subsets of features are derived from associated weight vectors using a threshold. Such a formulation requires, a weighted representation for each particle solution, but adhere to the original PSO schema without any need to its variants.

Initial particle weights (initial particle solution), are not set randomly, but initialized with filter scores. Therefore, the swarm particles start with weights reflecting feature-class dependency levels and, next, the PSO process adjusts its weights, by looking for relevant features combination. Since the solutions are represented weight vector, the particle position updates are done according to Equations 2 and 3 of the basic PSO scheme. In comparison to, the position update formula of the BPSO (see eq. 4 and 5), the particle moves are more dependent on velocities. Consequently, search space exploration would be enhanced by both filter specific problem knowledge and the suitable particle moves based on weights.

Such a formulation, in addition to the wrapper-filter integration within the PSO framework, allow final users to take advantage of best found subset and the associated scores (final particle weights values). Besides, the use of both wrapper and filters might enhance and guide the search to the exploration of interesting regions.

Algorithm 1. Binary P.S.O.

Input:
 nb_{part} : particles number; Cl_a : Classifier;
 ω : inertia weight;
 $v_{min}v_{max}$: velocity bounds
 ψ_1, ψ_2 : weight factors
 D : Dataset
Output: S_{best} : Best solution

```

1 begin
2   Population  $\leftarrow$   $\langle Particle \rangle$   $P \leftarrow CreateParticleSet(nb_{part})$ 
3    $S_{best} \leftarrow RandomParticlePosition()$ 
4   Evaluate( $S_{best}, Cl_a$ )
5   foreach  $p_i \in P$  do
6      $p_i.sol_{cur} \leftarrow RandomParticlePosition()$ 
7     Evaluate( $p_i.sol_{cur}, Cl_a$ )
8      $p_i.sol_{best} \leftarrow p_i.sol_{cur}$ 
9     if  $S_{best}.fitness < p_i.sol_{best}.fitness$  then
10       $S_{best} \leftarrow p_i.sol_{best}$ 
11   while Stopping criterion satisfied do
12     foreach  $p_i \in P$  do
13       //—particles move—
14        $k = 0, r_1 = 0, r_2 = 0$ 
15       while  $k < D.numAttributes$  do
16          $r_1 = rand(); r_2 = rand();$ 
17          $p_i.v^k \leftarrow$ 
18          $\omega * p_i.v^k + \psi_1 * r_1 * (p_i.sol_{best}^k - p_i.sol_{cur}^k) + \psi_2 * r_2 * (S_{best}^k - p_i.sol_{cur}^k)$ 
19         if  $(p_i.v^k \notin [v_{min}, v_{max}])$  then
20            $p_i.v^k \leftarrow max(\min(v_{max}, p_i.v^k), v_{min})$ 
21          $v \leftarrow \frac{1}{1 + \exp(-p_i.v^k)}$ 
22         if  $rand() < v$  then
23            $p_i.sol_{cur}^k \leftarrow 1$ 
24         else
25            $p_i.sol_{cur}^k \leftarrow 0$ 
26          $k \leftarrow k + 1$ 
27        $p_i.sol_{cur}.fitness \leftarrow Evaluate(p_i.sol_{cur}, Cl_a)$ 
28       //—Local and global best Updates—
29       if  $(p_i.sol_{cur}.fitness > p_i.sol_{best}.fitness)$  then
30          $p_i.sol_{best} \leftarrow p_i.sol_{cur}$ 
31       if  $p_i.sol_{best}.fitness > S_{best}.fitness$  then
32          $S_{best} \leftarrow p_i.sol_{best}$ 
33   Return ( $S_{best}$ )

```

The scores provided by filters are normalized to the range of $[-1, 1]$. Only features with positive values are considered in feature subset solutions.

The swarm particle solutions are initialized with the top- k attributes. For each particle, the value of K is randomly generated. This will allow the swarm particle to start from different search space solutions. The associated weights for the non-retained attributes are inverted, if they were positive, and inversely.

Any filter criterion could be used to generate scores for PSO. In this paper, We opt for five well known filters (scoring methods):

- Relief [10] attempts to assess features according to their discriminative power. A weight $W[i]$ is assigned to each feature. Weights are updated in a manner to reflect feature ability to distinguish between class values. This reference approach, remains one of the more representative filters used in FS.
- Information Gain and Gain Ratio filter [5] are based on information theory measures derived from information entropy and mutual information criteria.
- Symmetrical Uncertainty (SU) is another measure of the information theory. SU criterion is widely used and considered as a robust measure for attribute ranking [5][6].
- χ^2 filter [5]: is based on χ^2 statistics which computes the difference between attribute values distribution.

5 Empirical Study

We empirically assess the behavior of proposed PSO scheme as well as a the effectiveness of the filters used for the initializations stage. They will, also, be compared to the binary PSO proposed in [3].

Five benchmark datasets were used to validate the devised PSO. Sonar, Ionosphere, SpamBase, Audiology and Soybean with respectively 60, 34, 57, 69 and 35 attributes. These datasets are provided by the UCI repository [2]. Reported results, correspond to the average values of, at least, 20 trial runs. Means, Standard deviations and statistical test validation (t -test with confidence level of 97.5%) are also provided.

Two types of results are proposed: (i) those corresponding to the best solution fitness (generalization error rate) yielded from the PSO search. K-Nearest Neighbors (KNN) is used as wrapper classifier ($K = 3$) (ii) the validation on independent data set instances of the resulting features subsets using Artificial neural network (ANN) and Random forest [5]. The selection of different classification paradigms for both search and validation would make the validation less biased and independent of wrapper classifier. Besides, the validation stage is based on 10-folds cross-validation [4].

For each experiment we present, for each dataset, best solution fitness (lowest error rate %), test accuracy on independent dataset, average CPU runtime, cardinality of best solution ($\#features$) and the gain toward baseline BPSO fitness. In addition to the average, standard deviation values of the different trials, t -test was used for the assessment of the statistical validity of the obtained results toward the baseline method. Table 1 details empirical results for each data set.

Table 1. Results of the five datasets

Data	Model		Fitness%	Validation Error%		CPU Time(s)	# Attrib.	Gain% BPSO
	PSO	Filter		A.N.N.	R.F.			
Audiology	BPSO	-	35,74(3,04)	31,71(2,95)	54,45(0,56)	720(236)	51,85(5,81)	
	PSO	X2	42,64(3,36)	30,15(1,79)+	54,54(0,32)	665(219)-	40(6,17)+	-19,31%
	PSO	G.R.	33,16(5,42)+	34,41(4,71)	54,42(0,55)-	529(200)+	35,13(12,92)+	7,22%
	PSO	I.G.	33(4,38)+	33,61(3,18)	54,53(0,66)	535(206)+	35,73(13,52)+	7,67%
	PSO	Relief	33,06(5,24)+	35,07(4,28)	54,63(0,57)	505(193)+	32,95(11,33)+	7,50%
	PSO	S.U.	34,53(4,75)+	34,37(3,55)	54,6(0,44)	493(157)+	35,74(13,05)+	3,39%
Ionosphere	BPSO	-	7,26(0,79)	13,61(1,59)	15,83(1,1)	75(43)	19,62(5,64)	
	PSO	X2	9,88(0,8)	13,41(1,02)	15,71(0,97)-	111(36)	28,61(2,19)	-36,09%
	PSO	G.R.	5,34(1,89)+	13,16(1,87)+	16,15(1,42)	52(26)+	15,17(5,55)+	26,45%
	PSO	I.G.	5,72(1,71)+	13,68(1,8)	16,15(1,43)	64(45)+	16,22(5,22)+	21,21%
	PSO	Relief	4,72(1,27)+	13,47(1,2)-	16,33(1,73)	47(25)+	13,42(4,66)+	34,99%
	PSO	S.U.	4,96(1,53)+	12,48(0,95)+	16,12(1,02)	47(22)+	13,79(4,81)+	31,68%
Spam Base	BPSO	-	11,61(1,02)	9,24(0,88)	15,99(1,67)	5016(2015)	44,92(7,01)	
	PSO	X2	13,12(0,96)	8,66(0,4)+	16,45(0,25)	4661(1495)-	43,88(2,77)-	-13,01%
	PSO	G.R.	8,93(1,87)+	9,5(0,9)	15,18(2,24)+	2924(1268)+	33,32(7,65)+	23,08%
	PSO	I.G.	9,1(1,92)+	9,19(0,66)-	15,53(1,57)+	3599(1707)+	37,26(6,67)+	21,62%
	PSO	Relief	8,57(1,37)+	9,42(0,75)	15,59(1,89)-	2943(1226)+	35,39(4,64)+	26,18%
	PSO	S.U.	9,07(2,45)+	9,09(0,91)-	15,19(2,38)+	3300(1540)+	36,28(7,71)+	21,88%
Soybean	BPSO	-	10,95(0,42)	8,22(2,28)	60,02(1,97)	1836(666)	30,96(1,66)	
	PSO	X2	16,37(0)	6,94(0,27)+	58,86(0,23)+	2269(786)	35(0)	-49,50%
	PSO	G.R.	10,08(1,06)+	8,94(2,54)	60,53(2,37)-	1430(490)	25,6(4,02)+	7,95%
	PSO	I.G.	10,18(1,14)+	8,71(2,46)	60,47(2,36)	1494(572)+	26,52(4,13)+	7,03%
	PSO	Relief	9,76(1,19)+	8,77(2,8)	59,74(2,05)+	1266(485)+	23,32(2,98)+	10,87%
	PSO	S.U.	9,88(1,03)+	9,89(2,48)	60,55(2,12)	1413(450)+	25,47(2,89)+	9,77%
Sonar	BPSO	-	15,46(1,81)	25,47(1,99)	41,78(1,97)	113(41)	39,44(7,8)	
	PSO	X2	10,75(3,44)+	26,41(4,12)	39,74(2,59)+	82(46)+	30,73(7,8)+	30,47%
	PSO	G.R.	10,41(4,7)+	27,19(2,86)	41,62(2,44)+	77(42)+	30,34(9,93)+	32,66%
	PSO	I.G.	10,13(4,3)+	27,32(2,57)	41,21(2,49)+	79(40)+	30,81(8,88)+	34,48%
	PSO	Relief	9,72(2,98)+	27,35(3,02)	40,54(3,11)+	71(32)+	30,72(6,17)+	37,13%
	PSO	S.U.	9,33(3,96)+	27,07(3,47)	41,86(2,08)	62(31)+	25,6(6,79)+	39,65%

Result format: $m(sd)^{+/-}$: m : Mean; sd : Standard deviation; $(+/-)$: T-test validity

Globally, The assessed alternatives of the proposed PSO outperform the binary PSO according to the fitness criterion. The reported gain (toward BPSO) is ranging from 3% to 39%. In some cases, the gain value is negative and the BPSO accuracy is superior to our PSO. Another interesting result, is the enhancement of the classification accuracy on validation data with classifiers that were not used by PSO for fitness evaluation.

When we compare devised PSO instances with respective filters, variants based on Relief and SU tends to return the best accuracy. Nevertheless, X2 filter is not well adapted to the proposed PSO scheme, because it is usually outperformed by the baseline method (BPSO).

We can conclude that the superiority of the devised approach is not only related to the use of the weighted scheme but also, the effective filter-wrapper integration contribute to the enhance both classification accuracy but also reduces the number of the attributes without considerable increase in computational cost.

6 Conclusion

We devise a new FS approach based on continuous PSO. The proposed approach is able to integrate both filter and wrapper techniques within a swarm framework. Empirical study showed improvement of the new approaches over the over classic FS problem formulation with binary PSO. The investigated approach could be extended to a distributed and operative method. Another extension alternative is the filters fusion through the evolution of heterogeneous particles based on different scoring techniques. Exploration processes based on PSO could be, also, improved with adaptive local search through the exploration of particle neighborhoods.

References

1. Al-Ani, A.: Ant colony optimization for feature subset selection. In: WEC, vol. (2), pp. 35–38 (2005)
2. Blake, C., Merz, C.: UCI repository of machine learning databases (1998), <http://www.ics.uci.edu/mllearn/MLRepository.html>
3. Chuang, L.-Y., Chang, H.-W., Tu, C.-J., Yang, C.-H.: Improved binary pso for feature selection using gene expression data. *Computational Biology and Chemistry* 32(1), 29–38 (2008)
4. Engelbrecht, A.P.: *Computational Intelligence: An Introduction*, Inter edn. John Wiley & Sons, Chichester (2007)
5. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: *Feature Extraction, Foundations and Applications*. Series Studies in Fuzziness and Soft Computing. Springer, Heidelberg (2006)
6. Hall, M.A., Holmes, G.: Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering* 15(3) (June 2003)
7. Kennedy, J., Eberhart, R.C.: *Swarm intelligence*. Morgan Kaufmann Publishers Inc., San Francisco (2001)
8. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial Intelligence* 97, 273–324 (1997)
9. Liu, H., Motoda, H.: *Computational methods of feature selection*. Chapman and Hall/CRC Editions (2008)
10. Robnik-Sikonja, M., Kononenko, I.: An adaptation of relief for attribute estimation in regression. In: *ICML*, pp. 296–304 (1997)
11. Yusta, S.C.: Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters* 30(5), 525–534 (2009)

A Randomized Sphere Cover Classifier

Reda Younsi and Anthony Bagnall

School of Computing Sciences
University of East Anglia
Norwich NR4 7TJ
UK
{ry, ajb}@cmp.uea.ac.uk

Abstract. This paper describes an instance based classifier, the randomised sphere covering classifier (αRSC), that reduces the training data set size without loss of accuracy when compared to nearest neighbour classifiers. The motivation for developing this algorithm is the desire to have a non-deterministic, fast, instance based classifier that performs well in isolation but is also ideal for use with ensembles. Essentially we trade off decreased testing time for increased training time whilst retaining the simple and intuitive nature of instance based classifiers. We use twenty four benchmark datasets from UCI repository for evaluation. The first set of experiments demonstrate the basic benefits of sphere covering. We show that there is no significant difference in accuracy between the basic αRSC algorithm and a nearest neighbour classifier, even though αRSC compresses the data by up to 75%. We then describe a pruning algorithm that removes spheres that contain α or fewer training instances. The second set of experiments demonstrate that when we set the α parameter through cross validation, the resulting αRSC algorithm outperforms several well known classifiers when compared using the Friedman rank sum test. Thirdly, we highlight the benefits of pruning with a bias/variance decomposition. Finally, we discuss why the randomisation inherent in αRSC makes them an ideal ensemble component and outline our future direction.

1 Introduction

Instance based learning techniques [1] operate by keeping a typical sample of the training data then classifying new instances based on their similarity to the retained sample. Instance based learning algorithms are defined by three characteristics: a similarity function that specifies the closeness of two instances, a selection function that selects the samples to be kept by the algorithm, and a classification function that decides on the class of unseen instances. The simplest and most popular IBL algorithm is the nearest neighbour (NN) algorithm which retains the entire training set. Although surprisingly effective, one well documented problem with NN classifiers is that classifying a new instance requires a distance calculation for each instance in the training set. Data reduction algorithms have been studied in great depth [13, 7, 6]. In general, these algorithms

search the training data for a subset of cases and/or attributes with which to classify new instances with the objective of achieving the maximum compression with the minimum reduction in accuracy. Typically, these algorithms require multiple passes over the data set using some greedy heuristic to find a locally optimal subset. In this paper we propose a much simpler and faster randomised algorithm that creates spheres around a subset of instances, then bases classification on distance to spheres rather than instances.

2 Background

The sphere covering mechanism we use stems from the class covering approach to classification [2]. The Class Cover Problem (CCP) involves finding the smallest number of sets covering (i.e. containing) points from one class without covering any points from a second class. The solution to the CCP proposed in [12] involves constructing a Class Cover Catch Digraph (CCCD), a directed graph based on the proximity of training cases. However, finding the optimal covering via the CCCD is NP-hard [3]. Hence [9, 14] proposed a number of greedy algorithms to find an approximately optimal set covering. Whilst covering techniques have shown to be good classifiers that effectively compress the data [12], these algorithms have the draw back that they are still time consuming and that they only find pure coverings, i.e. sets that only contain cases of a single class. An algorithm that relaxes the requirement of class purity was proposed by [12]. This algorithm introduces two parameters to alleviate this constraint on a pure proper cover. Whilst effective, the parameters are non intuitive and hard to set. The greedy algorithms proposed all have a run time complexity of $O(n^2)$, and hence (to the best of our knowledge) there has been very limited experimental evaluation of the algorithms proposed in [9, 12] since most of them are impractical for large and complex datasets. The sphere covering algorithm we propose follow the same principles proposed in [10]. The algorithm is computationally efficient, randomizes the process of finding a set covering and allows for pruning through the setting of a single parameter α which directly penalise complex covers.

3 A Randomized Sphere Cover Classifier (αRSC)

The reason for designing the αRSC algorithm was to develop an instance based classifier to use in ensembles. Hence our design criteria were that it should be randomised (to allow for diversity), fast (to mitigate against the inevitable overhead of ensembles) and comprehensible (to help produce meaningful interpretations from the models produced). The αRSC algorithm has a single integer parameter, α , that specifies the minimum size for any sphere. Informally, for any given α , αRSC works as follows.

1. Repeat until all data are covered
 - (a) Randomly select a data point and add it to the set of covered cases.
 - (b) Create a new sphere centered at this point.

- (c) Find the closest case in the training set of a different class to the one selected as a centre.
- (d) Set the radius of the sphere to be the distance to this case.
- (e) Find all cases in the training set within the radius of this sphere.
- (f) If the number of cases in the sphere is greater than α , add all cases in the sphere to the set of covered cases and save the sphere details (centre, class and radius).

A more formal algorithmic description is given in Algorithm 1. For all our experiments we use the Euclidean distance metric, although the algorithm can work with any distance function.

Algorithm 1. A Randomized Sphere Cover Classifier (αRSC)

```

1: Input: Cases  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ , distance function  $d(\mathbf{x}_i, \mathbf{x}_j)$  parameter  $\alpha$ .
2: Output: Set of spheres  $B$ 
3: Let covered cases set  $C = \emptyset$ 
4: while  $D \neq C$  do
5:   Select a random element  $(\mathbf{x}_i, y_i) \in D - C$ 
6:   Copy  $(\mathbf{x}_i, y_i)$  to  $C$ 
7:   Find  $\min_{(x_j, y_j) \in D} d(\mathbf{x}_i, \mathbf{x}_j)$  such that  $y_i \neq y_j$ 
8:   Let  $r_i = d(\mathbf{x}_i, \mathbf{x}_j)$ 
9:   Create a  $B_i$  with a center  $\mathbf{c}_i = \mathbf{x}_i$ , radius  $r_i$ 
   and target class  $y_i$ 
10:  Find all the cases in  $B_i$  and store in temporary set  $T$ 
11:  if  $|T| \geq \alpha$  then
12:     $C = C \cup T$ 
13:    Store the sphere  $B_i$  in  $B$ 
14:  end if
15: end while

```

The parameter α allows us to smooth the decision boundary, which has been shown to provide better generalization by mitigating against noise and outliers, (see for instance (8)). Figure 1 provides an example of the smoothing effect of removing small spheres on the decision boundary.

The αRSC algorithm classifies a new case by the following rules:

1. **Rule 1.** A test example that is covered by a sphere, takes the target class of the sphere. If there is more than one sphere of different target class covering the test example, the classifier takes the target class of the sphere with the closest center.
2. **Rule 2.** In the case where a test example is not covered by a sphere, the classifier selects the closest spherical edge.

A case covered by Rule 2 will generally be an outlier or at the boundary of the class distribution. Therefore, it may be preferable not to have spheres over-covering areas where such cases may occur. These areas are either close to the decision boundary specifically when the high overlap between classes exist

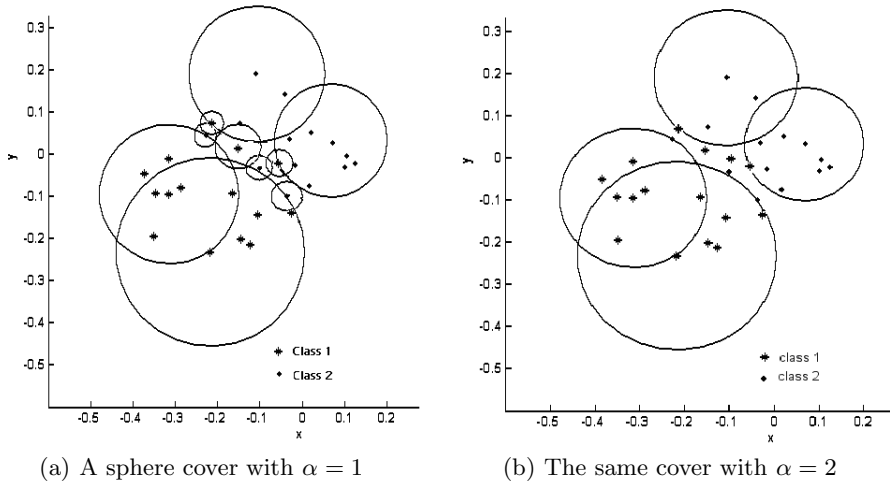


Fig. 1. An example of the smoothing effect of removing small spheres

(an illustration is given in Figure 1 (a)), and areas where noisy cases are within dense areas of examples of different target class. The αRSC method of compressing through sphere covering and smoothing via boundary setting provides a robust simple classifier that is competitive with other commonly used classifiers.

4 Results

In this section, we perform three sets of experiments. The first set is meant to demonstrate the general principle that we can intelligently compress the data set using αRSC without significantly reducing classification error. The second set of experiments shows that the αRSC classifier performs as well as or better than other classifiers based on similar principles. The third set of experiments investigates the bias/variance trade-off of using pruning (α parameter), and the role of bias and variance in the reduction of the generalization error.

4.1 Experimental Setup

To evaluate the performance of αRSC , we used twenty four datasets from both UCI data repository and boosting repository¹. These datasets are summarized in table 1. They were selected because they vary in the numbers of training examples, classes and attributes and thus provide a diverse testbed. In addition, they all have only continuous attributes.

For the experiments in Section 4.2 we used a stratified ten-fold cross validation (CV). For the later experiments we performed model selection on the parameter values, in that for each fold of the overall cross-validation we first

¹ <http://ida.first.gmd.de/raetsch/data/benchmarks.htm>

Table 1. Benchmark datasets used for the empirical evaluations

Dataset	examples	features	classes	Dataset	examples	features	classes
Sonar	208	60	2	Vehicle	846	18	4
Glass6	214	9	6	Vowel	990	10	11
Glass2	214	9	2	German	1000	20	2
Thyroid	215	5	2	Concentric	2000	2	2
Heart	270	13	2	Image segment	2310	18	2
Haberman	306	3	2	Abalone	4177	8	3
Cancer	315	13	2	Clouds	5000	2	2
Ecoli	336	7	8	Waveforme	5000	40	3
Ionosphere	351	34	2	Ringnorm	7400	20	2
wdbc	569	30	2	Twonorm	7400	20	2
Winsconsin	699	9	2	Pendigitis	10991	14	10
Pima Diabetes	768	8	2	Magic	19020	2	10

took a test sample, then cross validated on the remainder to set the parameter. For comparison purposes we compare to K-NN using the full data, the Non Nested Generalized Exemplar (NNge) [11], C4.5, Naive Bayes and NBTree. K-NN and NNge are the most relevant for comparison purposes, the other three are included for completeness. Weka implementations are used for the standard classifiers, bespoke implementations for αRSC and NNge.

4.2 Experiment 1: Compression without Loss of Accuracy

Our first experiment demonstrates that even by using the most basic form of αRSC we can massively compress the data without significantly reducing the accuracy compared to an instance based learner using the full data set. We ran a tenfold cross validation on each of our 24 data sets Table 2 shows the accuracy on the test data. A paired Mann-Whitney test cannot reject the null hypothesis that the average of the difference in accuracy is zero. Table 2 also shows the average compression rate achieved for each data set.

Table 2. 10 fold Cross-Validation classification accuracy (in%) and standard deviation over the folds. The final column gives the average compression rate (Comp) for unpruned RSC.

Data Set	1-NN	unpruned RSC	Comp	Data Set	1-NN	unpruned RSC	Comp
vehicle	69.61(4.62)	68.13(4.75)	50%	glass6	70.3 (8.96)	69.0(9.49)	52%
segment	97.14(1.07)	96.1 (1.21)	89%	cancer	67.65(7.8)	68.08(7.76)	52%
abalone	50.13(2.25)	49.46(2.02)	32%	breastw	95.67(2.48)	95.36(2.42)	90%
waveform	85.88(1.57)	85.41(1.55)	73%	concentric	98.54(0.79)	98.21(0.82)	97%
ringnorm	72.59(0.53)	95.16(0.49)	63%	clouds	84.64(1.68)	84.75(1.48)	76%
magic	80.16(0.32)	79.95(0.35)	68%	wdbc	94.01(2.95)	95.38(2.65)	88%
pendigits	98.95(0.1)	97.72(0.25)	93%	thyroid	96.8 (3.33)	95.4(4.44)	88%
vowel	98.9 (1.05)	95.7(2.34)	77%	german	70.7 (4.34)	70.3(3.86)	52%
twonorm	94.51(0.29)	93.78(0.34)	83%	diabetes	70.62(4.67)	68.87(5.02)	51%
glass2	94.25(4.72)	93.86(5.67)	87%	ionosphere	87.1 (5.12)	92.8(3.75)	69%
ecoli	80.66(6.16)	81.75(6.26)	66%	heart	75.78(7.34)	75.26(8.98)	60%
haberman	65.77(6.92)	68.58(7.38)	53%	sonar	86.23(7.41)	82.8 (8.48)	61%

The average compression rate was 75%. These experiments clearly show that by using αRSC we can discard a large proportion of the data whilst maintaining the same level of accuracy.

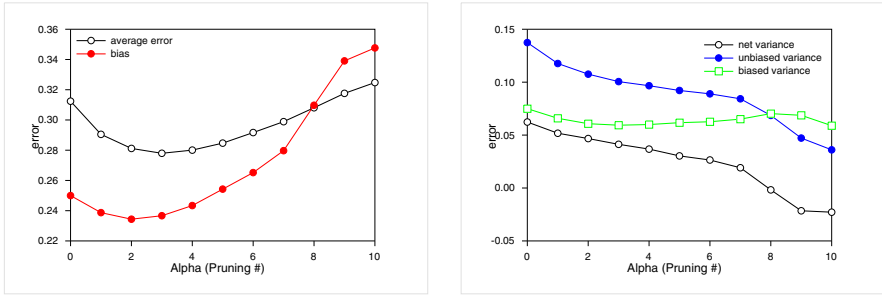
Table 3. Test set classification accuracy (in %) of αRSC , K-Nearest neighbour (KNN), Decision tree (J48), Naive Bayes tree (NBT), NaiveBayes (NB) and Non-nested Generalised Hyper-rectangle (NNge) using average results of 10 different runs for αRSC .

Data Set	αRSC	KNN	NB	J48	NBT	NNge
abalone	52.76	54.12	51.65	52.08	52.5	53.48
Clouds	89.05	89	87.18	88.88	88.88	88.29
Concentric	97.98	98.24	95.65	96.82	95.65	96.71
Diabetes	76.18	75.57	78.24	78.24	79.77	72.14
German	73.35	75.59	70	65.88	70.59	68.24
Magic	83.68	83.27	77.47	84.75	85.63	80.75
Pend	98.17	99.14	84.64	95.59	94.38	95.4
Ringnorm	95.2	73.89	98.81	91.3	98.05	84.7
Segment	95.88	97.2	76.46	95.29	94.4	93.38
Twonorm	96.42	97.46	97.58	85.06	95.71	87.24
Vehicle	65.9	62.5	54.51	66.32	68.75	60.76
Vowel	90.59	93.77	67.66	74.18	75.96	83.09
Waveform	89.65	89.24	84.06	87.12	88	84.88
Wdbc	93.04	93.3	91.75	93.81	95.88	95.88
Wins	95.92	95.8	97.06	95.38	96.22	95.38
Sonar	79.72	87.32	77.46	67.61	73.24	70.42
Thyroid	98.11	98.65	97.3	97.3	97.3	91.89
Glass2	92.6	91.78	87.67	89.04	93.15	91.78
Glass6	69.59	72.6	65.75	64.38	64.38	75.34
Haberman	78.86	78.1	79.05	76.19	67.62	68.57
Heart	77.56	82.61	82.61	71.74	77.17	80.43
Iono	95.5	80.83	91.67	85	87.5	89.17
Ecoli	83.97	81.9	83.62	77.59	81.9	74.14
Cancer	70.14	73.68	70.53	73.68	71.58	64.21
Mean Rank	4.46	4.42	3.65	2.98	2.85	2.65
Friedman Rank	1	2	3	4	5	6

4.3 Experiment 2: Performance Equivalent to Other Classifiers

The accuracy results in table 3 are based on an independent test set drawn randomly from the data set. We use 66% of the dataset for training and tested the classifiers on the same remaining test set. However, given the randomisation nature of αRSC , we choose to use the average of 10 runs in order to make a fair comparison. Tuning the parameters for both α and K is based on 10 CV on the training set alone. NNge was trained based on the best parameters suggested by its authors. The decision tree is trained without pruning (J48) using the default parameters in WEKA. Naive Bayes has no parameters.

We are primarily interested in the relative performance of the classifiers over the range of data sets. In order to compare the algorithms on the overall datasets, we use Friedman ranks sum test [5]. This test ranks the classifiers over each dataset (with the best performing algorithm getting the Rank of 1, the second best rank 2, etc.). Let r_{ij} be the rank of the j^{th} of k algorithms on the i^{th} of N datasets. The average rank of classifier, $R_j = \frac{1}{N} \sum_i r_{ij}$ gives a non-parametric summary of the relative performance over all the data sets, and it has been shown that the ranking themselves provide a fair comparison of the algorithms [5]. αRSC has the highest average rank of the five classifiers tested. KNN has the highest number of top ranks but on some datasets it performed relatively badly, reducing its overall rank. These results suggest that αRSC can perform well on a variety of datasets in comparison to other classifiers, and that the smoothing



(a) Average error and bias curves of αRSC on Diabetes dataset (b) Variance curves of αRSC on Diabetes dataset

Fig. 2. Average error, Bias and Variance graphs of αRSC classifier in relation to α

process reduces the tendency of instance based learners to perform very badly on some data sets. In order to understand the generalisation error of αRSC , we employed the bias/variance decomposition based on Domingo’s framework [4].

4.4 Experiment 3: Bias/Variance Decomposition of αRSC

We conducted Domingos’ bias/variance decomposition of the generalisation error on several datasets. The bias is attributed to the systematic part of the error, while variance to the stochastic part of the error [4]. The trade-off between bias and variance is noticed using a variety of algorithms on several experiments found in the literature.

1. Bias arises when the classifier cannot represent the true function. That is, the classifier underfits the data.
2. Variance arises when the classifier overfits the data.
3. There is often a trade-off between bias and variance.

Figure 2 shows the bias and variance curves in relation to α for the Diabetes dataset. The important observation that can be made from the bias/variance results is that pruning reduces significantly unbiased variance. However, in only few cases that a small reduction in bias is shown. Therefore, the decrease of αRSC average error is caused mainly by the decrease of net variance.

5 Conclusions

The classification accuracy of our proposed randomised classifier is competitive in comparison to various deterministic algorithms. In addition, the classification accuracy is significantly improved by pruning, to the extent that on average it outranks five other classifiers. The reason for this improvement is the result of unbiased variance reduction as demonstrated by the bias/variance experiments. Pruning is only responsible in the reduction of unbiased variance which indicate that further improvement is possible by reducing bias. Feature selection is

known to reduce bias for IBL classifiers. We intend on investigating the effect of feature selection on αRSC and assess the usefulness of the classifier when used in ensembles.

References

- [1] Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. *Machine Learning* 6(1), 37–66 (1991)
- [2] Cannon, A., Cowen, L.: Approximation algorithms for the class cover problem. In: *AAAI* (2000)
- [3] Cannon, A., Mark Ettinger, J., Hush, D., Scovel, C.: Machine learning with data dependent hypothesis classes. *The Journal of Machine Learning Research* 2, 335–358 (2002)
- [4] Domingos, P.: A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. In: *AAAI/IAAI*, pp. 564–569 (2000)
- [5] Janez, D.: Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* 7, 1–30 (2006)
- [6] Kim, S.W., Oommen, B.J.: A brief taxonomy and ranking of creative prototype reduction schemes. *Pattern Anal. Appl.* 6(3), 232–244 (2003)
- [7] Kuncheva, L.I., Bezdek, J.C.: Presupervised and postsupervised prototype classifier design. *IEEE Transactions on Neural Networks* 10(5), 1142–1152 (1999)
- [8] Liu, H., Motoda, H.: On issues of instance selection. *Data Mining and Knowledge Discovery* 6(2), 115–130 (2002)
- [9] Marchette, D.J., Priebe, C.E.: Characterizing the scale dimension of a high-dimensional classification problem. *Pattern Recognition* 36(1), 45–60 (2003)
- [10] Marchette, D.J., Wegman, E.J., Priebe, C.E.: A Fast Algorithm for Approximating the Dominating Set of a Class Cover Catch Digraph. Technical Report, JHU DMS TR #635 (2003)
- [11] Martin, B.: Instance-Based learning: Nearest Neighbor With Generalization, University of Waikato, Hamilton, New Zealand, MSC thesis (1995)
- [12] Priebe, C., DeVinney, J.G., Marchette, D., Socolinsky, D.A.: Classification Using Class Cover Catch Digraphs. *Journal of Classification* 20(1), 3–23 (2003)
- [13] Wilson, D., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. *Machine Learning* 38, 257–286 (2000)
- [14] Marchette, D.J.: *Random Graphs for Statistical Pattern Recognition*. Wiley Interscience, Hoboken (2004)

Directed Figure Codes with Weak Equality

Włodzimierz Moczurad

Institute of Computer Science, Jagiellonian University
Łojasiewicza 6, 30-348 Kraków, Poland
wkm@ii.uj.edu.pl

Abstract. We consider directed figures defined as labelled polyominoes with designated start and end points, equipped with catenation operation that uses a merging function to resolve possible conflicts. This is one of possible extensions generalizing words and variable-length codes to planar structures, creating a new tool for *e.g.* image analysis and retrieval.

Within this model, we define four kinds of codes that differ in their treatment of start and end points. This is a generalization that weakens (one or both of) equality tests in the classical definition of a code. We show a strict hierarchy of those kinds and prove that testing whether a given set of figures is a code of a given kind is decidable. This is an extension of previous results, leading to a verification algorithm for codes of all four kinds.

1 Introduction

Extensions of classical words and variable-length word codes have been studied by many authors. For instance, Aigrain and Beauquier introduced polyomino codes in [1]; two-dimensional rectangular pictures were studied by Giammarresi and Restivo in [4], whilst in [8] Mantaci and Restivo described an algorithm to verify tree codes. The interest in picture-like structures is not surprising, given the huge amounts of pictorial data that we process. Unfortunately, properties related to codicity are often lost in the two-dimensional, geometric world. In particular, codicity testing is undecidable for polyominoes and similar structures, *cf.* [2,9].

In [7] we introduced directed figures defined as labelled polyominoes with designated start and end points. This setting is similar to symbolic pixel pictures, described by Costagliola *et al.* in [3], and admits a natural definition of catenation. The attribute “directed” is used to emphasize the way figures are catenated; this should not be confused with the meaning of “directed” in *e.g.* directed polyominoes. We proved that verification whether a given finite set of directed figures is a code is decidable. This is a significant change in comparison to previously mentioned picture models, facilitating the use of directed figures for *e.g.* encoding and indexing of pictures in databases.

In the present paper we extend the previous results, allowing a more general definition of a code. This gives rise to four kinds of codes, some of which may

be better-suited to particular applications like picture indexing or “pictorial barcoding.” We prove that those kinds form a non-trivial hierarchy with respect to inclusion. The main result is the decidability of codicity testing for all four kinds of codes. Algorithms that find a double factorization of a figure for a non-code can be derived from the proof, although for the sake of brevity we do not include them in this paper.

Our generalization weakens (one or both of) figure equality tests in the classical definition of a code:

$$x_1 \cdots x_k = y_1 \cdots y_l \Rightarrow k = l \text{ and } x_i = y_i \text{ for } i \in \{1, \dots, k\}.$$

This is an attitude that resembles the (R, S) -codes defined by Halava *et al.* in [5]. The weak equality disregards the start and end points of figures, thus allowing simpler representations that might be useful in the above-mentioned applications.

We begin, in Section 2, with definitions of directed figures, codes and related notions. Then, in Section 3, we show the relationship between the different kinds of codes. Section 4 contains the main result of the paper, proof of the decidability of codicity verification. Finally, we conclude with remarks on algorithms that may be derived from the main theorem and possible extensions of the result.

2 Preliminaries

Let Σ be a finite, non-empty alphabet. A *translation* by vector $u = (u_x, u_y) \in \mathbb{Z}^2$ is denoted by $\tau_u, \tau_u : \mathbb{Z}^2 \ni (x, y) \mapsto (x + u_x, y + u_y) \in \mathbb{Z}^2$. By extension, for a set $V \subseteq \mathbb{Z}^2$ and an arbitrary function $f : V \rightarrow \Sigma$ define $\tau_u : P(\mathbb{Z}^2) \ni V \mapsto \{\tau_u(v) \mid v \in V\} \in P(\mathbb{Z}^2)$ and $\tau_u : \Sigma^V \ni f \mapsto f \circ \tau_{-u} \in \Sigma^{\tau_u(V)}$. For points $u = (u_x, u_y), v = (v_x, v_y) \in \mathbb{Z}^2$ and a subset $U \subseteq \mathbb{Z}^2, U \neq \emptyset$ define $\text{dist}(u, v) = |u_x - v_x| + |u_y - v_y|$, $\text{dist}(U, v) = \min_{u \in U} \{\text{dist}(u, v)\}$ and $\Gamma(U) = \{v \in \mathbb{Z}^2 \mid \text{dist}(U, v) \leq 1\}$. A set $U \subseteq \mathbb{Z}^2$ is *connected* if for any $x, y \in U$ there exists a sequence $x = x_1, x_2, \dots, x_n, x_{n+1} = y$ contained in U such that $\text{dist}(x_i, x_{i+1}) = 1$ for $i \in \{1, \dots, n\}$.

Definition 1 (Directed figure). *Let $D \subseteq \mathbb{Z}^2$ be finite and connected, $b \in D, e \in \Gamma(D)$ and $l : D \rightarrow \Sigma$. A quadruple $f = (D, b, e, l)$ is a directed figure (over Σ) with*

domain	$\text{dom}(f) = D,$
start point	$\text{begin}(f) = b,$
end point	$\text{end}(f) = e,$
labelling function	$\text{label}(f) = l.$

Translation vector of f is defined as $\text{tran}(f) = \text{end}(f) - \text{begin}(f)$. Additionally, the empty directed figure ε is defined as $(\emptyset, (0, 0), (0, 0), \{\})$, where $\{\}$ denotes a function with an empty domain.

The set of all directed figures over Σ is denoted by Σ° . Two directed figures x, y are *equal* (denoted by $x = y$) if there exists $u \in \mathbb{Z}^2$ such that

$$y = (\tau_u(\text{dom}(x)), \tau_u(\text{begin}(x)), \tau_u(\text{end}(x)), \tau_u(\text{label}(x))).$$

Two directed figures x, y are *weakly equal* (up to their start/end points; denoted by $x \simeq y$) if there exists $u \in \mathbb{Z}^2$ such that

$$\text{dom}(y) = \tau_u(\text{dom}(x)) \text{ and } \text{label}(y) = \tau_u(\text{label}(x)).$$

Thus, we actually consider figures up to translation. Note that for all $x, y \in \Sigma^\diamond$, $x = y$ implies $x \simeq y$.

Example 1. A directed figure and its graphical representation. Each point of the domain, (x, y) , is represented by a unit square in \mathbb{R}^2 with bottom left corner in (x, y) . A circle marks the start point and a diamond marks the end point of the figure. Figures are considered up to translation, hence we do not mark the coordinates.

$$(\{(0, 0), (1, 0), (1, 1)\}, (0, 0), (2, 1), \{(0, 0) \mapsto a, (1, 0) \mapsto b, (1, 1) \mapsto c\})$$

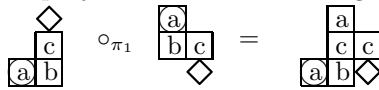


Definition 2 (Catenation). Let $x = (D_x, b_x, e_x, l_x)$ and $y = (D_y, b_y, e_y, l_y)$ be directed figures. Catenation of x and y with respect to a merging function $m : \Sigma \times \Sigma \rightarrow \Sigma$ is defined as $x \circ_m y = (D_x \cup \tau_{x_e - y_b}(D_y), b_x, \tau_{x_e - y_b}(e_y), l)$, where

$$l(z) = \begin{cases} l_x(z) & \text{for } z \in D_x \setminus \tau_{x_e - y_b}(D_y), \\ \tau_{x_e - y_b}(l_y)(z) & \text{for } z \in \tau_{x_e - y_b}(D_y) \setminus D_x, \\ m(l_x(z), \tau_{x_e - y_b}(l_y)(z)) & \text{for } z \in D_x \cap \tau_{x_e - y_b}(D_y). \end{cases}$$

When m is fixed, we simply write xy instead of $x \circ_m y$.

Example 2. Let π_1 be the projection onto the first argument.



Proposition 1. $\Sigma_m^\diamond = (\Sigma^\diamond, \circ_m)$ is a monoid if and only if m is associative.

Proof. By definition of \circ_m .

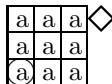
From now on let m be an arbitrary associative merging function. We will use the notation Σ_m^\diamond for both the monoid and the set of directed figures itself. Abusing this notation, we will also write X_m^\diamond to denote the set of all figures that can be composed by \circ_m catenation from figures in $X \subseteq \Sigma_m^\diamond$. We omit the m subscript when m is irrelevant or clear from the context, e.g. when $|\Sigma| = 1$.

Proposition 2. Monoid Σ_m^\diamond is never free.

Proof. The basis of Σ_m^\diamond contains the figures

$$E = \boxed{a} \diamond, N = \diamond \boxed{a}, W = \diamond \boxed{a}, S = \boxed{a} \diamond,$$

where $a \in \Sigma$. This contradicts the freeness, since the figure below can be decomposed as $NNESSENNE$ and $EENWWNEEE$, regardless of the merging function.



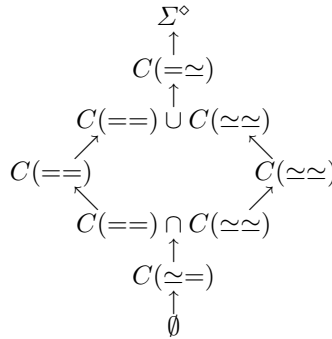
3 Codes

In this section we define four kinds of directed figure codes, resulting from the possible use of weak equality in the classical definition of a code, and we show their hierarchy with respect to inclusion.

Definition 3 (Code). Let $\stackrel{R}{=}$ and $\stackrel{S}{=}$ be one of the equalities on figures, $=$ or \simeq . $X \subseteq \Sigma_m^\diamond$ is a $(\stackrel{R}{=} \stackrel{S}{=})$ -code if for any $x_1, \dots, x_k, y_1, \dots, y_l \in X$ the equality $x_1 \cdots x_k \stackrel{R}{=} y_1 \cdots y_l$ implies $k = l$ and $x_i \stackrel{S}{=} y_i$ for each $i \in \{1, \dots, k\}$.

The set of all $(\stackrel{R}{=} \stackrel{S}{=})$ -codes is denoted by $C(\stackrel{R}{=} \stackrel{S}{=})$. For the sake of clarity the notation does not reference Σ_m^\diamond . We thus define four families of codes, $C(==)$, $C(=\simeq)$, $C(\simeq=)$, and $C(\simeq\simeq)$, with $C(==)$ being the usual directed figure codes.

Proposition 3. The diagram illustrates inclusions between different families of codes. All the inclusions are strict.



Proof. Since $x = y$ implies $x \simeq y$, it is clear that $C(\simeq=)$ is contained in both $C(==)$ and $C(\simeq\simeq)$, and these are contained in $C(=\simeq)$. Examples that follow show that all inclusions are strict and that $C(==)$ and $C(\simeq\simeq)$ are incomparable with respect to inclusion. Moreover, $C(\simeq=)$ and $C(=\simeq)$ are not trivial.

Example 3. Consider

$$X = \left\{ \begin{array}{|c|c|} \hline a \\ \hline a \ a \\ \hline \end{array} \diamond, \begin{array}{|c|c|} \hline b \\ \hline a \ a \\ \hline \end{array} \diamond, \begin{array}{|c|c|} \hline c \\ \hline a \ a \\ \hline \end{array} \diamond \right\} \subseteq \{a, b, c\}^\diamond.$$

X is clearly a $(\simeq=)$ -code and thus a code of all four kinds. Hence $C(\simeq=)$ is not empty.

Example 4. Take

$$X = \left\{ w = \begin{array}{|c|c|} \hline \\ \hline a \ a \\ \hline \end{array} \diamond, x = \begin{array}{|c|c|} \hline a \ a \\ \hline \\ \hline \end{array} \diamond, y = \begin{array}{|c|} \hline a \\ \hline a \\ \hline \end{array} \diamond, z = \begin{array}{|c|} \hline a \\ \hline a \\ \hline \end{array} \diamond \right\} \subseteq \{a\}^\diamond.$$

X is not a $(=\simeq)$ -code since

$$wx = yz = \begin{array}{|c|c|} \hline a \ a \\ \hline a \ a \\ \hline \end{array} \diamond.$$

Thus, it is not a code of any other kind, and consequently $C(=\simeq) \neq \Sigma^\diamond$.

Example 5. Consider

$$X = \{w = \begin{array}{|c|} \hline a \\ \hline b \\ \hline \end{array} \diamond, x = \begin{array}{|c|} \hline c \\ \hline d \\ \hline \end{array} \diamond, y = \begin{array}{|c|} \hline a \\ \hline b \\ \hline \end{array} \diamond, z = \begin{array}{|c|} \hline c \\ \hline d \\ \hline \end{array} \diamond\} \subseteq \{a, b, c, d\}^\diamond.$$

X is clearly a $(\simeq\simeq)$ -code, but not a $(=)$ -code, since $wx = yz$.

Example 6. Now let

$$X = \{x = \begin{array}{|c|c|} \hline a & b \\ \hline a & b \\ \hline \end{array} \diamond, y = \begin{array}{|c|} \hline a \\ \hline a \\ \hline \end{array} \diamond, z = \begin{array}{|c|c|} \hline b & a \\ \hline b & a \\ \hline \end{array} \diamond\} \subseteq \{a, b\}^\diamond.$$

X is a $(=)$ -code, but not a $(\simeq\simeq)$ -code, since $xy \simeq yz$. Thus, $C(=)$ and $C(\simeq\simeq)$ are incomparable with respect to inclusion.

Example 7. Take X of Example 6 with two figures added:

$$X' = X \cup \{x' = \begin{array}{|c|c|} \hline a & b \\ \hline a & b \\ \hline \end{array} \diamond, z' = \begin{array}{|c|c|} \hline b & a \\ \hline b & a \\ \hline \end{array} \diamond\}.$$

X' is a (\simeq) -code, but neither a $(\simeq\simeq)$ -code, nor a $(=)$ -code, since $x'z = xz'$.

Example 8. Finally, let

$$X = \{x = \begin{array}{|c|} \hline a \\ \hline a \\ \hline \end{array} \diamond, y = \begin{array}{|c|} \hline a \\ \hline a \\ \hline \end{array} \diamond\} \subseteq \{a\}^\diamond.$$

X is both a $(=)$ -code and a $(\simeq\simeq)$ -code, but not a (\simeq) -code, since $x \simeq y$.

The last two examples additionally show that $C(=\simeq)$ is strictly larger than the union $C(\simeq\simeq) \cup C(=)$ and that $C(\simeq=)$ is strictly smaller than this union.

Before proceeding with the main decidability result, note that there is an “easy case” that can be verified quickly just by analyzing the translation vectors of figures. This is reflected in the following theorem.

Theorem 1 (Necessary condition). *Let $X = \{x_1, \dots, x_n\} \subseteq \Sigma_m^\diamond$. If there exist $\alpha_1, \dots, \alpha_n \in \mathbb{N}$, not all equal to zero, such that $\sum_{i=1}^n \alpha_i \text{tran}(x_i) = (0, 0)$, then X is not a $(=\simeq)$ -code.*

Proof. Let $x = \underbrace{x_1 \cdots x_1}_{\alpha_1} \underbrace{x_2 \cdots x_2}_{\alpha_2} \cdots \underbrace{x_n \cdots x_n}_{\alpha_n}$. Now consider the powers of x , x^i

for $i \geq 1$. Since $\text{tran}(x) = (0, 0)$, each of the powers has the same domain. There is only a finite number of possible labellings of this domain, which implies that regardless of the merging function and labelling of x , there exist $a, b \in \mathbb{N}$, $a \neq b$ such that $x^a = x^b$. Hence X is not a $(=\simeq)$ -code.

The condition of Theorem 1 can be interpreted as follows: the translation vectors do not fit in a half-plane. Let \cdot denote the usual dot product.

Corollary 1. *If X is a code (of any kind), then there exists $v \in \mathbb{Z}^2$ such that $v \cdot \text{tran}(x) > 0$ for all $x \in X$. In other words, there exists a line passing through $(0, 0)$ such that all translation vectors are on one side of it.*

4 Decidability of Verification

We now prove that testing whether a given set of figures is a code of a given kind is decidable. We begin with a construction of a bounding area for figures and then show several properties that guarantee finiteness of configuration sets, implying the decidability of codicity verification for all four kinds of codes. This is a generalisation of the proof given in [7].

Let $\stackrel{R}{=}$ and $\stackrel{S}{=}$ be one of the equalities on figures, $=$ or \simeq . We are going to verify whether a given set $X = \{x_1, \dots, x_n\} \subseteq \Sigma_m^\circ$ is a $(\stackrel{R}{=} \stackrel{S}{=})$ -code.

Observe that either there exists a vector $\tau_E \in \mathbb{Z}^2$ such that for all $x \in X$, $\tau_E \cdot \text{tran}(x) > 0$ or, by Theorem [1], X is not a code. If τ_E exists, it can be chosen long enough so that for all $x \in X$, $\text{dom}(x) \cup \{\text{end}(x)\} \subseteq \text{HP}(\tau_E, \text{begin}(x))$, where, for given $v, w \in \mathbb{R}^2$, $\text{HP}(v, w)$ denotes a half-plane given by $\{u \in \mathbb{Z}^2 \mid v \cdot (u - (w + v)) \leq 0\}$.

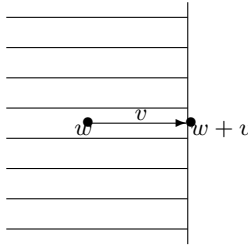


Fig. 1. A half-plane $\text{HP}(v, w)$

Consequently, $\text{dom}(x) \cup \{\text{end}(x)\} \subseteq \text{HP}(\tau_E, \text{end}(x))$ for all $x \in X$. Without loss of generality we can assume

$$\angle(R_{-\frac{\pi}{2}}(\tau_E), \text{tran}(x_1)) \leq \angle(R_{-\frac{\pi}{2}}(\tau_E), \text{tran}(x_2)) \leq \dots \leq \angle(R_{-\frac{\pi}{2}}(\tau_E), \text{tran}(x_n))$$

(where R_ϕ denotes a rotation by ϕ and \angle is the angle spanned by two vectors) and $\text{begin}(x) = (0, 0)$ for all $x \in X$.

Now choose constants $r_S, r_N, r_W > 0$ such that the vectors $\tau_N = r_N R_{\frac{\pi}{2}}(\text{tran}(x_n))$, $\tau_W = -r_W \tau_E$ and $\tau_S = r_S R_{-\frac{\pi}{2}}(\text{tran}(x_1))$ define a bounding area for figures in X , i.e., for all $x \in X$

$$\text{dom}(x) \cup \{\text{end}(x)\} \subseteq \bigcap_{\tau \in \{\tau_N, \tau_W, \tau_S\}} \{ \text{HP}(\tau, \text{begin}(x)) \}.$$

Figure [2] shows half-planes $\text{HP}(\tau, \text{begin}(x))$ for $\tau \in \{\tau_E, \tau_N, \tau_W, \tau_S\}$.

For $x \in X_m^\circ$ define

$$\begin{aligned} CE^+(x) &= \text{HP}(\tau_S, \text{end}(x)) \cap \text{HP}(\tau_N, \text{end}(x)) \cap \text{HP}(\tau_W, \text{end}(x)), \\ CE^-(x) &= \mathbb{Z}^2 \setminus CE^+(\text{end}(x)), \\ CW^+(x) &= \bigcup_v \{ v + (CE^+(\text{end}(x)) \cap \text{HP}(\tau_E, \text{end}(x))) \}, \\ CW^-(x) &= \mathbb{Z}^2 \setminus CW^+(\text{end}(x)), \end{aligned}$$

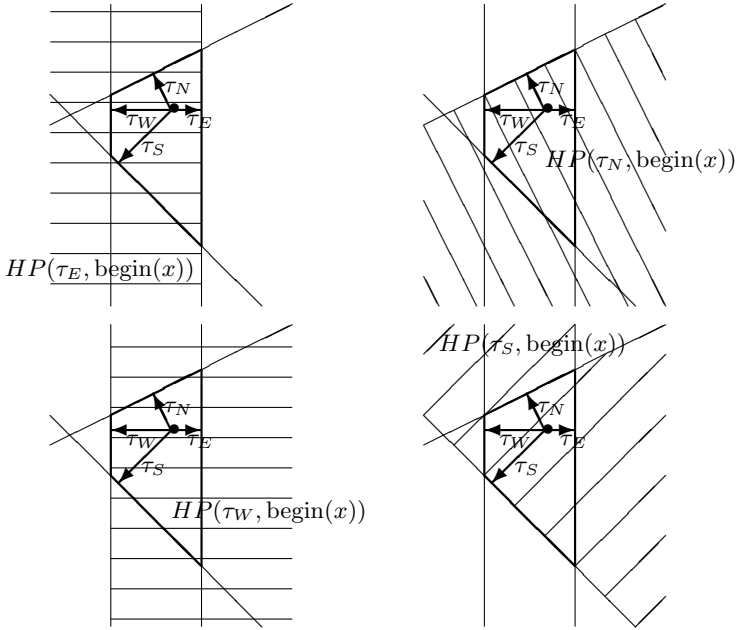


Fig. 2. Half-planes $HP(\tau, \text{begin}(x))$ for $\tau \in \{\tau_E, \tau_N, \tau_W, \tau_S\}$; the black dot denotes the start point of x

where the union in the definition of $CW^+(x)$ is taken over $v \in \mathbb{Z}^2$ lying within an angle spanned by vectors $-\text{tran}(x_1)$ and $-\text{tran}(x_n)$.

The following properties are now immediately proven:

Proposition 4. For all $x, y \in X_m^\diamond$ labels of $CE^-(x)$ cannot be overwritten in xy , i.e., $u \in CE^-(x) \cap \text{dom}(x) \Rightarrow \text{label}(x)(u) = \text{label}(xy)(u)$ and $u \in CE^-(x) \setminus \text{dom}(x) \Rightarrow u \notin \text{dom}(xy)$.

Proposition 5. For all $x \in X_m^\diamond$ labels of $CW^-(x)$ are not defined in x , i.e., $u \in CW^-(x) \Rightarrow u \notin \text{dom}(x)$.

Proposition 6. For all $x, y \in X_m^\diamond$, $CE^+(xy) \subseteq CE^+(x)$ and $CW^+(x) \subseteq CW^+(xy)$.

We define a *configuration* as a pair (x, y) , with $x, y \in X_m^\diamond$. We say that $(x', y') \in (X_m^\diamond)^2$ is a *successor* of (x, y) and write $(x, y) \prec (x', y')$ if

$$\begin{aligned} x' &= xx'' \text{ for some } x'' \in X \text{ and } y = y', \text{ or} \\ y' &= yy'' \text{ for some } y'' \in X \text{ and } x = x'. \end{aligned}$$

By \prec^* we denote the transitive closure of \prec . Obviously X is not a $(\frac{R}{S})$ -code if and only if there exist $x, y \in X$ and $z, z' \in X_m^\diamond$ such that $x \stackrel{S}{\neq} y, z \stackrel{R}{=} z'$ and $(x, y) \prec^* (z, z')$.

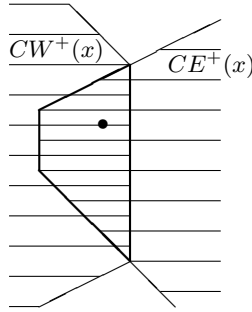


Fig. 3. $CW^+(x)$ and $CE^+(x)$ regions; the black dot denotes the end point of x

Our goal is either to find a configuration (x, y) such that $x \not\stackrel{S}{\prec} y$ and $(x, y) \prec \dots \prec (z, z')$ for some $z \stackrel{R}{=} z'$ (then X is not a $(\stackrel{R}{=} \stackrel{S}{=})$ -code), or to prove that such a configuration does not exist (then X is a $(\stackrel{R}{=} \stackrel{S}{=})$ -code). A configuration satisfying the above condition will be called an *eventually terminating configuration*.

Unfortunately, there are potentially infinitely many configurations to check. The following propositions will let us reduce the number of configurations under consideration.

Proposition 7. *If (x, y) is eventually terminating and $(x', y') \prec (x, y)$, then (x', y') is eventually terminating.*

Proof. Obvious.

Proposition 8. *If (x, y) is eventually terminating, then*

$$\begin{aligned} \text{end}(x) &\in CW^+(y) \cup CE^+(y), \\ \text{end}(y) &\in CW^+(x) \cup CE^+(x). \end{aligned}$$

Proof. See definitions of CW^+ and CE^+ .

Proposition 9. *If (x, y) is eventually terminating, then*

$$\text{label}(x) \mid_{CE^-(x) \cap CE^-(y)} \equiv \text{label}(y) \mid_{CE^-(x) \cap CE^-(y)}.$$

Proof. See definition of CE^- and Proposition 4

Notice that we do not need all of the information contained in configurations, just those labellings that can be changed by future catenations. By Proposition 9, instead of (x, y) we can consider a *reduced configuration* defined as a pair $(\pi_{RC}(x, y), \pi_{RC}(y, x))$ where

$$\pi_{RC}(z, z') = (\text{end}(z), \text{label}(z) \mid_{\text{dom}(z) \setminus (CE^-(z) \cap CE^-(z'))}).$$

Now Proposition 7 implies that we need only consider configurations where the span along τ_E is bounded by $|\tau_E|$, i.e., $|\tau_E \cdot (\text{end}(x) - \text{end}(y))| \leq |\tau_E|^2$, since no single figure advances $\text{end}(x)$ or $\text{end}(y)$ by more than $|\tau_E|$. Moreover, Proposition 8 restricts the perpendicular span (in the direction of $R_{-\frac{\pi}{2}}(\tau_E)$). Hence the number of reduced configurations, up to translation, is finite.

This leads us to the main theorem of the paper:

Theorem 2. *It is decidable whether a given finite set $X \subseteq \Sigma_m^\diamond$ is a code of a given kind.*

5 Final Remarks

The proof of Theorem 2 leads to an algorithm that verifies whether a given set of directed figures is a code. The algorithm, in a basic ($=$)-code version, can be found in 7. Its running time depends on the angle spanned by the translation vectors of figures. Bigger angles result in higher running time, since there are more configurations to check. However, when the angle reaches π , the running time drops radically because of Theorem 1. When the angle tends to zero, reduced configurations resemble simple word factors and the algorithm becomes similar to the well-known algorithm of Sardinas and Patterson.

One of possible extensions of our result is to consider a slightly different figure model, where catenation is a partial operation without a merging function, defined for those figures that do not overlap when catenated, cf. 6.

References

1. Aigrain, P., Beauquier, D.: Polyomino tilings, cellular automata and codicity. *Theoretical Computer Science* 147(1-2), 165–180 (1995)
2. Beauquier, D., Nivat, M.: A codicity undecidable problem in the plane. *Theoretical Computer Science* 303(2-3), 417–430 (2003)
3. Costagliola, G., Ferrucci, F., Gravino, C.: Adding symbolic information to picture models: definitions and properties. *Theoretical Computer Science* 337, 51–104 (2005)
4. Giammarresi, D., Restivo, A.: Two-dimensional finite state recognizability. *Fundamenta Informaticae* 25(3), 399–422 (1996)
5. Halava, V., Harju, T., Kärki, T.: Relational codes of words. *Theoretical Computer Science* 389(1-2), 237–249 (2007)
6. Kolarz, M.: Directed figures: extended models. *Theoretical Informatics and Applications RAIRO* (submitted), <http://www.ii.uj.edu.pl/~kolarz/ita09010.pdf>
7. Kolarz, M., Moczurad, W.: Directed figure codes are decidable. *Discrete Mathematics and Theoretical Computer Science* 11(2), 1–14 (2009)
8. Mantaci, S., Restivo, A.: Codes and equations on trees. *Theoretical Computer Science* 255, 483–509 (2001)
9. Moczurad, W.: Brick codes: families, properties, relations. *International Journal of Computer Mathematics* 74, 133–150 (2000)

Surrogate Model for Continuous and Discrete Genetic Optimization Based on RBF Networks

Lukáš Bajer^{1,*} and Martin Holeňa²

¹ Faculty of Mathematics and Physics, Charles University
Malostranské nám. 25, Prague 1, Czech Republic
<http://bajeluk.matfyz.cz/>

² Institute of Computer Science, Czech Academy of Sciences
Pod Vodárenskou věží 2, Prague 8, Czech Republic
<http://www2.cs.cas.cz/~martin/>

Abstract. Surrogate modelling has become a successful method improving the optimization of costly objective functions. It brings less accurate, but much faster means of evaluating candidate solutions. This paper describes a model based on radial basis function networks which takes into account both continuous and discrete variables. It shows the applicability of our surrogate model to the optimization of empirical objective functions for which mixing of discrete and continuous dimensions is typical. Results of testing with a genetic algorithm confirm considerably faster convergence in terms of the number of the original empirical fitness evaluations.

Keywords: surrogate modelling, RBF networks, genetic algorithms, continuous and discrete variables.

1 Introduction

Many of the current optimization tasks in industry or engineering are characterized by high dimensionality and combination of continuous and discrete variables [1, 2]. Such tasks can be found, for example, in situations where the problem is described by empirical objective function – the evaluation of solutions is attained by some experiment or measurement.

Substituting an approximating model for an empirical objective function is a popular approach to tasks with costly objective functions. This approach, called surrogate modelling, is widely used in connection with evolutionary algorithms (EAs), in spite of having been originally introduced in the area of smooth optimization.

Assessing some of the individuals with not necessary accurate, but much faster model brings an important benefit: a notably larger population can be evolved in parallel. Even though the precise evaluation can be made only on a limited number of individuals, the EA can explore a larger part of the input space.

* This work was supported by the Czech Science Foundation (GAČR), grant number 201/09/H057.

This paper describes a particular surrogate model based on radial basis function (RBF) networks. Differently to existing works dealing with similar kinds of surrogate models [3–5], our model considers simultaneously both continuous and discrete input dimensions. Multiple RBF networks are trained and discrete variables are used for focusing training of the networks on the most appropriate data. Further, detailed genetic algorithm which utilizes this model is provided as well as two kinds of tests.

The paper is organized as follows: in the next section, we recall principles of evolutionary algorithms, RBF networks and surrogate modelling. Section 3 describes our approach to constructing a surrogate model and using it in optimization. Finally, Section 4 provides the results of testing on a benchmark function and real-world data.

2 Involved Methods

Surrogate Modelling. Approximation of the fitness function with some regression model is a common cure for tasks when empirical objective function has to be used. These *surrogate models* simulate behaviour of the original function while being much cheaper and much less time consuming to evaluate. As a surrogate model, any suitable regression model can be used [6–8].

In connection with evolutionary optimization, artificial neural networks of the type multilayer perceptrons [9] and networks with radial basis functions [3, 4] have been particularly popular. The last mentioned kind of neural networks underlies also the model reported in this paper.

RBF Networks [10] compute a mapping from the input space (typically a subspace of \mathbb{R}^n) to \mathbb{R} (for simplicity we will focus on versions with scalar output). The mapping can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^g \pi_i f_i(\|\mathbf{x} - \mathbf{c}_i\|) \quad (1)$$

where \mathbf{x} is the input, g the number of components, f_i are radial basis functions, π_i their weights, \mathbf{c}_i radial functions' centres, and $\|\cdot\|$ is a norm.

Evolution control (EC) determines how the original fitness function and the surrogate model are combined during the optimization. In the literature [9], individual and generation based approaches are distinguished.

At the beginning of each generation, *individual-based* EC evaluates all the individuals with the approximating model at first. Then, some of them are chosen for re-evaluation with the original fitness function.

The second type is *generation-based* EC. The basic idea of this approach is rather simple: generations are grouped into cycles of a fixed length λ . In each cycle, η of the generations are controlled by the original fitness function and the rest by the approximate model.

3 Our Strategy for Using Surrogate-Assisted Genetic Optimization

We introduce our version of the surrogate-assisted genetic algorithm in this section. A standard genetic algorithm is integrated with a surrogate model for the objective function. Basic steps of the algorithm are outlined in Fig. 1.

The algorithm starts with preparation of the initial population of admissible solutions. In each generation, the algorithm fits the model, evaluates individuals either with the original fitness or with the surrogate model, and generates a new population with selection, a crossover and mutation. These steps are repeated until some acceptable solution is found, or user-specified resources are exhausted.

The surrogate model can be used only when the initial set of original-fitness-evaluated input points of size s_{\min} has been collected. Whenever an individual is evaluated by the original objective function, the evaluation is stored into the database and can be later used for model fitting.

3.1 Model Construction

Since RBF networks, as defined in Section 2, are able to work only with continuous values, the algorithm has to separate discrete variables from RBF networks fitting. It uses these discrete values for clustering available original fitness function data into separate clusters $\{C_j\}_{j=1}^m$ (step (1) in Fig. 2), and trains an independent RBF network for each cluster using only continuous variables. As a distance for clustering, Hamming or Jaccard distance is used.

The sizes of the clusters $\{|C_j|\}_{j=1}^m$ has to be at least s_{\min} – the minimal number of data needed for fitting one component of the RBF network. This number is provided by the user and its best value depends on a particular task. The higher the s_{\min} is, the more components can each RBF network have, but the more distinct discrete values are usually grouped together in one cluster. In any case, the s_{\min} has to fulfil a condition $s_{\min} \geq \lceil \frac{k}{k-1} \rho \rceil$ where ρ is the number of trained RBF network parameters and k the number of folds for cross-validation. We have enhanced a standard bottom-up hierarchical clustering [11] to obtain clusters of specified minimal size.

One separate RBF network rbf_j is trained from the data of each cluster C_j . The maximal number of components of each network is upper-bounded by $g_j^{\max} = \lfloor (\frac{k-1}{k} |C_j|) / \rho \rfloor$. The algorithm uses k -fold cross-validation to estimate mean squared error (MSE) of the j -th RBF network for all the possible numbers of components $g_j = 1, \dots, g_j^{\max}$. The number of components g_j^* leading to the lowest error measured by MSE from cross-validation (MSE_{CV}), Akaike's information criterion (AIC) or Bayes' information criterion (BIC) is selected (step (4) in Fig. 2). Finally, the network is retrained once more with g_j^* components using all the available data from the corresponding cluster.

Training of the RBF networks leads to least squares minimization for which suitable methods of continuous optimization exist. The starting values of function centres are taken randomly from uniform distribution; the k -th dimension of

the i -th centre \mathbf{c}_i is for j -th RBF network sampled as $(\mathbf{c}_{ji}^0)_k \sim U(\min_{l \in C_j}(\mathbf{x}_l^{(C)})_k, \max_{l \in C_j}(\mathbf{x}_l^{(C)})_k)$ where $\mathbf{x}_l^{(C)}$ corresponds to the continuous part of the l -th data from the cluster C_j . This random sampling gave better results than additional clustering and setting \mathbf{c}_i^0 as clusters' centres. The initial weights π_{ji}^0 were set to an average value of the previously measured objective function y from the data belonging to each cluster C_j .

Surrogate GA

Input:

- specification of the optimization task (list of discrete variables $v_1^{(D)}, \dots, v_d^{(D)}$ and continuous variables $v_1^{(C)}, \dots, v_n^{(C)}$, constraints on values of variables, p – size of the population to be evaluated by the *original fitness*; optionally: P_0 – initial population)
- evolution control parameters (*individual*, resp. *generation* based; q – surrogate population size coef., resp. λ and η_0 – initial value of η)
- model parameters (s'_{\min} – min. number of data for ever using a model, functions f_i , \mathbf{A} – matrix defining the norm, s_{\min} – min. number of data for fitting one RBF network, k – number of folds in cross-validation, e – type of error estimate: MSE, AIC, or BIC)
- database $\mathbf{D} = \{(\mathbf{x}_i^{(C)}, \mathbf{x}_i^{(D)}, y_i)\}_{i=1}^N$ of the original fitness function data

Steps of the algorithm:

- (1) $P \leftarrow$ initial population: take P_0 (if given on input) or randomly generate new one satisfying constraints; $\eta = \eta_0$, *counter* = 0 (if *generation*-based evolution control)
- (2) **if** (not enough data in \mathbf{D} , i.e. $N < s'_{\min}$)
 - evaluate P with the *original fitness*,
 - store the results into the \mathbf{D} , and go to (12)
- if** (*individual*-based evolution control)
 - (3) $\{rbf_j, mse_j, \mathbf{N}_j\}_{j=1}^m \leftarrow$ FitTheModel($s_{\min}, f_i, \|\cdot\|, \mathbf{D}, n, e$) (Fig. 2)
 - (4) evaluate P (of size qp) with the surrogate *model*
 - (5) select p individuals for reevaluation,
 - (6) reevaluate them with the *orig. fitness*, and store the results into \mathbf{D}
- else** (*generation*-based evolution control)
 - if** (η of λ generations already passed: *counter* $\geq \eta$)
 - (7) $\{rbf_j, mse_j, \mathbf{N}_j\}_{j=1}^m \leftarrow$ FitTheModel($s_{\min}, f_i, \|\cdot\|, \mathbf{D}, n, e$)
 - (8) evaluate P (of size p) with the surrogate *model*
 - else**
 - (9) evaluate P with the *original fitness* and store the results into \mathbf{D}
 - (10) adjust η according to the *model's* error estimate $\frac{1}{m} \sum_{j=1}^m mse_j$
 - (11) *counter* \leftarrow (*counter* + 1) mod λ
- (12) $P \leftarrow$ new population generated with selection, mutation and crossover
- (13) continue with (2) **until** (acceptable solution is found **or** solving budget exhausted)

Output: all the individuals proposed by the GA

Fig. 1. Pseudo-code of the surrogate-assisted GA

3.2 Evaluation with the Surrogate Model

Once the model is fitted, it can be used for evaluating individuals resulting from the evolution. The model can be expressed as a set of triples $\{(rbf_j, mse_j, \mathbf{N}_j)\}_{j=1}^m$. For each cluster C_j , rbf_j denotes RBF network parameters, mse_j is the MSE_{CV} obtained from cross-validation, and \mathbf{N}_j is a set of combinations of values of discrete variables grouped within the cluster.

An individual evaluated by the model generally consists of continuous $\mathbf{x}^{(C)}$ and discrete $\mathbf{x}^{(D)}$ parts. At first, the index c of the cluster with the data closest to the individual's discrete values is found

$$c = \arg \min_{j=1, \dots, m} \frac{1}{|\mathbf{N}_j|} \sum_{\mathbf{y} \in \mathbf{N}_j} d_{\text{DISCR}}(\mathbf{x}^{(D)}, \mathbf{y}). \tag{2}$$

Here, d_{DISCR} denotes Hamming or Jaccard metric. Then, the RBF network corresponding to this cluster is used as a surrogate model of the original fitness by computing its return value from the individual's continuous dimensions $\mathbf{x}^{(C)}$

$$\hat{y} = \sum_{i=1}^{g_c^*} \pi_{c,i} f_{c,i}(\|\mathbf{x}^{(C)} - \mathbf{c}_{c,i}\|). \tag{3}$$

If more than one cluster is at the same distance from the individual, the RBF network with the lowest MSE_{CV} is chosen.

FitTheModel($s_{\min}, f_i, \|\cdot\|, \mathbf{D}, n, e$)

Arguments: s_{\min} – min. size of clusters, f_i – RBF type,
 $\|\cdot\|$ – norm parameters, \mathbf{D} – database, n – number of continuous var.,
 e – type of error estimate: MSE, AIC, or BIC

Steps of the procedure:

- (1) cluster the orig. fitness data from \mathbf{D} according to the discrete variable values $\mathbf{x}^{(D)}$ into clusters $\{C_j\}_{j=1}^m$, $C_j = \{l : \mathbf{x}_l^{(D)} \text{ is in the } j\text{-th cluster}\}$ of size at least s_{\min} ; $m \leftarrow$ the resulting number of clusters;

$\{|C_j|\}_{j=1}^m \leftarrow$ the sizes of clusters

$\{\mathbf{N}_j\}_{j=1}^m \leftarrow$ sets of clusters' discrete variables values

- (2) $\{g_j^{\max}\}_{j=1}^m \leftarrow$ the maximal number of components of the j -th network

(for Gaussians $g_j^{\max} = \lfloor \frac{k-1}{2+n} |C_j| \rfloor$)

for each cluster C_j , $j = 1, \dots, m$

for the number of components $g_j = 1, \dots, g_j^{\max}$

- (3) $mse[j, g_j] \leftarrow$ average MSE_{CV} from k -fold cross-validation from fitting of the network for j -th cluster

- (4) $g_j^* \leftarrow$ the number of components of the RBF network with the lowest error of type e (MSE_{CV} from $mse[j, g_j^*]$, AIC or BIC)

- (5) $rbf_j \leftarrow$ parameters of the RBF network with g_j^* components, retrained on all the available data from cluster C_j

- (6) $mse_j \leftarrow mse[j, g_j^*]$

Output: $\{rbf_j, mse_j, \mathbf{N}_j\}_{j=1}^m$

Fig. 2. Pseudo-code of the fitting procedure

4 Implementation and Results of Testing

Our algorithms were implemented in the MATLAB environment. As we have concentrated on the surrogate model for discrete and continuous variables, we have been utilizing the Global Optimization Toolbox (former GADS Toolbox) which provided us with a platform easily adjustable for testing the model on a benchmark optimization task. Similarly, our hierarchical clustering method extends the cluster analysis from the Statistical Toolbox, and we employ a non-linear curve-fitting procedure `lsqcurvefit()` from the Optimization Toolbox for fitting RBF networks.

4.1 Presence of Continuous Variables

Testing the algorithm on real-world data has shown that empirical objective functions are more sensitive to presence of continuous components than on their distinct values. Therefore, we have expanded the discrete-variables-vector of every individual by a binary vector $b = (b_1 \ b_2 \ \dots \ b_n)$ where each component b_i , $i = 1, \dots, n$ indicates presence of i -th continuous variable. This vector helps making clusters containing more similar fitness values.

4.2 Model Fitting

The described model has been tested on a benchmark fitness function designed in [12] to be similar to empirical fitness functions from chemical engineering. The function has a global optimum, found by a continuous optimization method, at approximately $\vartheta(0.35, 0.01, 0.30, 0.15, 0.10) = 547.72$. The model has been 18 times trained on 2000 randomly generated data reaching on validation sets $\text{MSE}_{CV} = (202.04 \pm 12.04)$; the average response in the dataset was $\bar{\vartheta} = 327.0$. Graphical evaluation of the regression results can be seen in Fig. 3.

In addition to a benchmark function, we have trained the model on a dataset from a real application in the field of the optimization of chemical catalyts. Solutions of this task were composed of one discrete and 11 continuous variables, but additional 11 binary variables were added as described in Section 4.1. Although the model fitting results were substantially different from the benchmark case (considering the average response $\bar{y} = 23.80$, the measured $\text{MSE}_{CV} = (222.51 \pm 139.44)$ from 100 fittings is relatively much higher in accordance with the graphical results in Fig. 4), the model is still expected to serve as a surrogate model reasonably well.

4.3 Genetic Algorithm Performance on the Benchmark Fitness

The benchmark fitness enabled us to test the model with the GA. As shown in Table 1, the GA with the surrogate model reaches the same fitness values as the non-surrogate GA using only less than 30 per cent of the original fitness function evaluations (generation-based EC), or it is able to find 1.1-times better solution with 80 per cent of the original fitness evaluations (individual-based EC).

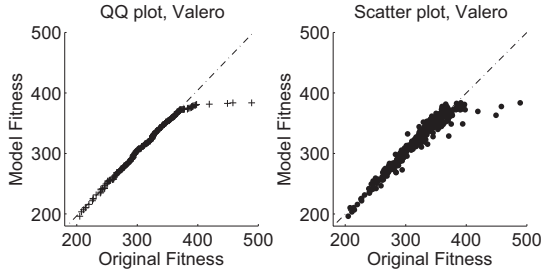


Fig. 3. Quantile-quantile and scatter plot of the RBF-based model of the benchmark fitness [12]

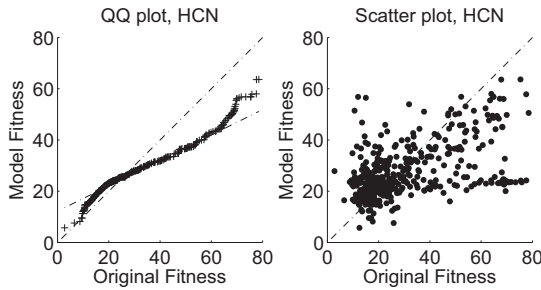


Fig. 4. Quantile-quantile and scatter plot of the RBF-based model of the fitness from catalyst development

Table 1. GA performance without surrogate model and with the RBF-based model; average results from 100 runs of the algorithm

EC settings of the surrogate model	fitness of the best found individual	# of generations to convergence	# of original fitness evaluations
<i>model not used</i>	486.38 ± 56.5	44.89 ± 16.8	4130 ± 1546 (100%)
individual-based	544.73 ± 3.9	35.22 ± 10.1	3241 ± 926 (78.5%)
generation-based	490.28 ± 44.9	49.18 ± 14.5	1185 ± 358 (28.7%)

5 Conclusion

Our work introduced a surrogate model of costly objective functions based on RBF networks which takes into account both discrete and continuous variables. A training algorithm for this model was outlined, and results of testing on both benchmark and real-world data were presented. Using the model resulted in saving approximately 70 per cent of the original evaluations or 10 per cent increase of the final solution quality.

One of the most similar works dealing with surrogate models is the paper of Zhou [3]. He uses RBF networks as a local surrogate model in combination with a global model based on Gaussian processes. Other literature employs

polynomials [6], Gaussian processes [7, 8], or multilayer perceptron networks [9], but all of these publications consider only continuous optimization.

References

1. Olhofer, M., Arima, T., Sendhoff, T.S.B., Japan, G.: Optimisation of a Stator Blade Used in a Transonic Compressor Cascade with Evolution Strategies. *Evolutionary Design and Manufacture*, 45–54 (2000)
2. Holeňa, M., Cukic, T., Rodemerck, U., Linke, D.: Optimization of catalysts using specific, description-based genetic algorithms. *Journal of Chemical Information and Modeling* 48(2), 274–282 (2008)
3. Zhou, Z., Ong, Y., Nair, P., Keane, A.: Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 37(1), 66–76 (2007)
4. Ong, Y.S., Nair, P.B., Keane, A.J., Wong, K.W.: Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. *Knowledge Incorporation in Evolutionary Computation*, 307–332 (2004)
5. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing – A Fusion of Foundations, Methodologies and Applications* 9(1), 3–12 (2005)
6. Hosder, S., Watson, L., Grossman, B.: Polynomial response surface approximations for the multidisciplinary design optimization of a high speed civil transport. *Optimization and Engineering* 2(4), 431–452 (2001)
7. Buche, D., Schraudolph, N., Koumoutsakos, P.: Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 35(2), 183–194 (2005)
8. Ulmer, H., Streichert, F., Zell, A.: Model-assisted Evolution Strategies. *Knowledge Incorporation in Evolutionary Computation*, 333 (2005)
9. Jin, Y., Hüsken, M., Olhofer, M., Sendhoff, B.: Neural networks for fitness approximation in evolutionary optimization. *Knowledge Incorporation in Evolutionary Computation*, 281 (2005)
10. Buhmann, M.D.: Radial basis functions: theory and implementations. Cambridge Univ. Press, Cambridge (2003)
11. Gordon, A.D.: Classification: methods for the exploratory analysis of multivariate data. Chapman and Hall, Boca Raton (1981)
12. Valero, S., Argente, E., Botti, V.: DoE framework for catalyst development based on soft computing techniques. *Computers & Chem. Engineer.* 33(1), 225–238 (2009)

Relevance of Contextual Information in Compression-Based Text Clustering

Ana Granados, Rafael Martínez,
David Camacho, and Francisco de Borja Rodríguez

Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain
{ana.granadosf,r.martinez,david.camacho,f.rodriguez}@uam.es

Abstract. In this paper we take a step towards understanding compression distances by analyzing the relevance of contextual information in compression-based text clustering. In order to do so, two kinds of word removal are explored, one that maintains part of the contextual information despite the removal, and one that does not maintain it. We show how removing words in such a way that the contextual information is maintained despite the word removal helps the compression-based text clustering and improves its accuracy, while on the contrary, removing words losing that contextual information makes the clustering results worse.

1 Introduction

Despite the wide use of compression distances in several domains such as data mining [4], and clustering [7], the interpretation of compression distances results has not been deeply studied yet due to the immense gap between their theoretical foundation and the state-of-the-art compression algorithms used in applications. Whenever some analytical work on compression distances is carried out, it is usually focused on the algebraic manipulation of algorithmic information theory concepts [3,11,20].

In this paper we analyze the relevance of contextual information in compression-based text clustering by exploring two kinds of word removal with the purpose of better understanding compression distances. Text distortion has been used to study the behavior of compression distances previously. For example, the impact of sporadic erasures on the limits of lossless data compression from theoretical and experimental perspectives can be found at [8,9,17]. On the other hand, word substitution has been suggested as a kind of text protection, based on the subsequent automatical detection of such substitutions by looking for discrepancies between words and their contextuales [6]. A technique which reduces the complexity of the documents while preserving most of their relevant information by means of annealing text distortion is presented in [8,9]. That technique, which is based on word removal, fine-tunes the representation of the documents, and as a consequence improves the non-distorted clustering results. In this paper, we explore a new removal technique, with the aim of better understanding why that occurs.

The paper is structured as follows. Section 2 describes the word removal techniques, reviews the compression distance used in our work, and describes the clustering assessment. Section 3 gathers and analyzes the obtained results. Section 4 summarizes the conclusions.

2 Experimental Framework

Two word removal techniques are used in this paper with the aim of studying the relevance of contextual information in compression-based text clustering. The word removal techniques are explained in detail in the first section. The compression distance evaluated in this paper is reviewed in the second section. The compression-based clustering algorithm used in this paper is discussed in the third section.

2.1 Distortion Techniques: Word Removal

Other works have shown that distorting the documents by removing the stop-words may have beneficial effects both in terms of accuracy and computational load when clustering documents or when retrieving information from them [19]. There are two main approaches to word removal, one in which a generic fixed stop-word list is used [8,9,13,16], and other in which this list is generated from the collection itself [18]. The first approach is ‘safer’ in terms of maintaining the most relevant information of the documents, while the second approach produces a more aggressive word removal.

We used the first approach in previous work [8,9] with the aim of exploring how the selection of the words to be removed affects the clustering results when a compression-based clustering method is used. It was shown that selecting the most frequent words of the English language the non-distorted clustering results could be improved if a specific word removal technique was used. In this paper, we explore a new removal technique, with the aim of better understanding why that occurs. Our intuition is that not only the presence of the remaining words affects the clustering algorithm, but also the contextual information remaining in the documents after the use of the above mentioned word removal technique. The purpose of our work is to investigate whether our intuition is right or not.

In order to do so, we progressively remove the most frequent words of the English language from the documents using two different word removal techniques. The frequencies of the words are estimated using the British National Corpus (BNC) [5]. We select the words to be removed by calculating the cumulative sum of the word-frequencies. Thus, we select the words that accumulate a frequency of 0.1, 0.2, 0.3, and so on, until 1.0, where finally all the words belonging to the BNC are selected.

Depending on the way in which the words are removed from the documents we have two different word removal methods:

Table 1. Word removal methods. An extraction of a document on Anemia.

Original text	Maintaining contextual information	Losing contextual information
anemia is a condition in which the body does not have enough healthy red blood cells. red blood cells provide oxygen to body tissues.	anemia ** * ***** ** ***** ** * **** ** * ***** healthy *** ***** ***** ** * ***** ***** oxygen ** **** tissues	anemia * ** ** * ** * ** ** * ** * ** * *** healthy ***** ***** ***** ***** ** * ***** oxygen ***** ***** tissues

- *Asterisk per character*: each character of the word is replaced by an asterisk. This substitution method was analyzed in [8,9]. It is important highlighting that this method keeps the contextual information despite the removal because the place of appearance of the removed words and their lengths are maintained (see second column in Table 1).
- *Asterisk per character and random sorting asterisks*: each character of the word is replaced by an asterisk as well, but after the replacing, the strings of asterisks are randomly sorted. That is, the remaining words are maintained in their original places of appearance, while the removed words are not. This method is created in order to study whether the structure of the contextual information is important to the clustering or not. It is important noting that this technique does not maintain the contextual information because the place of appearance of the removed words is lost when this method is used (see third column in Table 1).

2.2 The Similarity Metric: The Normalized Compression Distance

The *normalized compression distance* (NCD) [3,11] provide a measure of the similarity between two objects, x and y , using compressors. The definition is as follows

$$NCD(x, y) = \frac{\max\{C(xy) - C(x), C(yx) - C(y)\}}{\max\{C(x), C(y)\}}, \tag{1}$$

where C is a compression algorithm, $C(x)$ is the size of the C -compressed version of x , and $C(xy)$ is the compressed size of the concatenation of x and y . NCD generates a non-negative number $0 \leq NCD(x, y) \leq 1$. Distances near 0 indicate similarity between objects, while distances near 1 reveal dissimilarity.

For example, if $x \approx y$ then the compressed size of the concatenation xy ($C(xy)$) would be approximately equal to the compressed size of x ($C(x)$). The same would happen with $C(xy)$ and $C(y)$. Thus, the denominator of the formula would be near 0, and therefore the NCD would be near 0 too.

The NCD is just one of the many similarity distances that use compression algorithms. Others [11,10,20], are small variations and can be easily reduced to it, as it is possible to prove that this distance minorizes (is as good as) any other

that can be computed by a universal Turing machine [14]. In this paper, we use the NCD to evaluate how the different word removal techniques affect the loss of information, and therefore the clustering.

2.3 Compression-Based Text Clustering

In order to study how the word removal techniques affect the compression-based text clustering, we use the CompLearn Toolkit [2] which implements the clustering algorithm described in [3]. This clustering algorithm comprises two phases. First, the NCD matrix is calculated using a compressor (in our experiments we use the LZMA compressor [12]). Second, the NCD matrix is used as input to the clustering phase and a dendrogram is generated as output. A dendrogram is an undirected binary tree diagram, frequently used for hierarchical clustering, that illustrates the arrangement of the clusters produced by a clustering algorithm. A representative part of a dendrogram can be seen in Fig 1. Each leaf of the dendrogram corresponds to a document and its label helps us to easily analyze the quality of the dendrogram obtained, because each label starts with the name of the cluster in which the document should be included.

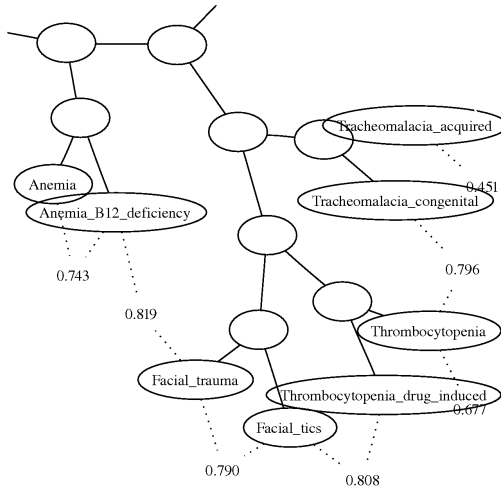


Fig. 1. Part of a dendrogram from the MedlinePlus repository. Each leaf of the dendrogram corresponds to a document. This part of the dendrogram contains documents belonging to four different classes: Anemia, Facial, Thrombocytopenia, and Tracheomalacia.

We define the clustering error associated to a dendrogram using the concept of distance between two nodes, which is the minimum number of internal nodes needed to go from one to the other. First, we add all the pairwise distances between nodes starting with the same string (note that all those nodes should be clustered together). After calculating this addition, we subtract the pairwise distances addition of the dendrogram that corresponds to the perfect clustering

from the total quantity previously obtained. As a result, we obtain a measure of the clustering error that would be 0 if the dendrogram clusters perfectly all the documents, and in general, the bigger the measure, the worse the clustering would be.

3 Experimental Results

Two different data sets have been used to evaluate the relevance of contextual information in compression-based text clustering. One of them consists of fourteen classical books from universal literature. The other one consists of 50 documents from the MedlinePlus repository [15]. The size of the data sets is not very big due to the fact that the CompLearn uses the quartet tree method to generate the dendrogram and that algorithm has an asymptotical cost of $O(n^3)$ from version 1.1.3 onwards, which is one of the fastest clustering methods around.

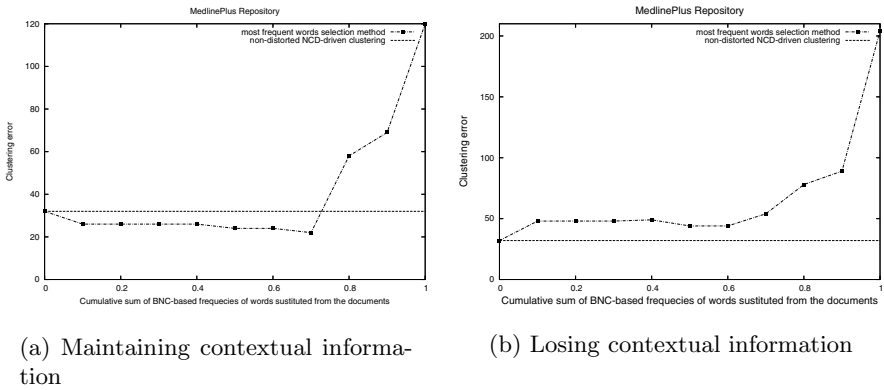


Fig. 2. Clustering results for the MedlinePlus data set. The results that correspond to the *Asterisk per character* word removal method can be seen in (a), and the results corresponding to the *Asterisk per character and random sorting* word removal method can be seen in (b). It can be noticed that losing the contextual information the clustering results get worse.

The obtained clustering results can be seen in Figs 2 and 3. Each figure corresponds to a data set, and each panel corresponds to a word removal method. In all the figures the value on the horizontal axis corresponds to the cumulative sum of the BNC-based frequencies of the words substituted from the documents. The constant line corresponds to the non-distorted NCD-driven clustering error. We depict it as a constant line although it only has sense for a cumulative sum of frequencies of 0, because it is easier to see the difference between the line and the clustering error curves.

Figs 2(a) and 3(a) depict the clustering results that correspond to the *Asterisk per character* removal method, while Figs 2(b) and 3(b) depict the ones that correspond to the *Asterisk per character and random sorting asterisks* removal method. Analyzing those figures, it can be observed that the results that

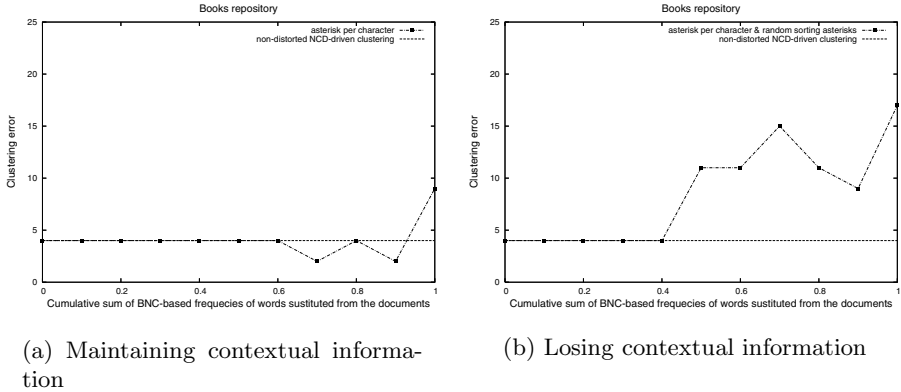


Fig. 3. Clustering results for the Books data set. It can be noticed that losing the contextual information the clustering results get worse. Thus, the results are consistent with the ones obtained using the MedlinePlus data set.

correspond to the *Asterisk per character* removal method are better than the ones obtained when using the *Asterisk per character and random sorting asterisks* removal method. It is important noting that the place of appearance of the removed words and their lengths are maintained when the *Asterisk per character* removal technique is used, while it is not maintained when the *Asterisk per character and random sorting asterisks* removal technique is used (see Table 1). Thus, it seems that maintaining the contextual information improves the clustering results, while on the contrary, losing the contextual information makes them worse. Therefore, not only the presence of the remaining words is important to the compression-based text clustering but also their contextual information.

4 Conclusions

In this paper we have moved a small step towards understanding compression distances by analyzing the relevance of contextual information in compression-based text clustering. In order to do so, we have explored two different word removal techniques, one that maintains the contextual information despite the word removal, and one that does not maintains it (see section 2.1). In terms of implementation, we have used the CompLearn Toolkit [2] to perform the clustering (see section 2.3).

Two different data sets have been used in this paper, obtaining similar results in all the cases. Analyzing these results it can be noticed that using the word removal technique that maintains part of the contextual information, the non-distorted clustering results can be improved, while using the other one the non-distorted clustering results get worse. Therefore, maintaining the contextual information helps the clustering algorithm to perform better. It seems that preserving the contextual information the compressor is able to capture the

internal structure of the documents in spite of the word removal. Consequently, the compressor obtains more reliable similarities, and the non-distorted clustering results can be improved.

Acknowledgments

This work was supported by the Spanish Ministry of Education and Science under TIN 2007-65989, TIN 2007-64718, and CAM S-SEM-0255-2006.

References

1. Benedetto, D., Caglioti, E., Loreto, V.: Language Trees and Zipping. *Physical Review Letters* 88(4), 48702 (2002)
2. Cilibrasi, R., Cruz, A.L., de Rooij, S., Keijzer, M.: CompLearn Toolkit, <http://www.complearn.org/>
3. Cilibrasi, R., Vitanyi, P.M.B.: Clustering by compression. *IEEE Transactions on Information Theory* 51(4), 1523–1545 (2005)
4. Cilibrasi, R., Vitanyi, P.M.B.: The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering* 19(3), 370–383 (2007)
5. BNC Consortium. British National Corpus. Oxford University Computing Services, Oxford, <http://www.natcorp.ox.ac.uk/>
6. Fong, S., Roussinov, D., Skillicorn, D.B.: Detecting word substitutions in text. *IEEE Transactions on Knowledge and Data Engineering* 20(8), 1067–1076 (2008)
7. González, A., Granados, A., Camacho, D., Rodríguez, F.: Influence of music representation on compression-based clustering. In: *IEEE Congress on Evolutionary Computation* (2010)
8. Granados, A., Cebrián, M., Camacho, D., Rodríguez, F.: Reducing the loss of information through annealing text distortion. *IEEE Transactions on Knowledge and Data Engineering* (in press, 2010)
9. Granados, A., Cebrián, M., Camacho, D., Rodríguez, F.: Evaluating the impact of information distortion on normalized compression distance. In: Barbero, Á. (ed.) *ICMCTA 2008*. LNCS, vol. 5228, pp. 69–79. Springer, Heidelberg (2008)
10. Kraskov, A., Stoegbauer, H., Andrzejak, R.G., Grassberger, P.: Hierarchical clustering using mutual information. *Europhysics Letters* 70(2), 278–284 (2005)
11. Li, M., Chen, X., Li, X., Ma, B., Vitanyi, P.: The similarity metric. *IEEE Transactions on Information Theory* 50(12), 3250–3264 (2004)
12. Pavlov, I.: LZMAX, <http://www.7-zip.org/sdk.html>
13. Salton, G.: *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co. Inc., Boston (1989)
14. Turing, A.: On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society* 2(42), 230–265 (1936)
15. U.S. National Library of Medicine and National Institutes of Health MedlinePlus Health Information, MedlinePlus website, <http://medlineplus.gov/>
16. Van Rijsbergen, C.J.: *Information Retrieval*. Butterworth-Heinemann, Newton (1979)

17. Verdú, S., Weissman, T.: The information lost in erasures. *IEEE Transactions on Information Theory* 54(11), 5030–5058 (2008)
18. Wilbur, W.J., Sirotkin, K.: The automatic identification of stop words. *Journal of Information Science* 18(1), 45 (1992)
19. Yang, Y.: Noise reduction in a statistical approach to text categorization. In: *Proceedings of SIGIR*, pp. 256–263 (1995)
20. Zhang, X., Hao, Y., Zhu, X., Li, M.: Information distance from a question to an answer. In: *KDD '07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 874–883. ACM, New York (2007)

Simple Deterministically Constructed Recurrent Neural Networks

Ali Rodan and Peter Tiño

School of Computer Science, University of Birmingham
Birmingham B15 2TT, United Kingdom
{a.a.rodan,P.Tino}@cs.bham.ac.uk

Abstract. A large number of models for time series processing, forecasting or modeling follows a state-space formulation. Models in the specific class of state-space approaches, referred to as Reservoir Computing, fix their state-transition function. The state space with the associated state transition structure forms a reservoir, which is supposed to be sufficiently complex so as to capture a large number of features of the input stream that can be potentially exploited by the reservoir-to-output readout mapping. The largely “black box” character of reservoirs prevents us from performing a deeper theoretical investigation of the dynamical properties of successful reservoirs. Reservoir construction is largely driven by a series of (more-or-less) ad-hoc randomized model building stages, with both the researchers and practitioners having to rely on a series of trials and errors. We show that a very *simple deterministically constructed* reservoir with simple cycle topology gives performances comparable to those of the Echo State Network (ESN) on a number of time series benchmarks. Moreover, we argue that the memory capacity of such a model can be made arbitrarily close to the proved theoretical limit.

Keywords: Recurrent neural networks, Echo state networks, Memory capability, Time series prediction.

1 Introduction

Recently there has been an outburst of research activity in the field of reservoir computing (RC) [3]. RC models are dynamical models for processing time series that make a conceptual separation of the temporal data processing into two parts: 1) representation of temporal structure in the input stream through a non-adaptable dynamic “*reservoir*”, and 2) a memoryless easy-to-adapt *readout* from the reservoir. For a comprehensive recent review of RC see [3]. Perhaps the simplest form of the RC model is the Echo State Network (ESN) [3].

Roughly speaking, ESN is a recurrent neural network with a non-trainable sparse recurrent part (reservoir) and a simple linear readout. Connection weights in the ESN reservoir, as well as the input weights are randomly generated. The reservoir weights are scaled so as to ensure the “Echo State Property” (ESP): the reservoir state is an “echo” of the entire input history. Typically, spectral radius of the

reservoir's weight matrix W is made $< 1^1$. ESN has been successfully applied in time-series prediction, speech recognition, dynamic pattern classification and language modeling [3].

Formally, ESN (shown in Fig.1 (A)) is a discrete-time recurrent neural network with K input units, N recurrent (reservoir) units and L output units. The activation vectors of the input, internal, and output units at time step t are denoted by $s(t)$, $x(t)$, and $y(t)$, respectively. The connections between the input units and the recurrent units are given by an $N \times K$ weight matrix V , connections between the internal units are collected in an $N \times N$ weight matrix W , and connections from internal units to output units are given by an $L \times N$ weight matrix U .

The recurrent units are updated according to²:

$$x(t+1) = f(Vs(t+1) + Wx(t)), \quad (1)$$

where f is the reservoir activation function (typically linear, tanh or some other sigmoidal function). The linear readout is computed as³:

$$y(t+1) = Ux(t+1). \quad (2)$$

The output weights U are typically trained both offline and online by minimizing the Normalized Mean square Error:

$$NMSE = \frac{\langle \|\hat{y}(t) - y(t)\|^2 \rangle}{\langle \|y(t) - \langle y(t) \rangle\|^2 \rangle}, \quad (3)$$

where $\hat{y}(n)$ is the readout output, $y(n)$ is the desired output (target), $\|\cdot\|$ denotes the Euclidean norm and $\langle \cdot \rangle$ denotes the empirical mean.

Elements of W and V are fixed prior to training with random values drawn from a uniform distribution over a (typically) symmetric interval. To account for ESP, the reservoir connection matrix W is typically scaled as $W \leftarrow \alpha W / |\lambda_{max}|$, where $|\lambda_{max}|$ is the spectral radius of W and $0 < \alpha < 1$ is a scaling parameter [3].

Many extensions of the original standard ESN have been suggested in the literature, e.g. intrinsic plasticity, decoupled reservoirs, leaky-integrator reservoir units etc. However, there are still serious problems preventing ESN to become a widely accepted tool (see e.g. [8,9]): (1) there are properties of the reservoir that are poorly understood, (2) specification of the reservoir and input connections require numerous trials and even luck, (3) the random connectivity and weight structure of the reservoir is unlikely to be optimal and does not give a clear insight into the reservoir dynamics organization.

In this paper we would like to identify the minimum complexity one needs in constructing reservoirs competitive with standard ESN. We also study theoretical memory capacity of such minimum complexity reservoirs.

¹ Note that this is not necessary and sufficient condition for ESP.

² There are no feedback connections from the output to the reservoir and no direct connections from the input to the output.

³ The reservoir activation vector is extended with a fixed element accounting for the bias term.

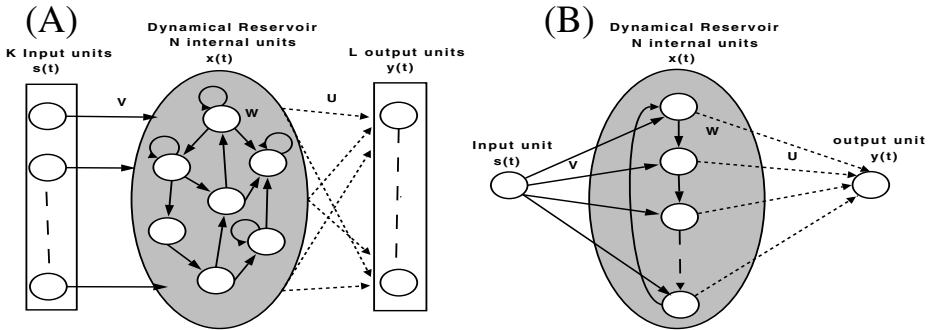


Fig. 1. (A) Echo state network (ESN) and (B) Simple Cycle Reservoir (SCR) models

2 Simple Cycle Reservoir (SCR)

As a simple alternative to the standard *ESN reservoir* introduced above, we consider a *Simple Cycle Reservoir (SCR)* (Fig. 1 (B)) with recurrent units organized in a cycle⁴. Nonzero elements of W are on the lower sub-diagonal $W_{i+1,i} = r$ for $i = 1 \dots N - 1$, and at the upper-right corner $W_{1,N} = r$ where $r > 0$ is the reservoir weight connection.

Standard ESN have either full or sparse input-to-reservoir connectivity (full connectivity in our experiments), with weights generated randomly according to some (typically) uniform and symmetric around zero distribution with support $[-a, a]$, $a > 0$. With SCR topology we use full input-to-reservoir connectivity with the same absolute value connection weight $v > 0$. We found that any attempt to impose a regular pattern on the input weight signs (e.g. a periodic structure of the form $+ - + - \dots$, or $+ - - + - - \dots$ etc.) lead to performance deterioration. It proved to be sufficient to relate the sign pattern to *deterministically* generated aperiodic sequences. Such sign patterns worked universally well across all benchmark data sets used in this study. We generated the universal input sign patterns in two ways:

1. (*SCR-PI*): The input signs are determined from decimal expansion $d_0.d_1d_2d_3\dots$ of irrational numbers (in our case π). The first N decimal digits d_1, d_2, \dots, d_N are thresholded at 4.5, e.g. if $0 \leq d_n \leq 4$ and $5 \leq d_n \leq 9$, then the n -th input connection sign (linking the input to the n -th reservoir unit) will be $-$ and $+$, respectively,
2. (*SCR-L*): The input signs are determined by the first N iterates in binary symbolic dynamics of the logistic map $f(x) = 4x(1-x)$ in a chaotic regime (initial condition was 0.33, generating partition for symbolic dynamics with cut-value at $1/2$).

The values v (for SCR) and a (for ESN) are chosen on the validation set.

⁴ We also considered other simple reservoir topologies (see [2]). Due to space limitations, the complete results will be published elsewhere.

3 Experiments

For each data set and each model class (ESN, SCR) we picked on the validation set a model representative to be evaluated on the test set. Readout training was done using Ridge Regression⁵ For the SCR architecture the model representative is defined by the ridge regression regularization factor λ , input weight value v and the reservoir weight r . For the ESN architecture, the model representative is specified by λ , input weight connectivity, reservoir sparsity, input weight range and spectral radius of the weight matrix.

For each model we have calculated NMSE over 10 simulation runs. Our experiments are organized along two degrees of freedom: **1)** input weight structure and **2)** reservoir size. We consider *linear reservoirs* that consist of neurons with identity activation function, as well as *non-linear reservoirs* consisting of neurons with the commonly used tangent hyperbolic (tanh) activation function. The models were evaluated on the following data sets:

10th order NARMA: We considered a NARMA system of order 10 [6]:

$$y(t+1) = 0.3y(t) + 0.05y(t) \sum_{i=0}^9 y(t-i) + 1.5s(t-9)s(t) + 0.1, \quad (4)$$

where $y(t)$ is the system output at time t , $s(t)$ is the system input at time t (an i.i.d stream of values generated uniformly from an interval $[0, 0.5]$). The current output depends on both the input and the previous outputs. In general, modeling this system is difficult, due to the non-linearity and possibly long memory. NARMA sequence has a length of 8000 samples where 2000 were used for training, 3000 for validation, and the remaining 3000 used for testing the models. The first 200 values from training, validation and testing sequences were used as the initial washout period. The networks were trained on system identification task to output $y(t)$ based on $s(t)$. The input $s(t)$ and target data $y(t)$ are shifted by -0.5 and scaled by 2 as in [6].

IPIX Radar: The sequence (used in [8]) contains 2000 values. The target signal is the sea clutter data (the radar backscatter from an ocean surface) The first 800 values were used for training, the next 500 for validation, and the remaining (700 values) was used for testing the models. The first 100 values from training, validation and testing sequences were used as the initial washout period. The task was to predict $y(t+1)$ and $y(t+5)$ when $y(t)$ presented at the network input (5 step prediction).

Sunspot series: The dataset (obtained from [1]) contains 3100 sunspots numbers from Jan 1749 to April 2007, where the first 1600 values were used for training, the next 500 for validation, and the remaining (1000 values) for testing the models. The first 100 values from training, validation and testing sequences were used as the initial washout period. The task was to predict the next value $y(t+1)$.

⁵ We also tried other forms of readout training, such as recursive least squares, pseudoinverse and singular value decomposition. Ridge regression lead to the most accurate results.

Table 1. Mean NMSE results across 10 simulation runs (standard deviations in parenthesis) for traditional ESN and SCR topologies

Dataset	N	ESN	SCR-PI	SCR-L
NARMA	50	0.157 (0.0177)	0.155 (0.0153)	0.152 (0.0138)
	100	0.0970 (0.0160)	0.0923 (0.0116)	0.0918 (0.0132)
	150	0.0580 (0.00794)	0.0606 (0.00855)	0.0594 (0.00892)
	200	0.0428 (0.0166)	0.0415 (0.00792)	0.0408 (0.00837)
Nonlinear communication channel	50	0.0048 (4.06E-4)	0.0046 (1.82E-04)	0.0033 (1.09E-04)
	100	0.0033 (4.42E-4)	0.0016 (7.96E-05)	0.0015 (8.85E-5)
	150	0.0021 (4.01E-4)	0.0012 (7.12E-05)	0.0012 (4.56E-05)
	200	0.0019 (1.71E-4)	8.85E-04 (2.55E-05)	9.39E-04 (3.33E-05)

Nonlinear Communication Channel: The data set was created as follows [5]: First, an i.i.d. sequence $d(t)$ of symbols transmitted through the channel is generated by randomly choosing values from $\{-3, -1, 1, 3\}$ (uniform distribution). Then, $d(t)$ values are used to form a sequence $q(t)$ through a linear filter

$$q(t) = 0.08d(t+2) - 0.12d(t+1) + d(t) + 0.18d(t-1) - 0.1d(t-2) \\ + 0.09d(t-3) - 0.05d(t-4) + 0.04d(t-5) + 0.03d(t-6) + 0.01d(t-7).$$

Finally, a nonlinear transformation is used to $q(n)$ to produce the signal $s(n)$:

$$s(t) = q(t) + 0.0036q(t)^2 - 0.11q(t)^3. \quad (5)$$

The task was to predict $d(t-2)$ when $s(t)$ was presented as network input. Following [5], the input $s(t)$ signal was shifted +30.

The results are presented in tables 1 and 2. Note that NARMA and Nonlinear communication channel are randomized series, producing a different series in each of the 10 simulation runs. That is why the NMSE results for SCR models fluctuate across simulation runs, even though the SCR models are constructed in a completely deterministic manner. Construction of standard ESN involves a stochastic component and so the NMSE results fluctuate for all data sets (in case of IPIX Radar and sunspot series, 10 simulation runs involved different realizations of the best performing ESN architecture picked on the validation set). For IPIX Radar and Sunspot series, the results are presented for models with $N = 80$ and $N = 200$ recurrent units, respectively. These model sizes represent a typical behavior - the results did not improve significantly with further reservoir expansion. In most cases the SCR architectures perform slightly better than standard ESN, even though the differences are not statistically significant.

4 Short-Term Memory Capacity of SCR

Jaeger [4] suggests to quantify the inherent capacity of recurrent network architectures to represent past events through a measure correlating the past events in an i.i.d. input stream with the network output. In particular, assume that the

Table 2. Mean NMSE results for traditional ESN across 10 simulation runs (standard deviations in parenthesis). NMSE results for SCR topologies. For IPIX Radar and Sunspot series, the results are presented for models with $N = 80$ and $N = 200$ recurrent units, respectively.

Dataset	prediction horizon	ESN	SCR-PI	SCR-L
IPIX Radar	1	0.00115 (2.48E-05)	0.00109	0.00108
	5	0.0301 (8.11E-04)	0.0299	0.0297
Sunspot	1	0.1042 (8.33E-5)	0.1063	0.1059

network is driven by a univariate stationary input signal $s(t)$. For a given delay k , we consider the network with optimal parameters for the task of outputting $s(t - k)$ after seeing the input stream $\dots s(t - 1)s(t)$ up to time t . The goodness of fit is measured in terms of the squared correlation coefficient between the desired output (input signal delayed by k time steps) and the network output $y(t)$:

$$MC_k = \frac{Cov^2(s(t - k), y(t))}{Var(s(t)) Var(y(t))}, \tag{6}$$

where Cov denotes the covariance and Var the variance operators. The short term memory capacity is then [4] $MC = \sum_{k=1}^{\infty} MC_k$. Jaeger [4] proved that for *any* recurrent neural network with N recurrent neurons, under the assumption of i.i.d. input stream, MC cannot exceed N . Theorem 1 assures that (under the assumption of zero-mean i.i.d. input stream) that the memory capacity of linear SCR architecture can be made arbitrarily close to the theoretical limit. Detailed proof is quite involved and can be found in [2].

Since there is a single input (univariate time series), the input matrix V is an N -dimensional vector $V = (V_1, V_2, \dots, V_N)^T$. Consider a vector rotation operator rot_1 that cyclically rotates vectors by 1 place to the right, e.g. $rot_1(V) = (V_N, V_1, V_2, \dots, V_{N-1})^T$. For $k \geq 1$, the k -fold application of rot_1 is denoted by rot_k . The $N \times N$ matrix with k -th column equal to $rot_k(V)$ is denoted by Ω , e.g. $\Omega = (rot_1(V), rot_2(V), \dots, rot_N(V))$.

Theorem 1. *Consider a linear SCR network with reservoir weight $0 < r < 1$ and an input weight vector V such that the matrix Ω is regular. Then the SCR network memory capacity is equal to*

$$MC = N - (1 - r^{2N}).$$

Sketch of the Proof:

It can be shown that the covariance matrix of recurrent activations can be expressed as $R = \frac{\sigma^2}{1-r^{2N}} \Omega^T \Gamma^2 \Omega$, where σ^2 is the variance of the input source and $\Gamma = \text{diag}(1, r, r^2, \dots, r^{N-1})$. Under the square loss assumption this leads to optimal (delay conditional) readout weights $U = (1 - r^{2N})^{-1} r^k A^{-1} rot_k(V)$, where $A = \Omega^T \Gamma^2 \Omega$.

Denoting $(\text{rot}_k(V))^T A^{-1} \text{rot}_k(V)$ by ζ_k , one can show that if Ω is regular, then $\zeta_N = 1$ and $\zeta_k = r^{-2k}$, $k = 1, 2, \dots, N - 1$. From $MC_k = r^{2k} (1 - r^{2N}) \zeta_k$ and $MC = MC_{\geq 0} - MC_0$, where $MC_{\geq 0} = \sum_{k=0}^{\infty} MC_k$, one can deduce that $MC = \sum_{k=1}^N r^{2k} \zeta_k$.

Since $r^{2k} \zeta_k = 1$ for $k = 1, 2, \dots, N - 1$, and $r^{2N} \zeta_N = r^{2N}$. It follows that $MC = N - 1 + r^{2N}$. *Q.E.D.*

5 Discussion and Conclusion

Models in the special class of recurrent neural networks, termed reservoir computing models, differ in how the fixed reservoir is constructed and what form the readout takes. Echo state networks (ESN) (used in this study) typically have a linear readout and a reservoir formed by a fixed recurrent neural network type dynamics. *Liquid state machines* (LSM) [3] have also mostly linear readout and the reservoirs are driven by the dynamics of a set of coupled spiking neuron models. *Fractal prediction machines* (FPM) [11] have been suggested for processing symbolic sequences. Their reservoir dynamics is driven by fixed affine state transitions over an N -dimensional interval. The readout is constructed as a collection of multinomial distributions over next symbols.

The field of reservoir computing has been growing rapidly with dedicated special sessions at conferences and special issues of journals. Reservoir computing has been successfully applied in many practical applications. However, reservoir computing has been rightfully criticized for not being principled enough [7]. There have been several attempts to address the question of what exactly is a ‘good’ reservoir for a given application (e.g. [9]), but no coherent theory has yet emerged. The largely “black box” character of reservoirs prevents us from performing a deeper theoretical investigation of the successful reservoirs. Reservoir construction is largely driven by a series of (more-or-less) ad-hoc randomized model building stages, with both the researchers and practitioners having to rely on a series of trials and errors. Often reservoirs have been evolved in a costly and difficult to analyze evolutionary computation setting.

On four data sets of different origin and characteristics, we have shown:

1. A very simple cycle reservoir topology is sufficient for obtaining performances comparable to those of standard ESN. Only two free parameters are sufficient, irrespective of the reservoir size - the reservoir and input weights.
2. Competitive reservoirs to standard ESN can be constructed in a completely deterministic manner: The reservoir connections all have the same weight value. The input connections have the same absolute value with sign distribution following one of the universal deterministic aperiodic patterns.

Compared with standard ESN, recent extensions and reformulations of reservoir models often achieved improved performances [6,8], at the price of even less transparent models and less interpretable dynamical organization. We stress that the main purpose of this study is not a construction of yet another reservoir model achieving an (incremental or more substantial) improvement over

the competitors on the benchmark data sets. Instead, we would like to propose as simplified as possible reservoir construction, without any stochastic component, that while competitive with standard ESN, yields transparent models, more amenable to theoretical analysis. We also considered other simple reservoir topologies, such as bi-directional cycles, tap-delay line with/without backward connections etc., but the SCR topology emerged as the simplest best performing to reservoir organization (see [2]).

The simple deterministic nature of our SRC model enabled us to calculate analytically its memory capacity, which can be made arbitrarily close to the proved theoretical limit. Obtaining such a result for standard ESN is not possible, since (1) for standard ESN one could only calculate the mean memory capacity (with respect to randomization of ESN construction) and (2) closed form equality is very difficult to obtain for reservoirs with a range of possible recurrent/input weight values.

Acknowledgments. The authors would like to thank Jochen Steil for stimulation discussions on reservoir computing.

References

1. Sunspot numbers. National Geophysical Data Center, NGDC (2007)
2. Rodan, A., Tino, P.: Minimum Complexity Echo State Network. Technical Report CSRP-10-02-7, School of Computer Science, University of Birmingham (2010), http://www.cs.bham.ac.uk/~pxt/PAPERS/ESN_tr.pdf
3. Lukosevicius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. *Computer Science Review* 3(3), 127–149 (2009)
4. Jaeger, H.: Short term memory in echo state networks. German National Research Center for Information Technology, Technical Report GMD report 152 (2002)
5. Jaeger, H., Hass, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. *Science* 304, 78–80 (2004)
6. Schrauwen, B., Wardermann, M., Verstraeten, D., Steil, J., Stroobandt, D.: Improving reservoirs using intrinsic plasticity. *Neurocomputing* 71(7-9), 1159–1171 (2008)
7. Prokhorov, D.: Echo state networks: appeal and challenges. In: Proc. of International Joint Conference on Neural Networks, Montreal, Canada, pp. 1463–1466 (2005)
8. Xue, Y., Yang, L., Haykin, S.: Decoupled echo state networks with lateral inhibition. *Neural Networks* 20, 365–376 (2007)
9. Ozturk, M.C., Xu, D., Principe, J.: Analysis and design of echo state network. *Neural Computation* 19(1), 111–138 (2007)
10. Cernansky, M., Tino, P.: Predictive modelling with echo state networks. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 778–787. Springer, Heidelberg (2008)
11. Tino, P., Dorffner, G.: Predicting the future of discrete sequences from fractal representations of the past. *Machine Learning* 45(2), 187–218 (2001)

Non-negative Matrix Factorization Implementation Using Graphic Processing Units

Noel Lopes^{1,2} and Bernardete Ribeiro¹

¹ CISUC - Center for Informatics and Systems of University of Coimbra, Portugal

² UDI/IPG - Research Unit, Polytechnic Institute of Guarda, Portugal
noel@ipg.pt, bribeiro@dei.uc.pt

Abstract. Non-Negative Matrix Factorization (NMF) algorithms decompose a matrix, containing only non-negative coefficients, into the product of two matrices, usually with reduced ranks. The resulting matrices are constrained to have only non-negative coefficients. NMF can be used to reduce the number of characteristics in a dataset, while preserving the relevant information that allows for the reconstruction of the original data. Since negative coefficients are not allowed, the original data is reconstructed through additive combinations of the parts-based factorized matrix representation. A Graphics Processing Unit (GPU) implementation of the NMF algorithms, using both the multiplicative and the additive (gradient descent) update rules is presented for the Euclidean distance as well as for the divergence cost function. The performance results on an image database demonstrate extremely high speedups, making the GPU implementations excel by far the CPU implementations.

1 Introduction

The boundless Internet disseminates a staggering amount of data and its growth is increasing so much that humans need help from automated learning systems for extracting relevant information. Moreover, the continuous decrease in the costs for storing more (and more complex) data and the systematic emergence of an increasing number of devices, capable of gathering, storing and sharing information, anytime and anywhere, resulted in abundant wealth of data. To cope and take advantage of these large repositories of data, intelligent methods for the extraction of knowledge are becoming increasingly relevant. However, systems must be able to distinguish between useful information and accessory (or noisy) data. In this context, Non-Negative Matrix Factorization (NMF) can be used to reduce the data dimensionality, while preserving the information of the most relevant features in order to rebuild accurate approximations of the original data.

NMF is a recent unsupervised technique with applications in image processing, text mining, document clustering, multimedia data, bioinformatics, microarray data analysis, molecular pattern discovery, physics, air emission control, collaborative filtering and financial analysis among others [1-3]. However, as many other Machine Learning (ML) methods, NMF algorithms are computationally expensive, especially when dealing with high volumes of data. Thus, instead of

relying on traditional standalone implementations, creating multi-core architecture counterpart implementations is of significant importance.

Recently, the Graphics Processing Unit (GPU) emerged as a major player in the trend shifting of the computing industry from single core to multi-core architectures. Current GPUs are general-purpose engines, for high throughput floating-point computation, with enough power and flexibility to accelerate non-graphics applications [4, 5]. Moreover, they offer a peak floating-point performance far superior to those of modern CPUs. On top of this, they are relatively inexpensive and regularly replaced by new generation GPUs with increasing computational power. The combination of these factors with the appearance of CUDA (Compute Unified Device Architecture) which supports for accessible programming interfaces, has resulted in the GPU becoming the platform of choice in the scientific community [6]. Thus the GPU is the ideal candidate to accelerate many ML algorithms in general and NMF algorithms in particular.

The remainder of this paper is organized as follows. The next section presents a brief overview of the CUDA programming model and architecture. Section three introduces the NMF algorithms whose GPU parallel implementation is described in section four. Section five discusses the results and the speedups obtained for a well-known dataset. Finally, in section six conclusions are drawn.

2 Compute Unified Device Architecture

The CUDA programming model, empowers developers with the possibility of using industry-standard languages, such as C and C++, to create special functions, named kernels. Kernels explicitly specify data parallel computations to be executed on a device (GPU) that operates as a co-processor to the host (CPU) running the program. They define the sequence of work to be carried out in each thread mapped over a domain (the set of threads to be invoked) [7]. Threads must be organized into independent blocks, which in turn form a grid (observe Figure 1(a)). Kernels can access a set of intrinsic thread-identification variables (e.g. *threadIdx*, *blockIdx*, *blockDim* and *gridDim*) that allow them to identify the actual thread location in the domain [7].

The CUDA programming model is supported by an architecture built around a scalable array of multi-threaded Streaming Multiprocessors (SMs), as shown in Figure 1(b). On newer devices (GPUs), such as the GTX 480, each SM contains 32 scalar processor cores, while on older devices, such as the GTX 280, each SM contains only eight cores. When the host invokes a kernel grid, its blocks are enumerated and distributed to the SMs with available execution capacity. Each block runs entirety on a single SM as a unit. As thread blocks finish their execution, new blocks are launched on the vacated SMs. Threads within a block can cooperate among themselves by sharing data and synchronizing their execution. However, the number of threads comprising a block can not exceed 512, thus placing a limit on the scope of synchronization and communication in the computation. Nevertheless this limit is necessary in order leverage the GPU high core count by allowing threads to be distributed across the available cores.

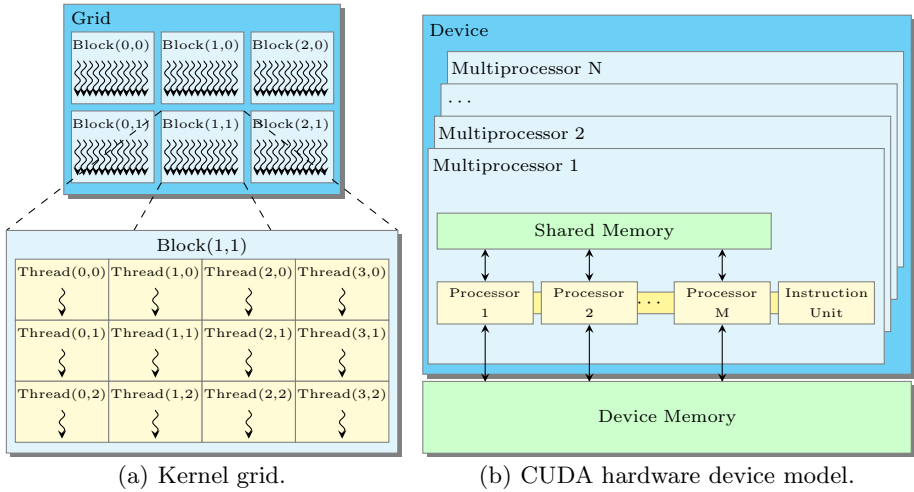


Fig. 1. Kernel grid to be executed on a CUDA device

3 Non-negative Matrix Factorization

Given a matrix $V \in \mathbb{R}_+^{n \times m}$ containing only non-negative coefficients ($V_{ij} \geq 0$) and a pre-specified positive integer, $r < \min(n, m)$, NMF finds two matrices, with non-negative coefficients, $W \in \mathbb{R}_+^{n \times r}$ and $H \in \mathbb{R}_+^{r \times m}$ whose product approximates V (as closely as possible):

$$V \approx WH. \tag{1}$$

The value of r is generally chosen to satisfy $(n + m)r < nm$, so that the approximation WH can be viewed as a compressed form of the original data [8].

The non-negativity constraints imposed to the elements of W and H are compatible with the intuitive notion of combining parts to form a whole, which is how NMF learns a parts-based representation [3]. For example, if each column of V represents a human face, then the basis elements of W , generated by NMF, can be facial features, such as eyes, noses and lips [1].

In order to measure the quality of the approximation defined in (1) it is necessary to define cost functions, between matrix V and its approximation WH . Two common metrics are the Euclidean distance, given by (2) and the (generalized) Kullback-Leibler divergence, given by (3):

$$\|V - WH\|^2 = \sum_{ij} (V_{ij} - (WH)_{ij})^2. \tag{2}$$

$$D(V \| WH) = \sum_{ij} \left(V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right). \tag{3}$$

Analogously to the Euclidean distance, the divergence is also lower bounded by zero and vanishes only when $V = WH$. However it cannot be called a “distance”, since it is not symmetric in V and WH [9]. Minimizing (2) and (3) subject to the constraints $W_{ij} \geq 0$ and $H_{ij} \geq 0$ leads to two different optimization problems that can be solved using either multiplicative or additive update rules. For the multiplicative rules, the updates specified in (4) and (5) can be used iteratively, respectively for the Euclidean distance and for the divergence:

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}, \quad W_{ia} \leftarrow W_{ia} \frac{(V H^T)_{ia}}{(W H H^T)_{ia}}. \quad (4)$$

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu} / (W H)_{i\mu}}{\sum_k W_{ka}}, \quad W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} V_{i\mu} / (W H)_{i\mu}}{\sum_v H_{av}}. \quad (5)$$

An alternative to the multiplicative update rules can be obtained by using the gradient descent technique. In such case (6) and (7) can be applied iteratively, for the Euclidean distance, until a good approximation is found:

$$H_{a\mu} \leftarrow \max(0, H_{a\mu} + \eta_{a\mu} [(W^T V)_{a\mu} - (W^T W H)_{a\mu}]), \quad \eta_{a\mu} = \frac{H_{a\mu}}{(W^T W H)_{a\mu}}, \quad (6)$$

$$W_{ia} \leftarrow \max(0, W_{ia} + \gamma_{ia} [(V H^T)_{ia} - (W H H^T)_{ia}]), \quad \gamma_{ia} = \frac{W_{ia}}{(W H H^T)_{ia}}. \quad (7)$$

Similarly (8) and (9) can be used for the divergence:

$$H_{a\mu} \leftarrow \max(0, H_{a\mu} + \eta_{a\mu} \left[\sum_i W_{ia} \frac{V_{i\mu}}{(W H)_{i\mu}} - \sum_i W_{ia} \right]), \quad \eta_{a\mu} = \frac{H_{a\mu}}{\sum_i W_{ia}}, \quad (8)$$

$$W_{ia} \leftarrow \max(0, W_{ia} + \gamma_{ia} \left[\sum_j H_{aj} \frac{V_{ij}}{(W H)_{ij}} - \sum_j H_{aj} \right]), \quad \gamma_{ia} = \frac{W_{ia}}{\sum_j H_{aj}}. \quad (9)$$

4 GPU Parallel Implementation

4.1 Euclidean Distance

The implementation of the NMF algorithm for the Euclidean distance, relies mainly on matrix multiplications, regardless of the update rules (multiplicative or additive) chosen. Our implementation uses the CUBLAS¹ library to perform matrix multiplications, due to its high-performance. However, we have developed a C++ class (*DeviceMatrix*) that not only simplifies this task, but also avoids the calculation of the transposed of a matrix. The latter is accomplished by changing the order in which the matrix is stored from column-major to row-major (or vice-versa) and reduces the amount of memory and processing required. Furthermore

¹ An implementation of BLAS (Basic Linear Algebra Subprograms) on top of CUDA.


```

-global- void UpdateMatrix_ME(float * A, float * B, float * X, int elements) {
    int idx = blockIdx.x * blockDim.x + threadIdx.x;
    if (idx < elements) X[idx] *= A[idx] / (B[idx] + SMALL_VALUE_ADD_DENOMINATOR);
}

```

Fig. 2. Kernel used to implement the NMF algorithm for the Euclidean distance

we discover that the order in which matrix multiplications are made also affects the performance of the resulting implementation (e.g. calculating $W(HH^T)$ is faster than calculating $(WH)H^T$).

For the multiplicative update rule implementation, an additional kernel, *UpdateMatrix_ME* (see Figure 2), that multiplies each element X_{ij} of matrix X by $\frac{A_{ij}}{B_{ij}}$, where A and B are matrices of the same size of X , is required (see (4)).

Likewise, the additive update rules also require an additional kernel (*UpdateMatrix_AE*) to update H and W . The complete source code of the NMF implementations was integrated in GPUMLib – a GPU Machine Learning library (<http://gpumlib.sourceforge.net/>) which aims to provide machine learning researchers and practitioners with a high performance library by taking advantage of the GPU enormous computational power [10].

4.2 Kullback-Leibler Divergence

Contrary to Euclidean distance, the divergence update rules do not depend heavily on matrix multiplications. Thus in the case of the multiplicative rules, four kernels are required (*SumW*, *SumH*, *UpdateH_MD* and *UpdateW_MD*). The *SumW* kernel calculates $\sum_k W_{ka}$ for each column a of W and puts the result in a vector of dimension r (see (5)). Similarly *SumH* calculates $\sum_v H_{av}$ for each row a of H , placing the result on a vector also with dimension r (see (5)).

The kernels *UpdateH_MD* and *UpdateW_MD* update respectively all the elements of H and W , according to (5). Both kernels work in a similar manner, thus we will focus our explanation on the *UpdateH_MD* kernel: In order to update a given element $H_{a\mu}$ we need to access all the elements in the column a of W and all elements in the column μ of V and WH , as shown in Figure 3. Thus, the CUDA thread assigned to update a given matrix element $H_{a\mu}$ needs to access the same elements of V and WH , than the threads assigned to process the elements $H_{i\mu}$ ($i \neq a$). Similarly it needs to access the same elements of W , than those required by the threads processing the elements H_{aj} ($j \neq \mu$). The organization of the threads into blocks is done in a way that allows them to share as much information as possible. This improves substantially the performance of the kernel, since accessing the shared memory is much faster than accessing the device memory. Given the amount of shared memory available per block we found that we were able to store pieces of 32×32 of W and of $V_{ij}/(WH)_{ij}$. Thus ideally our kernel should be executed in blocks of $32 \times 32 = 1024$ threads. However, CUDA imposes the maximum number of threads in a block to be 512.

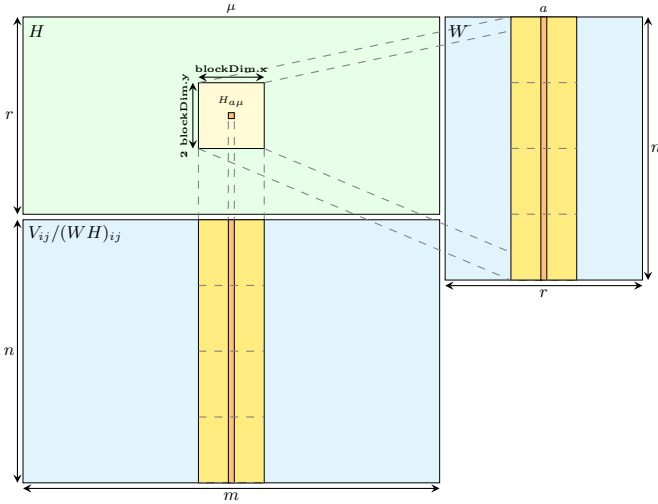


Fig. 3. Processing done, for each element $H_{a\mu}$, by the *UpdateH_MD* kernel

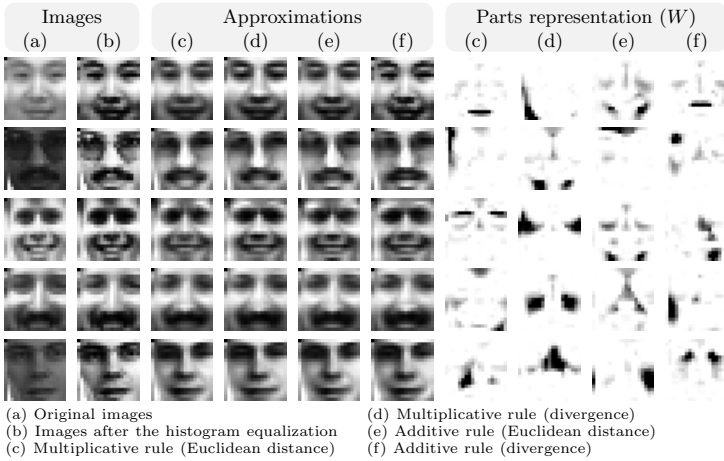


Fig. 4. Approximations and parts representation generated by the NMF algorithms

To solve this problem and maximize the amount of information shared, each block contains $32 \times 16 = 512$ ($\text{blockDim.x} = 32$, $\text{blockDim.y} = 16$) threads. Each thread gathers two elements of W , V and WH instead of one, and updates two elements of H (observe Figure 3). Therefore, although each block contains only 512 threads it will in fact update 1024 elements of H .

The additive update rules, for the divergence, require only two kernels (*UpdateH_AD* and *UpdateW_AD*) which are similar to *UpdateH_MD*.

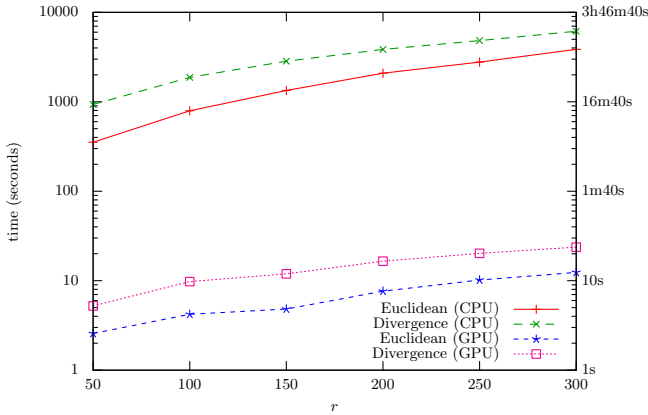


Fig. 5. Time required to run the NMF algorithms, with the multiplicative update rules

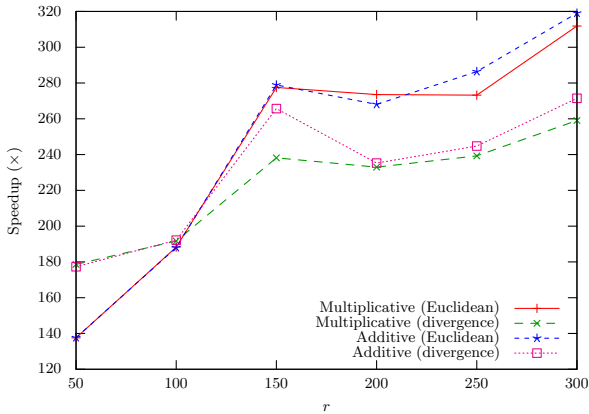


Fig. 6. Speedups provided by the GPU relative to the CPU, for the NMF algorithms

5 Experimental Results and Analysis

In order to test (and validate) the GPU implementations, we used the face database #1 from the MIT Center for Biological & Computational Learning². This database includes a total of 2429 face images of $19 \times 19 = 361$ pixels. Thus, matrix V is composed by 361 rows and 2429 columns. Before using the NMF algorithms an histogram equalization was applied to the images in order to reduce the influence of the surrounding illumination. This method improves the contrast of the images by changing its gray levels [3]. Figure 4 shows (a) five of the original face images, (b) the application of this method, (c)(d)(e)(f) the approximations generated by NMF ($r = 49$) and (c')(d')(e')(f') some of the resulting parts.

² Available at <http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html>

To determine the speedups provided by the GPU implementations relatively to the CPU, several tests were conducted, while varying the value of r . Figure 5 presents the time required to run the NMF method, using a Core 2 Quad Q 9300 2.5 GHz CPU and a GTX 280 device (containing 240 cores), with the multiplicative update rules, by performing 1000 iterations (the results for the additive rules are similar). The figure clearly shows that the GPU reduces drastically the amount of time required by the NMF algorithms. Moreover, the GPU implementations have proven to scale better when facing larger volumes of data. For example, considering the Euclidean cost function, when r is set to 50, the GPU needs approximately 2.5 seconds to run the iterations while the CPU requires approximately 6 minutes, which is translated into a speedup of $137.55\times$. However when r is to 300, the GPU now requires about 12 seconds while the CPU requires over an hour, which corresponds to a speedup of $311.86\times$. This is better emphasized in Figure 6 which exhibits the speedups provided by the GPU over the CPU.

6 Conclusion and Future Work

To cope with the staggering amount of information available, humans need help from intelligent automated systems. However, extracting useful knowledge from data is a hard task which involves separating useful information from accessory or noisy data. In this context, NMF can be used to reduce the data dimensionality, while preserving the information of the most relevant features in order to rebuild accurate approximations of the original data. However, NMF algorithms are computationally expensive, which poses an obstacle to the development of many applications. To alleviate this problem we presented a GPU parallel implementation of NMF methods, which delivers remarkable performance gains, allowing for the development of useful applications that could be disregarded otherwise, due to temporal and financial constraints. Future work will address the integration of the decomposition method with classification algorithms.

References

1. Gillis, N., Glineur, F.: Using underapproximations for sparse nonnegative matrix factorization. *Pattern Recognition* 43, 1676–1687 (2010)
2. Ribeiro, B., Silva, C., Vieira, A., Neves, J.: Extracting discriminative features using non-negative matrix factorization in financial distress data. In: Kolehmainen, M., et al. (eds.) *ICANN'09*. LNCS, vol. 5495, pp. 537–547. Springer, Heidelberg (2009)
3. Zilu, Y., Guoyi, Z.: Facial expression recognition based on NMF and SVM. *International Forum on Information Technology and Applications* 3, 612–615 (2009)
4. Steinkrau, D., Simard, P.Y., Buck, I.: Using GPUs for machine learning algorithms. In: *Proc. of the 8th Intern. Conference on Document Analysis and Recognition*, Washington, DC, USA, vol. 2, pp. 1115–1120. IEEE Computer Society, Los Alamitos (2005)
5. Catanzaro, B., Sundaram, N., Keutzer, K.: Fast support vector machine training and classification on graphics processors. In: *Proc. of the 25th Intern. Conference on Machine Learning*, vol. 307, pp. 104–111. ACM, New York (2008)

6. Schaa, D., Kaeli, D.: Exploring the multiple-GPU design space. In: IPDPS '09: Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing, Washington, DC, USA, pp. 1–12. IEEE Computer Society, Los Alamitos (2009)
7. Che, S., Boyer, M., Meng, J., Tarjan, D., Sheaffer, J.W., Skadron, K.: A performance study of general-purpose applications on graphics processors using CUDA. *J. Parallel Distrib. Comput.* 68(10), 1370–1380 (2008)
8. Xu, B., Lu, J., Huang, G.: A constrained non-negative matrix factorization in information retrieval. In: IEEE International Conference on Information Reuse and Integration, IRI 2003, pp. 273–277 (2003)
9. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS, pp. 556–562. MIT Press, Cambridge (2001)
10. Lopes, N., Ribeiro, B.: GPULib: a new library to combine machine learning algorithms with graphics processing units. In: 10th International Conference on Hybrid Intelligent Systems, Atlanta, USA (2010)

A Neighborhood-Based Clustering by Means of the Triangle Inequality

Marzena Kryszkiewicz and Piotr Lasek

Institute of Computer Science, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
{mkr,p.lasek}@ii.pw.edu.pl

Abstract. Grouping data into meaningful clusters is an important task of both artificial intelligence and data mining. An important group of clustering algorithms are density based ones that require calculation of a neighborhood of a given data point. The bottleneck for such algorithms are high dimensional data. In this paper, we propose a new TI- k -Neighborhood-Index algorithm that calculates k -neighborhoods for all points in a given data set by means the triangle inequality. We prove experimentally that the NBC (Neighborhood Based Clustering) clustering algorithm supported by our index outperforms NBC supported by such known spatial indices as VA-file and R-tree both in the case of low and high dimensional data.

1 Introduction

Grouping data into meaningful clusters is an important task of both artificial intelligence and data mining. An important group of clustering algorithms are density based ones that require calculation of a neighborhood of a given data point. The bottleneck for such algorithms are high dimensional data. In [4], we have offered a solution to this problem in the case of the DBSCAN (Density Based Clustering with NOISE) algorithm [1], which requires the calculation of a neighborhood within a given radius Eps (Eps -neighborhood) for each data point. Our solution was based on properties of the triangle property. In this paper, we examine a problem of an efficient calculation of the set of all k nearest neighboring points (k -neighborhood) for each data point. In the new task, the value of a neighborhood radius is not restricted. The theoretical solution we offer in this paper is also based on properties of the triangle inequality. Based on this solution, we offer a new TI- k -Neighborhood-Index algorithm that calculates k -neighborhoods for all points in a given data set. The usefulness of our method is verified by using it in the NBC (Neighborhood Based Clustering) clustering algorithm [7]. We prove experimentally that NBC supported by our index outperforms NBC supported by such known spatial indices as VA-file [2] and R-tree [3] both in the case of low and high dimensional data.

The paper has the following layout. Section 2 recalls the notions of an Eps -neighborhood and k -neighborhood. In Subsection 3.1, we recall a theoretical basis for calculating an Eps -neighborhood for a point within a given radius Eps , we proposed in [4]. In Subsection 3.2, we offer a theoretical basis for calculating k -neighborhood

for a point based on the triangle inequality. In Section 4, we offer the TI-k-Neighborhood-Index algorithm for calculating an index that stores k-neighborhoods for all points in a given data set. Section 5 reports the performance of the NBC algorithm depending on a used index including our proposed index. Section 6 concludes the obtained results.

2 Basic Notions

In the sequel, the distance between two points p and q will be denoted by $\text{distance}(p,q)$. Please, note that one may use a variety of distance metrics. Depending on an application, one metric may be more suitable than the other. In particular, if Euclidean distance is used, a neighborhood of a point has a spherical shape; when Manhattan distance is used, the shape is rectangular. For simplicity of the presentation, in our examples we will refer to Euclidean distance without loss of generality. Below, we recall definitions of an *Eps-neighborhood of a point* and *k-neighborhood of a point*.

Eps-neighborhood of a point p (denoted by $N_{\text{Eps}}(p)$) is defined as the set of points q in dataset D that are different from p and distant from p by no more than Eps ; that is,

$$N_{\text{Eps}}(p) = \{q \in D \mid q \neq p \wedge \text{distance}(p,q) \leq \text{Eps}\}.$$

Let p be a point in D . The set of all points in D that are different from p and closer to p than q will be denoted by $\text{CloserN}(p, q)$ that is,

$$\text{CloserN}(p, q) = \{q' \in D \mid q' \neq p \wedge \text{distance}(q',p) < \text{distance}(q,p)\}.$$

Clearly, $\text{Closer}(p, p) = \emptyset$.

The *k-neighborhood of a point* p ($\text{kNB}(p)$) is defined as the set of all points q in D such that the cardinality of the set of points different from p that are closer to p than q does not exceed $k-1$; that is,

$$\text{kNB}(p) = \{q \in D \mid q \neq p \wedge |\text{CloserN}(p, q)| \leq k-1\}.$$

Please note that for each point p , one may determine a value of parameter Eps in such a way that $N_{\text{Eps}}(p) = \text{kNB}(p)$. In particular, $N_{\text{Eps}}(p) = \text{kNB}(p)$ for $\text{Eps} = \text{distance}(q,p)$, where q is most distant neighbor of point p in $\text{kNB}(p)$.

3 Using the Triangle Inequality for Efficient Determination of Neighborhoods

3.1 Efficient Determination of Eps-Neighborhoods

Let us start with recalling the triangle inequality property:

Property 3.1.1. (Triangle inequality property). For any three points p, q, r :

$$\text{distance}(p,r) \leq \text{distance}(p,q) + \text{distance}(q,r)$$

Property 3.1.2 presents its equivalent form, which is more suitable for further considerations.

Property 3.1.2. (Triangle inequality property). For any three points p, q, r :

$$\text{distance}(p,q) \geq \text{distance}(p,r) - \text{distance}(q,r).$$

Now, we recall the results related to Eps-neighborhood we formulated in [4].

Lemma 3.1.1. Let D be a set of points. For any two points p, q in D and any point r :

$$\text{distance}(p,r) - \text{distance}(q,r) > \text{Eps} \Rightarrow q \notin N_{\text{Eps}}(p) \wedge p \notin N_{\text{Eps}}(q).$$

Proof. Let $\text{distance}(p,r) - \text{distance}(q,r) > \text{Eps}$ (*). By Property 3.1.2, $\text{distance}(p,q) \geq \text{distance}(p,r) - \text{distance}(q,r)$ (**). By (*) and (**), $\text{distance}(p,q) > \text{Eps}$, and $\text{distance}(q,p) = \text{distance}(p,q)$. Hence, $q \notin N_{\text{Eps}}(p)$ and $p \notin N_{\text{Eps}}(q)$.

By Lemma 3.1.1, if we know that the difference of distances of two points p and q to some point r is greater than Eps, we are able to conclude that $q \notin N_{\text{Eps}}(p)$ without calculating the actual distance between p and q .

Theorem 3.1.1. Let r be any point and D be a set of points ordered in a non-decreasing way with respect to their distances to r . Let p be any point in D , q_f be a point following point p in D such that $\text{distance}(q_f,r) - \text{distance}(p,r) > \text{Eps}$, and q_b be a point preceding point p in D such that $\text{distance}(p,r) - \text{distance}(q_b,r) > \text{Eps}$. Then:

- a) q_f and all points following q_f in D do not belong to $N_{\text{Eps}}(p)$.
- b) q_b and all points preceding q_b in D do not belong to $N_{\text{Eps}}(p)$.

3.2 Efficient Determination of k-Neighborhoods

Now, we will offer a theoretical basis useful for determining k -neighborhood of any point p ($k\text{NB}(p)$).

Theorem 3.2.1. Let r be any point and D be a set of points ordered in a non-decreasing way with respect to their distances to r . Let p be any point in D and Eps be a value such that $|N_{\text{Eps}}(p)| \geq k$, q_f be a point following point p in D such that $\text{distance}(q_f,r) - \text{distance}(p,r) > \text{Eps}$, and q_b be a point preceding point p in D such that $\text{distance}(p,r) - \text{distance}(q_b,r) > \text{Eps}$. Then:

- a) q_f and all points following q_f in D do not belong to $k\text{NB}(p)$.
- b) q_b and all points preceding q_b in D do not belong to $k\text{NB}(p)$.

Proof. Let r be any point and D be a set of points ordered in a non-decreasing way with respect to their distances to r .

a) Let p be any point in D , Eps be a value such that $|N_{\text{Eps}}(p)| \geq k$ (*), and q_f be a point following point p in D such that $\text{distance}(q_f,r) - \text{distance}(p,r) > \text{Eps}$. Then by Theorem 3.1.1a, q_f and all points following q_f in D do not belong to $N_{\text{Eps}}(p)$. Hence, there are at least k points different from p that are distant from p by no more than Eps (by *), and q_f and all points following q_f in D are distant from p by more than Eps. Thus, q_f and all points following q_f in D do not belong to $k\text{NB}(p)$.

b) The proof is analogous to the proof of Theorem 3.2.1a.

Example 1. Let r be a point (0,0). Figure 1 shows sample set D of two dimensional points. Table 1 illustrates the same set D ordered in a non-decreasing way with respect to the distance of its points to point r . Let us consider a determination of the

kNB of point $p = F$ for $k=3$. Let us assume that we have calculated the distances between F , and points H , G , and C , respectively, and they are as follows: $\text{distance}(F,H) = 1.64$, $\text{distance}(F,G) = 1.25$, $\text{distance}(F,C) = 1.77$. Let $\text{Eps} = \max(\text{distance}(H), \text{distance}(F,G), \text{distance}(F,C))$; i.e., $\text{Eps} = 1.77$. This means, that $F,H,G,C \in N_{\text{Eps}}(p)$. Thus, p has at least $k = 3$ points different from itself in its Eps -neighborhood. Now, we note that the first point q_f following point F in D such that $\text{distance}(q_f,r) - \text{distance}(F,r) > \text{Eps}$ is point A ($\text{distance}(A,r) - \text{distance}(F,r) = 5.8 - 3.2 = 2.6 > \text{Eps}$), and the first point q_b preceding point p in D such that $\text{distance}(F,r) - \text{distance}(q_b,r) > \text{Eps}$ is K ($\text{distance}(F,r) - \text{distance}(G,r) = 3.2 - 0.9 = 2.3 > \text{Eps}$). By Theorem 3.2.1, the points A, K as well as the points that follow A (here, point B) and precede K in D (here, no point precedes K) do not belong to $\text{kNB}(F)$.

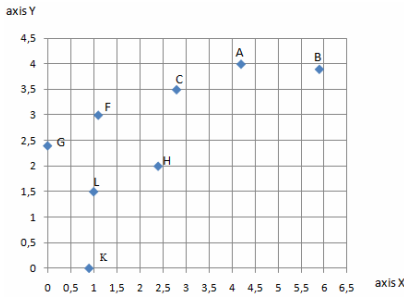


Fig. 1. Set of points D

Table 1. Ordered set of points D from Fig. 1 with their distance to reference point $r(0,0)$

$q \in D$	X	Y	$\text{distance}(q,r)$
K	0,9	0,0	0,9
L	1,0	1,5	1,8
G	0,0	2,4	2,4
H	2,4	2,0	3,1
F	1,1	3,0	3,2
C	2,8	3,5	4,5
A	4,2	4,0	5,8
B	5,9	3,9	7,1

In the sequel, a point r to which the distances of all points in D have been determined will be called a *reference point*.

4 Building k-Neighborhood Index by Using Triangle Inequality

In this section, we present the TI-k-Neighborhood-Index algorithm that uses Theorem 3.2.1 to determine k -neighborhoods for all points in a given dataset D and store them as a k -neighborhood index. The algorithm starts with calculating the distance for each point in D to a reference point r , e.g. to the point with all coordinates equal to 0. Then the points are sorted w.r.t. their distance to r . Next, for each point p in D the TI- k -Neighborhood function is called that returns $\text{kNB}(p)$. The function identifies first those k points q following and preceding point p in D for which the difference between $\text{distance}(p,r)$ and $\text{distance}(q,r)$ is smallest. These points are considered as candidates for k nearest neighbors of p . Then radius Eps is calculated as the maximum of the real distances of these points to p . It is guaranteed that real k nearest neighbors lie within this radius from point p . Then the remaining points preceding and following point p in D (starting from points closer to p in the ordered set D) are checked as potential k nearest neighbors of p until the conditions specified in Theorem 3.2.1 are fulfilled. If so, no other points in D are checked as they are guaranteed not to belong to $\text{kNB}(p)$. In order to speed up the algorithm, the value of Eps is modified each time a new candidate for a k nearest neighbor is identified.

```

Algorithm TI-k-Neighborhood-Index(set of points D, k);
/* assert: r denotes a reference point, e.g. with all coordinates = 0 */
/* assert: There are more than k points in D */
for each point p in set D do p.dist = Distance(p,r) endfor;
sort all points in D non-decreasingly wrt. attribute dist;
for each point p in the ordered set D starting from
    the first point until last point in D do
    insert (position of point p, TI-k-Neighborhood(D, p, k))
        to k-Neighborhood-Index
endfor

```

```

function TI-k-Neighborhood(D, point p, k)
b = p; f = p;
backwardSearch = PrecedingPoint(D, b);
forwardSearch = FollowingPoint(D, f);
k-Neighborhood = {}; i = 0;
Find-First-k-Candidate-Neighbours-Forward&Backward(D, p, b, f,
    backwardSearch, forwardSearch, k-Neighborhood, k, i);
Find-First-k-Candidate-Neighbours-Backward(D, p, b,
    backwardSearch, k-Neighborhood, k, i);
Find-First-k-Candidate-Neighbours-Forward(D, p, f,
    forwardSearch, k-Neighborhood, k, i);
p.Eps = max({e.dist | e ∈ k-Neighborhood});
Verify-k-Candidate-Neighbours-Backward(D, p, b, backwardSearch,
    k-Neighborhood, k);
Verify-k-Candidate-Neighbours-Forward(D, p, f, forwardSearch,
    k-Neighborhood, k);
return k-Neighborhood

```

```

function PrecedingPoint(D, var point p)
if there is a point in D preceding p then
    p = point immediately preceding p in D; backwardSearch = true
else backwardSearch = false endif
return backwardSearch

```

```

function FollowingPoint(D, var point p)
if there is a point in D following p then
    p = point immediately following p in D; forwardSearch = true
else forwardSearch = false endif
return forwardSearch

```

```

function Find-First-k-Candidate-Neighbours-Forward&Backward(D,
    var point p, var point b, var point f,
    var backwardSearch, var forwardSearch, var k-Neighborhood,
    k, var i)
while backwardSearch and forwardSearch and (i < k) do
    if p.dist - b.dist < f.dist - p.dist then
        dist = Distance(b, p); i = i + 1;
        insert element e = (position of b, dist)
            in k-Neighborhood holding it sorted wrt. e.dist;
        backwardSearch = PrecedingPoint(D, b)
    else
        dist = Distance(f, p); i = i + 1;
        insert element e = (position of b, dist)
            in k-Neighborhood holding it sorted wrt. e.dist;
        forwardSearch = FollowingPoint(D, f);
    endif
endwhile

```

```
function Find-First-k-Candidate-Neighbours-Backward(D, var point p,
var point b, var backwardSearch, var k-Neighborhood, k, var i)
```

```
while backwardSearch and (i < k) do
  dist = Distance(b, p); i = i + 1;
  insert element e = (position of b, dist)
  in k-Neighborhood holding it sorted wrt. e.dist;
  backwardSearch = PrecedingPoint(D, b)
endwhile
```

```
function Find-First-k-Candidate-Neighbours-Forward(D, var point p,
var point f, var forwardSearch, var k-Neighborhood, k, var i)
```

```
while forwardSearch and (i < k) do
  dist = Distance(f, p); i = i + 1;
  insert element e = (position of b, dist)
  in k-Neighborhood holding it sorted wrt. e.dist;
  forwardSearch = FollowingPoint(D, b)
endwhile
```

```
function Verify-k-Candidate-Neighbours-Backward(D, var point p,
var point b, var backwardSearch, var k-Neighborhood, k)
```

```
while backwardSearch and ((p.dist - b.dist) ≤ p.Eps) do
  dist = Distance(b, p);
  if dist < p.Eps then
    i = |{e ∈ k-Neighborhood | e.dist = p.Eps}|;
    if |k-Neighborhood| - i ≥ k - 1 then
      delete each element e with e.dist = p.Eps
      from k-Neighborhood;
      insert element e = (position of b, dist) in
      k-Neighborhood holding it sorted wrt. e.dist;
      p.Eps = max({e.dist | e ∈ k-Neighborhood});
    else
      insert element e = (position of b, dist) in
      k-Neighborhood holding it sorted wrt. e.dist;
    endif
  elseif dist = p.Eps
    insert element e = (position of b, dist) in
    k-Neighborhood holding it sorted wrt. e.dist
  endif
  backwardSearch = PrecedingPoint(D, b)
endwhile
```

```
function Verify-k-Candidate-Neighbours-Forward(D, var point p,
var point f, var forwardSearch, var k-Neighborhood, k)
```

```
while forwardSearch and ((f.dist - p.dist) ≤ p.Eps) do
  dist = Distance(f, p);
  if dist < p.Eps then
    i = |{e ∈ k-Neighborhood | e.dist = p.Eps}|;
    if |k-Neighborhood| - i ≥ k - 1 then
      delete each element e with e.dist = p.Eps
      from k-Neighborhood;
      insert element e = (position of b, dist) in
      k-Neighborhood holding it sorted wrt. e.dist;
      p.Eps = max({e.dist | e ∈ k-Neighborhood});
    else
      insert element e = (position of b, dist) in
      k-Neighborhood holding it sorted wrt. e.dist;
    endif
  endif
endwhile
```

```

    endif
elseif dist = p.Eps
    insert element e = (position of f, dist) in
        k-Neighborhood holding it sorted wrt. e.dist
endif
forwardSearch = FollowingPoint(D, f)
endwhile

```

5 Experimental Results

In this section, we report the run times of clustering with the NBC versions supported by k-neighborhood index (being the result of the TI-k-Neighborhood-Index algorithm), R-tree and VA-file. The versions are denoted as TI-NBC, R-tree-NBC and VA-file-NBC, respectively. The reported run times include the time of creating an index. In the experiments, we used a number of datasets (and/or their subsamples) of different cardinality and dimensionality. In particular, we used widely known datasets such as: birch [6], SEQUOIA 2000 [5], and kddcup 98 [8] as well as datasets generated automatically (random) or manually.

Table 2. Datasets used in experiments and run times (in milliseconds) of examined algorithms for $k=10$. Notation: dim. – number of dimensions, card. – number of points, N/A – results not available within at least 12 hours

No.	dataset	dim.	card.	R-tree-NBC	VA-file-NBC	TI-NBC
1	birch	2	100000	N/A	N/A	154 657
2	sequoia_2000	2	1252	1 719	1 546	78
3	sequoia_2000	2	2503	3 062	7 171	188
4	sequoia_2000	2	3910	4 750	21 360	329
5	sequoia_2000	2	5213	6 235	38 891	499
6	sequoia_2000	2	6256	7 547	63 250	655
7	sequoia_2000	2	62556	NA	NA	38 750
8	manual	2	2658	3 844	10 375	172
9	manual	2	14453	21 297	NA	2 296
10	random	3	50000	148 047	NA	65 656
11	random	5	100000	NA	NA	1 374 297
12	random	10	10000	NA	NA	44 297
13	random	10	20000	NA	NA	177 797
14	random	10	50000	NA	NA	1 421 797
15	random	20	500	2 969	NA	281
16	random	40	500	5 328	NA	500
17	random	50	10000	NA	NA	232 547
18	random	50	20000	NA	NA	1 111 531
19	covtype	54	150000	NA	NA	32 220 735
20	kddcup_98	56	56000	NA	NA	1 371 062
21	random	100	1000	NA	NA	4 641
22	random	100	10000	NA	NA	449 813
23	random	100	20000	NA	NA	1 860 921
24	random	200	500	NA	NA	2 297
25	random	200	1000	NA	NA	9 047

The run times are presented in Table 2. In several cases, it was impossible to cluster data based on R-tree or VA-file indices within 12 or more hours. In the cases, in which it was possible, TI-NBC outperformed both R-tree-NBC (up to 1 order of

magnitude) and VA-file-NBC (up to 2 orders of magnitude). In addition, TI-NBC turned out capable to cluster high dimensional data (up to a few hundreds dimensions).

6 Conclusions

In the paper, we have proposed the new method for determining k -neighborhood of a given point based on the triangle inequality. We used this method to propose a new algorithm for creating index with k -neighborhoods for all points in a data set. We proved that clustering with the NBC algorithm that uses our method is more efficient than NBC supported by such known spatial indices such as R-tree or VA-file. Unlike NBC supported by R-tree or VA-file, our method enables clustering in the case of large high dimensional datasets.

References

- [1] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Database with Noise. In: Proc. of KDD'96, pp. 226–231 (1996)
- [2] Blott, S., Weber, R.: A Simple Vector Approximation File for Similarity Search in High-dimensional Vector Spaces, Technical Report 19, ESPRIT project HERMES, vol. 9141, Technical Report number 19 (March 1997)
- [3] Guttman, A.: R-Trees: A Dynamic Index Structure For Spatial Searching. In: Proc. of ACM SIGMOD, Boston, pp. 47–57 (1984)
- [4] Kryszkiewicz, M., Lasek, P.: TI-DBSCAN: Clustering with DBSCAN by Means of the Triangle Inequality. In: Szczuka, M. (ed.) RSCTC 2010. LNCS, vol. 6086, pp. 60–69. Springer, Heidelberg (2010)
- [5] Stonebraker, M., Frew, J., Gardels, K., Meredith, J.: The SEQUOIA 2000 Storage Benchmark. In: Proc. of ACM SIGMOD, Washington, pp. 2–11 (1993)
- [6] Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: A New Data Clustering Algorithm and its Applications. *Data Mining and Knowledge Discovery* 1(2), 141–182 (1997)
- [7] Zhou, S., Zhao, Y., Guan, J., Huang, J.Z.: A Neighborhood-Based Clustering Algorithm. In: Proc. of PAKDD, pp. 361–371 (2005)
- [8] <http://kdd.ics.uci.edu/databases/kddcup98/kddcup98.html>

Selection of Structures with Grid Optimization, in Multiagent Data Warehouse

Marcin Gorawski, Sławomir Bańkowski, and Michał Gorawski

Silesian University of Technology

{Marcin.Gorawski, Slawomir.Bankowski, Michal.Gorawski}polsl.pl

Abstract. The query optimization problem in data base and data warehouse management systems is quite similar. Changes to Joins sequences, projections and selections, usage of indexes, and aggregations are all decided during the analysis of an execution schedule. The main goal of these changes is to decrease the query response time. The optimization operation is often dedicated to a single node. This paper proposes optimization to grid or cluster data warehouses / databases. Tests were conducted in a multi-agent environment, and the optimization focused not only on a single node but on the whole system as well. A new idea is proposed here with multi-criteria optimization that is based on user-given parameters. Depending on query time, result admissible errors, and the level of system usage, task results were obtained along with grid optimization.

Keywords: GRID, data warehouse, DWS, MGO, OLAP, MAS, approximate results.

1 Introduction

The architecture of a agent-based system has a grid character, which was presented in [1]. In the GRID, there is no single central control point; Software agents freely change information and data and collaborate in solving a common task. During task executions it is possible to use incomplete information about existing environment. The distribution and division algorithms are chosen in a manner that enables heuristic methods of data and tasks transfer. This approach makes the structure more elastic; however it also means that some data can be lost or the final result can be inexact. The agent-based system is oriented on a “Best Effort” approach – it assures maximal possible quality, but does not ensure full data accessibility. However obtained results are approximated using the DWS method presented in [6], and also using aggregate indices presented in [7]. Usually the obtained approximations have rather minimal errors, and simultaneous decreased query response time. This paper focuses on dynamic optimization. This means, not only creation of a local optimization schedule for query execution, but also the possibility of change in a query schedule in case of a specific data flow. This paper presents the Multi-criteria Grid Optimization (MGO). The groundbreaking multi-criteria paradigm consists of a new approach to a known query-response problem. The usual question “how much time will take to execute X

query in a G system” should be replaced with “the query X in a G system should take 3 seconds using 4 nodes in 50%, with admissible error $< 3\%$ ”.

It is not always possible for such theorem to be realized, but it is not important if it will happen in this iteration. What is more important is that software agents will try to adapt structurally to executing this certain theorem. Through adapting the connection architecture, data balancing, creation of new aggregation structures and other changes, the GRID becomes more and more adapted to executing certain theorems. The multi-criteria optimization allows definition of global parameters and user parameters. In this paper we focus on global parameters. The most represented and most used parameters are: query response time, response error, and nodes CPU usage. From the moment of the data loading process, through queries execution adequate speed-up structures are chosen, that allows to obtain expected parameters.

2 Other Works

The optimization of single nodes, data bases, and data warehouses is presented in many papers. In this paper we focus on agent-based system, which is presented in [1] and [2]. In [1] authors present the optimization of executed queries through dividing the single task into many subtasks, where each subtask is executed by certain agent. Such approach was introduced in [2], authors present a detailed language of information (about resources, both software and hardware) exchange between agents. The algorithm presented in [6] divides data warehouse using the unified probing method, using estimation the authors obtained the approximated result with limited data access. The idea of the DWS is adapted in currently implemented agent system. The OLAP processing was presented in [9], however authors used the centrally managed data warehouse [3, 4], without using the GRID structure [5].

3 The Multiagent System

The designed multiagent System (MAS) is adapted to data exchange in data warehouse. The software agent acts as a global information system about network resources. The exchange of information about accessible resources enables management of available data repositories in an efficient manner. In a agent-based distributed system it is hard to obtain coherence in a whole structure. We can only assure that the MAS will propagate the changes in the shortest time possible. Agents possess more actual data about their close neighborhood however they can have old information about more remote environment. The test performed in [10] show that this is enough for effective GRID managing. In GRID we assume that only a part of nodes possess information about most recent system state. With certain delay, information can be send through whole system, however those information can be too old to be useful. This is connected with the “Best Effort” policy. If we assume that in a time moment T_i from available N MAS nodes precisely M_i has information relevant to us. In a time moment T_{i+1} , M_{i+1} have information relevant to us. After creation of a component, information about it will be available in all system after certain time t . It should be noted that the task will be send in GRID system from one software agent to

another. However in GRID system all tasks are executed with certain error margin. It concerns not only knowledge distribution about a new data source but mostly partial data replication, partial responses and estimation connected with time boundaries of executed queries.

3.1 The AQL Language

The AQL language is a scripts language oriented on performing simple action in a agent-based environment. Using the AQL language we can, in an easy and clear manner, manage the agent-based system, without knowing the organization grid structure. This language does not contain pointers to certain system nodes, this allows us define instructions for the whole system and not only for single node. The AQL instructions are distributed as a broadcast over system, so every agent should execute received task given by client. This method is quite challenging, however it assures that while connecting any node we obtain information coherent for the whole system.

3.2 Query Execution in GRID

The system client connects to the structure, and executes the AQL queries:

- creation of a grid schema,
- creation of schema tables,
- creation of a New data source – as default data is grouped in folders, from which we retrieve them during ETL,
- creation of the ETL process that loads data from a data Source to certain schema.

Example of a AQL queries:

```
connect to 127.0.0.1:8881;
create schema sch01 on nodes ALL_NODES;
create dataset ds01 in schema sch01;
create table tab01 in dataset ds01 with columns
"meterId as ID, type as TYPE, x y as POSITION, timeId
as TIME, value as VALUE";
create datasource ds01 of type folder on nodes
ALL_NODES in folder './etl';
create etl etl01 for schema rep01 using datasource
ds01;
execute etl etl01;
```

While starting the ETL (Extraction, Transformation, Load) process first we retrieve data for source files. The data is divided into packages, each package contains from 128 up to 65536 rows and one package is assigned to one agent. In case of replication one package can be sent to several agents. Package distribution in the GRID system as default depends on agent knowledge (in which distribution takes place). The transportation of packages depends strongly on a load balancing policy - each agent chooses between several methods of data storage: data base, binary file, aggregational index.

4 Multicriterial Optimization in a GRID Environment

Each autonomous unit should collect information about its nearest neighborhood in case of efficient decision making. We cannot allow not having any information nor have incomplete information however exchange of all possible information in whole system is impossible. The amount of data traveling through system can be overwhelming and causing slowdowns or deadlocks.

4.1 Data in a Grid Environment

We can distinguish certain indicators that will appoint parameters to assure stability of executed actions. One of such indicators can be effectiveness of data loading – this sets the level of replication in system nodes. In the standard approach there are only two possibilities: data is replicated or not. In certain cases the loss of some data does not cause total loss of functionality. In case of OLTP systems shutdown of any node causes system lockdown whereas for the OLAP system such case means only partial loss of stored data but for most of the analysis it does not change its results in a significant manner. This was presented in [11].

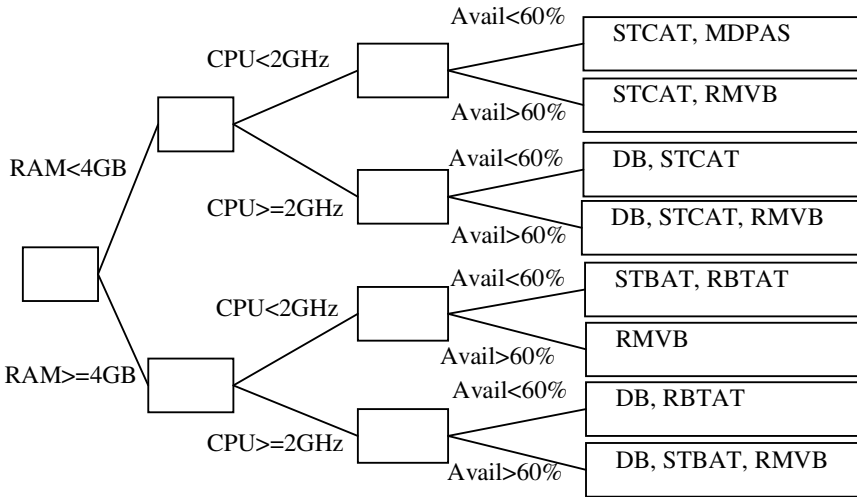


Fig. 1. Decision tree in a beginning of structures defining in agent

4.2 Package Division for a Certain Node

While loading data in the ETL process data is divided according to division policy, next packages are sent to adequate nodes. Division can be:

- random – each row gets to a random package
- round-robin – each row is assigned consecutively assigned to one of N packages,

- by certain columns – depending on columns value row is assigned to one of packages, partitions of row division by columns are created
- function – package is chosen according to function defined on data stored in row,.

The choice of rows is also connected with policy of a data replication in a agent-based system. As default this policy is chosen basing on precision indicators used in system.

The division analysis is conducted using algorithm similar to genetic (genetic witch modifications) which is strongly connected with cluster characteristics. The decision tree is created (fig.1.) as one of the methods of machine learning – the speed of a decision tree is the fastest and the best pinpoints behavior of certain rules. All rules are created and collected from computers basing on certain computer characteristics.

4.3 System Client Indicators

After package division it is assigned to certain indexing structure. Usually several structures are chosen in dependence of a cluster policy.

Available policies are dependent of (Fig.2.):

- data structures usage (γ),
- result inaccuracy indicator (Δ),
- response time indicator (T).

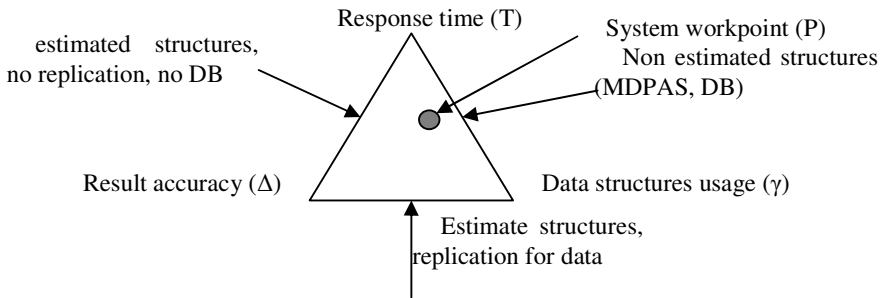


Fig. 2. Actual system workpoint for MGO in MAS

All presented indicators are expressed by a positive real numbers. Our goal is no structures usage, ideal precision (zero inaccuracy) and zero response time. In the ideal state all this indicators should be zero. Let's assume that user assign typical values, adequate for pattern queries: γ_0, Δ_0, T_0 . After execution of query A the real results for: γ_A, Δ_A, T_A are collected. The comparison of indicators shows which of them are satisfactory and which can be corrected. User can also define the severity of indicators: S_γ, S_Δ, S_T . The percentage indicators are calculated as a quotient of a real result to ideal result raised to the power of this indicator's severity (formula 1):

$$\gamma\%_A = \left(\frac{|\gamma_A|}{\gamma_0}\right)^{S_\gamma}, \Delta\%_A = \left(\frac{|\Delta_A|}{\Delta_0}\right)^{S_\Delta}, T\%_A = \left(\frac{|T_A|}{T_0}\right)^{S_T} \tag{1}$$

For each percentage indicator its value is denoted as: 0% - ideal, 0%-50% - very good, 50%-100% - good. 100% marks a boundary marked by user. 100%-200% - value that should be corrected, over 200% - critical value needed to be corrected.

We can also calculate the sum of indicators severity (formula 2):

$$S = S_{\gamma} + S_{\Delta} + S_T \quad (2)$$

While calculating end indicator for a certain query we can use formula 3:

$$P_A = \sqrt[3]{\gamma\%_A \cdot \Delta\%_A \cdot T\%_A} \quad (3)$$

The main indicator determines how far the query is remote from defined parameters (formula 4):

$$P_A = \sqrt[3]{\left(\frac{|\gamma_A|}{\gamma_0}\right)^{S_{\gamma}} \cdot \left(\frac{|\Delta_A|}{\Delta_0}\right)^{S_{\Delta}} \cdot \left(\frac{|T_A|}{T_0}\right)^{S_T}} \quad (4)$$

5 Tests

Example 1.

User defines parameters:

$$\gamma_0 = 15\%, \Delta_0 = 3\%, T_0 = 6\text{sek}$$

$$\text{Parameters severity: } S_{\gamma} = 1, S_{\Delta} = 2, S_T = 3$$

$$\text{The sum of indicators severity: } S = 1 + 2 + 3 = 6$$

Queries execution: A, B i C. After execution the A query concerned 90% data with precision 98% in time 8,5 seconds. We can assume:

$$\gamma_A = 10\%, \Delta_A = 2\%, T_A = 8,5\text{sek}; \gamma\%_A = (10,1/0,15)^1 = 0,66667$$

$$\Delta\%_A = (10,02/0,03)^2 = (0,66667)^2 = 0,44444; T\%_A = (18,5/6)^2 = (0,41)^3 = 2,8432$$

$$P_A = (0,842421)^{1/6} = 0,971825$$

After execution the B query concerned 80% data with precision 97% in time 3,5 seconds. We can assume:

$$\gamma_B = 20\%, \Delta_B = 3\%, T_B = 3,5\text{sek}; \gamma\%_B = (10,2/0,15)^1 = 1,33333$$

$$\Delta\%_B = (10,03/0,03)^2 = (1)^2 = 1; T\%_B = (13,5/6)^2 = (0,58)^3 = 0,19849$$

$$P_B = (0,842421)^{1/6} = 0,80127$$

After execution the C query concerned 85% data with precision 98,7% in time 5,5 seconds. We can assume:

$$\gamma_C = 15\%, \Delta_C = 3\%, T_C = 3,5\text{sek}; \gamma\%_C = (10,15/0,15)^1 = 1$$

$$\Delta\%_C = (10,013/0,03)^2 = (1)^2 = 0,18778; T\%_C = (15,5/6)^2 = (0,916667)^3 = 0,77025$$

$$P_C = (0,914887545)^{1/6} = 0,72451$$

Table 1. Experiments results

	% (usage)	% (accuracy)	T% (time)	P (general)
A	66,67%	44,4%	284,32%	97,1825%
B	133,3%	100%	19,849%	80,127%
C	100%	18,77%	77,025%	72,451%
Geometric mean	96,15%	43,7%	75,75%	82,63%

To go up direction we should use estimation structures and use data replication. To go down-right we should use estimation structures and do not use data replication. So, next step would be create more estimation structures like STCAT, RBTAT, RMVB.

6 Future Works

This paper presents the basics of MGO. The presented parameters enables user to actively optimize agent-based systems. In a GRID environment MGO is an addition to optimizers that exists in database engines. In GRID, the control system is needed that assures general optimization; the optimizer is distributed so there is no need for collecting data in one point. Actually only three parameters are taken into consideration, some systems may possess more parameters and this will be conducted in future works. The agent-based system now works only with Oracle database and indices: STCAT, MDPAS, RBTAT. Other integration works are planned for other data base systems and other aggregation systems.

References

1. Gorawski, M., Bańkowski, S.: Software Agents in Grid Processing (Agenci programowi w przetwarzaniu gridowym) WPP. In: Morzy, T., Rybiński, H. (eds.) I National Scientific Conference - Technology of the Data Processing (I Krajowa Konferencja Naukowa - Technologia Przetwarzania Danych - I KKN TPD), Poznań, Poland, October 28-29, pp. 118–129 (2005) ISBN 83-7143-354-9
2. Lim Choi Keung, H.N., Cao, J., Spooner, D.P., Jarvis, S.A., Nudd, G.R.: Grid Information Services using Software Agents (2003)
3. Kimball, R., Reeves, L., Ross, M., Thornthwaite, W.: The Data Warehouse Lifecycle Toolkit. John Wiley & Sons, Inc., Chichester (1998)
4. Bernardino, J., Madeira, H.: Data Warehousing and OLAP: Improving Query Performance Using Distributed Computing. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789. Springer, Heidelberg (2000)
5. Tao, Y., Papadias, D.: Range Aggregate Processing in Spatial Databases. IEEE Trans. Knowl. Data Eng. 16(12), 1555–1570 (2004)
6. Bernardino, J.R., Furtado, P.S., Madera, H.C.: Approximate Query Answering Using Data Warehouse Stripping (2003)

7. Gorawski, M., Gorawski, M., Bańkowski, S.: Nodes Indexing in Grid Telemetric Data Warehouse using MVB, STCAT and MDPAS Indexes (Indeksowanie węzłów gridowej hurtowni danych telemetrycznych o indeksach: MVB, STCAT i MDPAS). In: 4 th Conference on Information Technology (IV Konferencja Technologie Informacyjne), T1, May 21-24, vol. ZN.ETI (4), pp. 649–656. Faculty of ETI Annals in Information Technologies, Gdańsk (2006)
8. Gorawski, M., Gorawski, M.: Modified R-MVB-tree and BTV Algorithm used in a Distributed Spatio-Temporal Data Warehouse. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 199–208. Springer, Heidelberg (2008)
9. Gorawski, M., Gorawski, M., Bańkowski, S.: Interoperacyjny system przetwarzania analitycznego w trybie on-line - intOLAP. In: Hnatkowska, R.B., Warszawa, Z.H. (eds.) Żywność i oprogramowanie. Metody wytwarzania i wybrane zastosowania, pp. 209–224. Naukowe PWN, Wydaw (2008)
10. Gorawski, M., Bańkowski, S.: Grids Performance Methods (Metodyka pomiaru wydajności gridów) XIII Konferencja Sieci, Komputerowe Zakopane, 21–23 czerwiec (2006)
11. Gorawski, M., Bańkowski, S.: Tolerowanie upadków węzłów w gridowych hurtowniach danych (Tolerowanie upadków węzłów w gridowych hurtowniach danych), Systemy Czasu Rzeczywistego, Ustroń (2006)
12. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. *J. Mol. Biol.* 147, 195–197 (1981)

Approximating the Covariance Matrix of GMMs with Low-Rank Perturbations

Malik Magdon-Ismail and Jonathan T. Purnell

Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180
{magdon,purnej}@cs.rpi.edu

Abstract. Covariance matrices capture correlations that are invaluable in modeling real-life datasets. Using all d^2 elements of the covariance (in d dimensions) is costly and could result in over-fitting; and the simple diagonal approximation can be over-restrictive. We present an algorithm that improves upon the diagonal matrix by allowing a low rank perturbation. The efficiency is comparable to the diagonal approximation, yet one can capture correlations among the dimensions. We show that this method outperforms the diagonal when training GMMs on both synthetic and real-world data.

Keywords: Gaussian Mixture models; efficient; maximum likelihood; E-M.

1 Introduction

Sample covariance matrices provide important information about the probability distribution of the samples. This information can be used in pre-processing, such as PCA and whitening, where the samples can be reduced in dimensionality or decorrelated. Our focus is on fitting probabilistic models, such as the Gaussian Mixture Model. In particular, we will be looking at high dimensional models.

A major drawback of the covariance matrix is its computational cost, which grows quadratically with respect to the sample dimension. Thus, as the dimension of our data grows, the curse of dimensionality comes into effect. Although there are methods for feature selection or dimension reduction, it is not always desirable or sufficient. Leaving out samples does not help much since the computational cost grows linearly with respect to our sample size. The additional drawback of the full covariance matrix in high dimension is that the additional $O(d^2)$ parameters can lead to heavy overfitting.

Due to the computational cost associated with the full covariance matrix, an approximation of the covariance matrix is often used instead. The problem then becomes one of balancing computational cost with accuracy. A simple approximation is to use the diagonal of the covariance matrix or, in other words, use only the dimension by dimension variances. As we will explore further in Section 2, there are several proposed approximations which mostly utilize decompositions of the covariance matrix.

While the diagonal can be computed quickly, it loses all the correlation information. We propose to use the diagonal plus a low-rank perturbation. The motivation is to keep the computational cost linearly bounded to the dimension, while obtaining correlation

information; this yields a more accurate approximation by only using twice the number of parameters, d parameters for the diagonal and d for the perturbation. These extra parameters provide an improvement over the high restrictions of the diagonal. Note that the low rank perturbation does not yield a low rank covariance matrix. The perturbation is what will be low rank, but will add back most of the correlation.

Our method is tested against the diagonal approximation and the full covariance matrix for Gaussian Mixture Models. Our experiments involve various types of synthetic data as well as real world data. We examine the in-sample and out-sample average log probabilities of the models. The runtime is compared between the models to show that it is bounded linearly with respect to the sample dimension. The practicality of our method is shown by its application to real world data.

2 Covariance Matrix Approximation

We seek to address two problems. First is the computational cost of training a Gaussian Mixture Model (GMM) when using the full covariance matrix. Second is the potential to overfit due to $O(d^2)$ parameters in the full covariance matrix.

Problem Definition. The covariance is the defining characteristic of the GMM over other clustering algorithms. Typically the EM algorithm is used for training a GMM on a given dataset. The expectation step uses the inverse of the covariance matrix to calculate the probability of each sample. The maximization step updates the covariance matrix using these probabilities. The running time of a single step of this EM algorithm is $O(NKd^2)$, where N is the number of samples and K is the number of mixtures in the model. The appearance of d^2 can be prohibitive for high dimension problems, and thus one often uses an approximation to the full covariance. The ideal approximation for the full covariance is one that is not only accurate and calculated quickly, but also has an inverse that can be efficiently used to calculate sample probabilities quickly.

Since we are focusing on the GMM, our metric is the log likelihood of the GMM.

$$\log \mathbf{L} = -\frac{1}{2} \sum_{i=1}^N (x_i - \mu)^T \hat{\Sigma}^{-1} (x_i - \mu) + \frac{N}{2} \log |\hat{\Sigma}^{-1}| - \frac{Nd}{2} \log 2\pi, \quad (1)$$

where N is the number of samples, d is the sample dimension, and $\hat{\Sigma}$ is the estimate of Σ . Our discussion will focus on a single component. All our arguments extend to mixtures with multiple components. After differentiating, the log likelihood is maximized for

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad \Sigma = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T. \quad (2)$$

This can be computed in $O(Nd^2)$. The basic analytic task we address is how to efficiently choose $\hat{\Sigma}$, under a sparsity constraint, so as to do so efficiently.

Previous Work. There has been some work in covariance matrix approximations. These involve a variety of approaches, mainly decomposition [3][7], statistical estimation [4], and using assumptions on the matrix [2][1].

In [3], El Karoui looks into estimating the spectrum of large dimensional covariance matrices. The vector of eigenvalues is defined as an observation of a probability distribution. Using random matrix theory and convex optimization, this probability distribution is estimated. The resulting estimates can be used in algorithms such as PCA. In [7], decompositions are used to overcome the difficulty of estimating positive definite matrices. The Cholesky decomposition is seen as a better decomposition than variance-correlation or spectral decomposition. In [4], authors use kernel-based and parametric spectral estimation procedures to produce heteroskedasticity and autocorrelation consistent matrices. These above methods have the advantage of being applicable to most types of covariance matrices, but suffer by having quadratic, or worse, time complexities. The following approaches exploit various properties to achieve better runtimes.

In [2], an iterative method of estimation is proposed using an equivalent covariance graph and constraining the covariance matrix to be sparse. In [1], the authors estimate covariance matrices of particular distributions by two methods. First, by the EM algorithm. Secondly, by using a priori information on the distribution of the samples. With limited data, [6] finds that a mixture of the sample and common covariance matrices, along with their diagonals, can be a better approximation of the covariance matrices. In general, all these methods are $O(NKd^2)$ or cannot be efficiently inverted.

3 Approximating the Full Covariance

The diagonal approximation is the simplest. Let $\Delta x_n = x_n - \mu$ and $\Sigma = \frac{1}{N} \sum_i \Delta x_i \Delta x_i^T$; for $\hat{\Sigma} = D$, a diagonal matrix,

$$\log \mathbf{L} = -\frac{1}{2} \sum_{n=1}^N (x_n - \mu)^T D^{-1} (x_n - \mu) + \frac{N}{2} \log |D^{-1}| - \frac{Nd}{2} \log 2\pi, \quad (3)$$

which is maximized when we minimize

$$\varepsilon = \frac{1}{N} \sum_{n=1}^N \Delta x_n^T D^{-1} \Delta x_n - \log |D^{-1}| = \frac{1}{N} \sum_{n=1}^N \sum_{i=1}^d \frac{\Delta x_n(i)^2}{D_{ii}} - \sum_{i=1}^d \log \frac{1}{D_{ii}}. \quad (4)$$

We can now solve for D by setting the derivative to zero

$$\frac{\partial \varepsilon}{\partial D} = \frac{1}{N} \sum_{n=1}^N \Delta x_n^2 - D \Rightarrow D = \frac{1}{N} \sum_{n=1}^N \Delta x_n^2. \quad (5)$$

We can see that, with the diagonal constraint, the maximizer of $\log \mathbf{L}$ is the diagonal of Σ . This can be computed in $O(Nd)$, but loses all off-diagonal correlations.

3.1 Low Rank Perturbation to Diagonal

Our approximation to Σ is using a low-rank perturbation of a diagonal matrix

$$\Sigma^{-1} \approx D^2 + aa^T. \quad (6)$$

where D is our diagonal matrix and a is a d -dimensional vector that defines the low-rank perturbation. D^2 is used to ensure the approximation is positive semi-definite without having to add constraints The log-likelihood of a GMM, using this approximation is

$$\begin{aligned} \log \mathbf{L} = & -\frac{1}{2} \sum_{i=1}^N (x_i - \mu)^T (D^2 + aa^T) (x_i - \mu) \\ & + \frac{N}{2} \log |(D^2 + aa^T)| - \frac{Nd}{2} \log 2\pi . \end{aligned} \tag{7}$$

where N is the number of samples and d is the sample dimension. Maximization of this equation is equivalent to minimizing our optimization equation:

$$\varepsilon = \min \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^T (D^2 + aa^T) (x_i - \mu) - \log |(D^2 + aa^T)| . \tag{8}$$

By using an approximation based on a low-rank perturbation of a diagonal matrix, our intention is to obtain an improved accuracy over the diagonal approximation but maintain a linear bound with respect to the sample dimension. The low-rank perturbation adds another d parameters over the diagonal approximation's d parameters. Therefore, we expect the computational cost to be higher than the diagonal approximation, but still below that of the full-rank approximation which has $d(d + 1)/2$ parameters. Although this new approximation may seem nearly as restrictive as the diagonal, the addition of off-diagonal information allows for a significantly closer approximation. Further, by directly approximating the inverse of the covariance matrix, the cost of inversion is avoided. Note that this low rank perturbation implies a similar low rank perturbation expression for Σ itself. In this paper we explore a rank-1 perturbation.

In finding the optimal values for D and a , it may seem that using the diagonal of Σ for D is sufficient. However, this requires fitting aa^T to Σ with zeros along the diagonal. This pulls a away from its optimal value and towards a saddle-point at $a = 0^d$. In finding the optimal values for our parameters, D and a , we search for the point where the gradient of our optimization equation reaches zero. First, we find the gradient with respect to the α element of D , denoted D_α :

$$\begin{aligned} \frac{\partial \varepsilon}{\partial D_\alpha} = & \frac{2}{N} \sum_{n=1}^N x_{n\alpha}^T D_\alpha x_{n\alpha} - \text{trace} \left((D^2 + aa^T)^{-1} \frac{\partial (D^2 + aa^T)}{\partial D_\alpha} \right) \\ = & \sum_{n=1}^N x_{n\alpha}^T D_\alpha x_{n\alpha} - \text{trace} \left(\left((D^{-2} - \frac{D^{-2}aa^TD^{-2}}{1 + a^TD^{-2}a} \right) (2\delta_\alpha) \right) \\ = & \sum_{n=1}^N x_{n\alpha}^T D x_{n\alpha} - 2D_\alpha^{-1} + \frac{\text{trace}(2\delta_\alpha D^{-2}aa^TD^{-2})}{1 + a^TD^{-2}a} . \end{aligned} \tag{9}$$

where the derivative of the $\log |(D^2 + aa^T)|$ term is derived using Jacobi's formula $d \det(A) = \text{tr}(\text{adj}(A)dA)$; δ_α is the vector D with all elements set to zero except D_α ; and $\text{trace}()$ is the trace function. This formula can be written in vector form as:

$$\frac{\partial \varepsilon}{\partial D} = \text{diag}(\Sigma)D - 2D^{-1} + \frac{2d(D^{-2}a)^2}{1 + a^TD^{-2}a} . \tag{10}$$

Algorithm 1. Conjugate Gradient

```

1: Input: data  $x$ , diagonal matrix  $D_0$ , perturbation  $a_0$ , threshold  $T_g$ 
2:  $g_0 \leftarrow [\frac{\partial \varepsilon}{\partial a_0}, \frac{\partial \varepsilon}{\partial D_0}]$ ;  $v_0 \leftarrow -g_0$ .
3: while  $|g_i| > T_g$  do
4:   Perform a line search along  $v_i$  to find optimal step size,  $s$ .
5:    $a_{i+1} \leftarrow a_i + s \cdot v_i(1 \dots d)$ .
6:    $D_{i+1} \leftarrow D_i + s \cdot v_i(d+1 \dots 2d)$ .
7:    $g_{i+1} \leftarrow [\frac{\partial \varepsilon}{\partial a_{i+1}}, \frac{\partial \varepsilon}{\partial D_{i+1}}]$ 
8:    $\beta \leftarrow \frac{g_{i+1}^T (g_{i+1} g_i)}{g_i^T g_i}$ 
9:   Calculate new direction,  $v_i \leftarrow -g_{i+1} + \beta v_i$ .
10: end while

```

The gradient for a can be derived in a similar manner

$$\frac{\partial \varepsilon}{\partial a_\alpha} = \frac{2}{N} \sum_{n=1}^N \left(\sum_{i=1}^d x_{ni} a_i \right)^2 x_{n\alpha} - \text{trace} \left((D^2 + aa^T)^{-1} \frac{\partial (D^2 + aa^T)}{\partial a_\alpha} \right) \quad (11)$$

$$= \frac{2}{N} \sum_{n=1}^N \left(\sum_{i=1}^d x_{ni} a_i \right)^2 x_{n\alpha} - \frac{2a_\alpha}{D_\alpha} + \frac{2a_\alpha}{D_\alpha} \sum_{i=1}^d \frac{a_i^2}{D_i}. \quad (12)$$

which can be put into vector form as

$$\frac{\partial \varepsilon}{\partial a} = 2\Sigma a - 2D^{-2}a + \frac{2(D^{-2}a)(D^{-2}a)^T a}{1 + a^T D^{-2}a}. \quad (13)$$

Instead of an analytical solution, we use the conjugate gradient descent to reach an optimum [5] (See Algorithm 1). We initialize a to be a random vector, whose elements are random samples from the uniform distribution in $[0, 1]$; D is initialized as the inverse of the diagonal of Σ , which can be easily calculated in linear time. The gradient referred to as g in the algorithm is just the concatenation of the gradient for a and the gradient for D . The search direction is referred to as v . The threshold, T_g , is set to 0.001, our stopping criterion.

Although in theory we would use d steps of the conjugate gradient method, practice shows that only a few steps are needed. In the next section, we explore how our approximation compares to the diagonal approximation and the full covariance matrix.

4 Experiments

Synthetic Data. We synthesized GMM data with six clusters, randomly generated means, and randomly generated positive semi-definite covariance matrices for each mixture. An experiment consists of training a GMM on a dataset using one of the approximation methods described above (full covariance matrix, diagonal, or low rank perturbed). See Table 1 for the results, which are averaged over 100 runs.

Table 1. Probabilities for various sample sizes and dimensions

Dims.	In-Samp N=10k			In-Samp N=1k			Out-Samp N=1k		
	Diag.	Pert.	Full	Diag.	Pert.	Full	Diag.	Pert.	Full
10	-109	-101	-87	-10.9	-9.96	-8.53	-11.0	-10.2	-8.95
50	-1032	-1005	-734	-94.7	-93.2	-85.8	-95.6	-94.5	-95.6
100	-2250	-2219	-1667	-224.2	-225.6	-197.3	-225.9	-232.2	-249.1

Training Performance. We observe performance from two perspectives, the in-sample (samples used for training) and out-sample (samples used for testing) average log probabilities. Experiments were performed over various sample dimensions, dataset sizes, and number of model mixtures.

We see that the perturbed matrix probabilities are consistently closer to the full matrix probabilities than those for the diagonal matrix. Further, since this also holds in the out-sample results, we see that the perturbed matrix is not obtaining higher probabilities by overfitting the data. Note, however, that the full covariance matrix has the potential to severely overfit when d is large (see Table 1 for $d = 100$).

Runtimes. Our main motivation is to improve approximation without incurring large computational cost. Run time for the training phase is shown in Figure 1

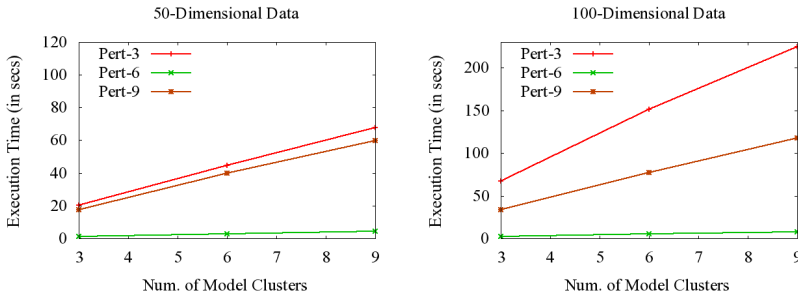


Fig. 1. Relation between runtimes and number of model clusters

In Figure 1, we compare the runtimes versus the number of clusters in our model. Here we can see that the approximation by perturbation is faster than the full rank matrix. This improvement becomes more significant with higher dimensional data and more complex models (i.e. more clusters). We see that the full rank performs almost as fast as our algorithm at low dimensions and low number of clusters. This is likely due to the overhead of the perturbation approximation, which could be overcome by further optimization. Also, it should be noted that in this case of low dimension and low model complexity, the total computational time is also very low, so any approximation would most likely fail to improve performance enough to compensate for the lack of accuracy; our algorithm is most useful when d is large.

The runtimes in Figure 2 show how the training time increases with dimension for models with various numbers of mixtures (in this case, 3, 6, and 9 mixtures). This makes it easier to see how the computational cost increases with respect to sample dimension.

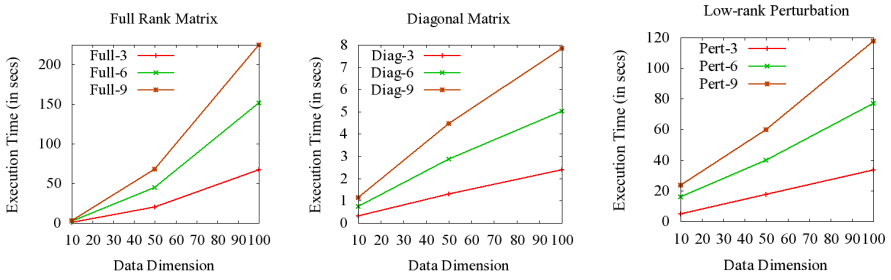


Fig. 2. Relation between runtimes and dimension

In these graphs, we can see that both the diagonal and perturbed approximations have a cost that increases linearly with respect to sample dimension; also, the full covariance matrix cost is quadratic.

4.1 Real Data

We look at GMMs for speech data. The speech comes from a TIMIT corpus of American English accents. Speakers who had grown up from various place in America were recorded while speaking 10 sentences. We look particularly at those speakers who were raised in the northern and southern regions. The data is processed so that every 25ms of speech yields a 39-dimensional sample. We then train two GMMs, one for the northern samples and another for the southern samples. Test samples are then categorized based on which GMM gives a higher log probability. Whole words are similarly classified based on which GMM gives a higher average log probability.

The out-sample results are show in the table to the right. As we saw in the synthetic data, the low rank perturbed approximation gives a better performance over the diagonal approximation. In particular, there is a significant increase with respect to the accuracy in classifying spoken words.

	Full Cov.	Diagonal	Perturbed
Sample Acc.	61.6%	49.5%	54.3%
Word Acc.	76.5%	45.9%	60.0%

Fig. 3. Accuracy of GMMs on speech samples

5 Conclusion

We have proposed a new method of approximating the covariance matrix for a Gaussian Mixture Model. Instead of simply using the diagonal, we use a low-rank perturbation of a diagonal matrix and approximate the inverse of the covariance matrix. The conjugate gradient method is used to optimize the parameters with respect to the log probability of a GMM. By approximating the inverse of the covariance matrix, we developed an algorithm that is bounded linearly with respect to sample dimension. This makes its computational cost comparable to the that of the diagonal approximation.

We compared our approximation method to the diagonal approximation and the full covariance matrix. The in-sample accuracies show that while the perturbed matrix is

not an exact approximation, it is consistently better than the diagonal. Further, the out-sample accuracy shows that this improved flexibility does not result in overfitting, since the perturbed approximation also out-performs the diagonal approximation in this respect as well. Our method has a comparable computation cost to the diagonal approximation (linear in d).

Although more complex in theory, the implementation of our method is efficient, since the conjugate gradient method has very efficient implementations. Also, this method is applicable to any sample-based covariance matrix, and not specifically in a GMMs.

In the future, we may look into lowering the absolute computational cost of the perturbed approximation by finding the optimal values and thresholds for the conjugate gradient method. Also, we plan to look into using various ranks of perturbation. Intuitively, it seems that all perturbations from rank 1 to full rank can be similarly derived and provide a greater flexibility in finding the balance between computational cost and accuracy. This would allow the approximation to be more adaptable.

Acknowledgements. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

1. Astrand, M., Mostad, P., Rudemo, M.: Improved covariance matrix estimators for weighted analysis of microarray data. *J. Comp Bio.: a Journal of Computational Molecular Cell Biology* 14(10), 1353–1367 (2007)
2. Chaudhuri, S., Drton, M., Richardson, T.S.: Estimation of a covariance matrix with zeros. *Biometrika* 94(1), 199–216 (2007)
3. El Karoui, N.: Spectrum estimation for large dimensional covariance matrices using random matrix theory. *The Annals of Statistics* 36(6), 2757–2790 (2008)
4. Haan, W.J.D., Levin, A.: A Practitioner's Guide To Robust Covariance Matrix Estimation. In: *Handbook of Statistics: Robust Inference*, vol. 15, pp. 291–341. North-Holland, Amsterdam (1997)
5. Hestenes, M., Stiefel, E.: Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards* 49(6) (1952)
6. Hoffbeck, J., Landgrebe, D.: Covariance matrix estimation and classification with limited training data. *IEEE Tran. Pattern Anal. and Mach. Intel.* 18(7), 763–767 (1996)
7. Pourahmadi, M.: Cholesky Decompositions and Estimation of A Covariance Matrix: Orthogonality of Variance Correlation Parameters. *Biometrika* 94(4), 1006–1013 (2007)

Learning Negotiation Policies Using IB3 and Bayesian Networks

Gislaine M. Nalepa, Bráulio C. Ávila,
Fabrício Enembreck, and Edson E. Scalabrin

PUCPR, Pontifical Catholic University of Paraná
PPGIA, Graduate Program on Applied Computer Science
R. Imaculada Conceição, 1155 Curitiba PR Brazil
{gislaine,avila,fabricio,scalabrin}@ppgia.pucpr.br
<http://www.ppgia.pucpr.br>

Abstract. This paper presents an intelligent offer policy in a negotiation environment, in which each agent involved learns the preferences of its opponent in order to improve its own performance. Each agent must also be able to detect drifts in the opponent's preferences so as to quickly adjust itself to their new offer policy. For this purpose, two simple learning techniques were first evaluated: (i) based on instances (IB3) and (ii) based on Bayesian Networks. Additionally, as it is known that in theory group learning produces better results than individual/single learning, the efficiency of IB3 and Bayesian classifier groups were also analyzed. Finally, each decision model was evaluated in moments of concept drift, being the drift gradual, moderate or abrupt. Results showed that both groups of classifiers were able to effectively detect drifts in the opponent's preferences.

Keywords: negotiation; instance-based learning; ensemble-based algorithm; bayesian network; drift detection.

1 Introduction

Software agents are computer programs that can replace humans in complex activities that require knowledge, experience and reasoning, such as having skills to solve conflicts through negotiation processes. A negotiation can be defined as a form of decision making involving two or more agents, which cannot make decisions independently; such agents are required to make concessions to reach an agreement [8]. Several studies [2, 4, 5, 9] were performed to improve an agent's negotiation capability aiming high performance. Those studies invest in learning algorithms that enable agents to learn the preferences of their opponents and to detect when these preferences have changed, adapting themselves automatically to them. *Drift detection* technique has proven itself useful in this scenario. It aims to quickly detect the concept drift so that fewer mistakes are committed, and so actual drifts are distinguished from noise.

This paper aims to present some techniques for *drift detection*. Therefore, an algorithm belonging to the Instance Based Learning (IBL) family [1], a Bayesian learning technique and an algorithm capable of controlling a group of learners are presented. Within this context, two systems were developed, in which the DWM (Dynamic Weighted Majority) was used along with a learning algorithm, namely: IB3 and Bayesian Network. The objective here is to compare the efficiency of the DWM-IB3 setting of Enembreck [5] with the efficiency of the DWM-BN (Bayesian Network) regarding the concept drift.

Section 2 presents the IBL family, detailing the operation of IB3. Section 3 introduces the concepts of Bayesian Networks, detailing types of construction, inference and learning. The DWM algorithm is presented in Section 4. The developed systems are in Section 5 and the experiments' results are discussed in Section 6. Finally, conclusions are presented.

2 Instance-Based Learning Algorithms

IBL is a methodology in which the technique of "Nearest Neighbour" (NN) is used to classify each new instance. Each of the variations of the IBL family came to circumvent some of the limitations of NN or an earlier version of their own. The IB1 algorithm is the simplest of the family; IB2 focuses on reducing the need for storage; and IB3 is tolerant to noisy instances. The set of training examples of an IBL algorithm expresses the concept description; such a description is determined by means of similarity functions and classification, defined by Aha [1].

IB1, IB2 and IB3 define the similarity between each training instance received and each instance stored in the Concept Description (CD). IB1 and IB2 select the most similar instance and IB3 selects the most similar acceptable instance. Aiming to overcome the noise intolerance, IB3 maintains a classification record with each instance and executes a test to determine if a given instance will be relevant to future classifications or if such proceedings represent only noise. For an instance to be good or acceptable, its accuracy intervals lower endpoint should be greater than the class frequency intervals higher endpoint. An acceptable instance will be selected randomly within the group of stored instances if no other acceptable instance is found. Only samples with a similarity greater or equal to the similarity of the most similar acceptable instance will have their performance changed, being subject to removal from the CD. IB1 stores all instances in the CD, while IB2 and IB3 store only incorrectly classified instances, aiming at the reduction of stored data. A detailed description of the algorithms can be found in Aha [1].

3 Bayesian Network

Several methods based on theory probability have been introduced, among which one of the most used is the Bayesian Network. A Bayesian Network [6] is a directed acyclic graph formed by nodes and edges. A node is used to represent

a variable, which can be discrete, continuous or propositional. The nodes are connected by links that represent the dependency between them. Each network node will have probabilistic relationships, in which the local probability value depends on the values of parent nodes.

A Bayesian Network can be constructed by using the knowledge of an expert, an automatic learning process through the analysis of previous data or a combination of both. The network construction is divided into two parts: structure learning and parameter learning. Some algorithms that allow automatic parameter learning are Counting, Expectation Maximization and Gradient Descent. After the network construction, it is possible to perform probabilistic inferences. A probabilistic inference consists of drawing conclusions as new information or evidence is known. Probabilistic inferences generate beliefs for each network node, but do not change the knowledge base. There are several methods to make probabilistic inferences, among them, JunctionTree, NodeAbsorptions and Sampling. For more details on parameter learning or methods to make probabilistic inferences, the reader is invited to check Russell [14].

Bayesian learning has proved itself efficient in several studies [7, 15], allowing agents execute their processes with better negotiating skills. However, studies [12] also suggest that groups of learners are able to achieve superior performance to that of an isolated learner. The next section deals with an algorithm capable of coordinating a group of classifiers or experts. The goal here is twofold: to increase the classification performance and to efficiently detect the concept drifts.

4 Dynamic Weighted Majority

DWM [10] is an ensemble method for concept drift that keeps a group of expert agents (base classifiers), each implementing an algorithm. The operation of the DWM is detailed below:

1. for every new instance (i) received, the DWM:
 - (a) starts weight prediction for each class with 0 (zero);
 - (b) interacts with each expert of the group, sending the training instance: (i) each expert performs its local prediction; (ii) if the learning time did not exceed ($i \bmod p = 0$), the experts which made incorrect predictions have their weight decreased. The new weight is determined by $w(j) = w(j) * \beta$, where $w(j)$ is the weight of expert j and β is a constant with value between 0 and 1, used to decrease the experts' weight; (iii) the received prediction weight of the class instance is updated based on the weight of the expert in use.
2. after all the experts have made their classifications, the class with the highest weight will be considered in the overall prediction;
3. if the learning time did not exceed ($i \bmod p = 0$):
 - (a) the weight of each expert is normalized, in which the highest weight should be 1;
 - (b) experts with a weight below a certain minimum ($w[j] < \Theta$) are deleted, where Θ is a threshold for removal experts;

- (c) if the overall prediction is incorrect, a new expert is created with weight 1.

It should be noted that the parameter p (a period of time) is required for environments with a high amount of noise.

4. Finally, the instance is sent to experts' training.

Being a generic coordination algorithm, the DWM allows the use of any learning algorithm. It is important to remember that the focus of this study is to compare the efficiency of the DWM setting with learning algorithm IB3, as demonstrated by Enembreck [5], with the efficiency of the DWM setting with Bayesian Networks. The developed environments and the experiments are detailed subsequently.

5 Test Environment

To evaluate the proposed study, software modules simulating a process of bilateral negotiation between a buyer agent and seller agent were defined. The seller agent starts the negotiation process with the publication of an interesting offer definition, and the buyer agent tries to create offers that are interesting to its opponent. Over time, the seller agent changes its interesting offer concept and the purchasing agent should automatically detect and adapt itself to this change.

Two settings were developed. First, the classification of offers and the detection of drift in the seller's concept are performed by a set of experts implementing the IB3 algorithm coordinated by DWM; afterward, the DWM coordinates a set of experts implementing Bayesian Networks. Each expert keeps an updated instance base, which represents the concept description.

The Table 1 gives an idea of the conceptual description of interesting offers for moderate drift. These concepts and the attributes used in the test environment were based on Enembreck [5], in which the reader may find details on the attributes and concepts in question.

Table 1. Conceptual Description for Moderate Drift

ID	Description
1	(Delivery Delay = very low and Amount = very little)
2	(Delivery Delay = very low and Amount = little)
...	...
8	(Delivery Delay = very low and Amount = non ensured)

The process begins by generating a set T of 50 test instances for each concept (c). These instances are used, one instance (i) at a time, to evaluate the accuracy of the system. The DWM, taking upon the role of process coordinator, verifies every test instance individually so each expert can make its own local prediction. After all experts have made their prediction for a single instance of T , the DWM chooses the class with the highest weight as the overall prediction. This process is repeated until all test instances have been classified. System performance is

calculated using Equation 1. *TP* (True Positives) and *FN* (False Negatives) are the number of interesting and non-interesting test instances correctly classified respectively and $|T|$ is the size of the test instances group.

$$accuracy(c, i) = \frac{TP + FN}{|T|} \times 100 \tag{1}$$

For each concept 50 training instances are also generated, so the weight of each expert is defined, experts with poor performance are removed, new experts are added to the group when necessary and finally, each training instance is sent to each of the expert, for training. IB3 experts perform classification using the similarity function described previously, while Bayesian Networks perform probabilistic inferences.

Bayesian Networks were built using the Netica API [3]. Each network node corresponds to a scalar variable; at the time of creation, the network received a uniform conditional probability distribution. Each new training instance corresponds to a new case, and its values are informed as findings or evidences. The probabilistic inference does not change the network’s conditional probabilities, being executed to calculate the probability of an offer being interesting given the observed values of the attributes color, price, payment, amount and deliveryDelay. The network updating activity is restricted to the moment of training. The architecture of the Bayesian network is illustrated in Fig. 1.

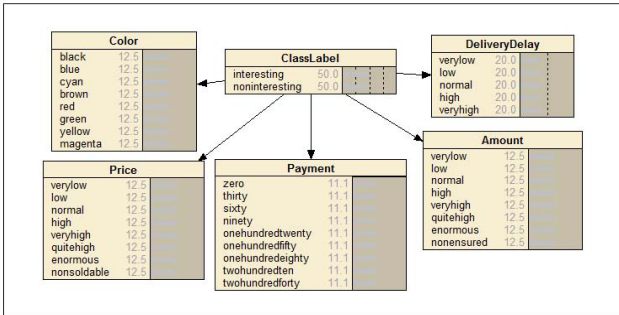


Fig. 1. Bayesian network architecture

6 Experiments

Experiments were performed using the following settings: DWM-IB3 and DWM-BN (Bayesian Network). Each setting was tested for moderate, gradual and abrupt concept drifts. The results presented in this section were generated from an average of 10 runs of each setting.

Tests carried out when the concept drift happened abruptly showed very similar results for the first two systems. Fig. 2 shows that the systems kept the lines that describe their performance very similar, 80% on average. Despite significant

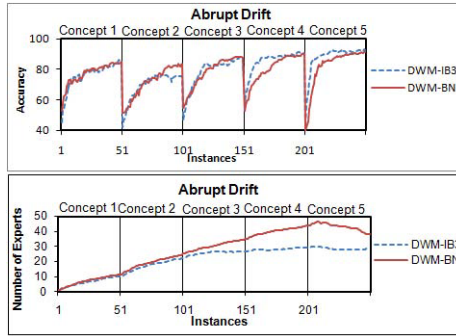


Fig. 2. Accuracy and number of experts with abrupt drift

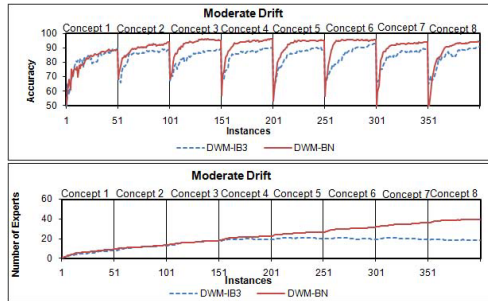


Fig. 3. Accuracy and number of experts with moderate drift

performance drops in the moments of concept drifts, both systems were able to restore performance in less than 10 iterations when the number of experts tended to stabilize or decrease (Concept 4 and Concept 5 for IB3 and Concept 5 to BN). A remarkable feature is the greater number of experts generated by the setting DWM-BN, as presented in Fig. 2.

Results obtained from data with moderate drift presented the performance of DWM-BN standing out slightly on the performance of DWM-IB3. Despite showing a higher performance drop, the DWM-BN could recover faster than the DWM-IB3. In this case the IB3 reacted more smoothly than it did when changes were abrupt, once a reasonable performance was reached, the stored instances presented good classification accuracy and were not modified so frequently. Fig. 3 shows that, same as with the previous setting, DWM-BN showed a larger amount of experts, while IB3 showed a tendency to stabilize itself.

Fig. 4 shows that during a gradual concept drift both DWM-IB3 and DWM-BN maintained an average performance of 80%. The performance drop in both settings was very similar and much less than those identified in Fig. 2 and Fig. 3, due the similarities among the produced concepts. The number of experts

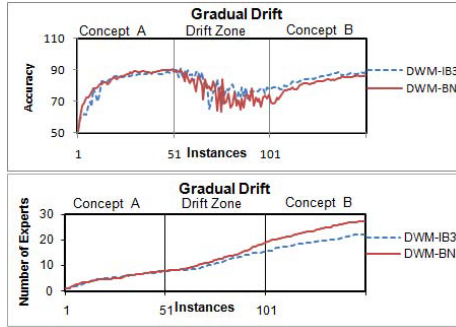


Fig. 4. Accuracy and number of experts with gradual drift

presented in Fig. 4 showed continued growth for the amount of generated data in both systems, tending to the beginning of stabilization in concept B.

The environments less affected by drifts and which managed to maintain a good performance with less sharp falls were the ones in which the concept drifted gradually and moderately. Moderate and gradual drifts also generated the smallest number of experts, once they are, theoretically, the easiest type of drifts to learn. The stabilization of the number of experts is more intense for the configuration DWM-IB3, because ensemble-based algorithms use the diversity of classifiers to cover different regions of the tuple space; when enough region is covered, some classifiers become useless, once they represent regions already covered by other classifiers. The DWM-BN setting allowed the presence of most experts throughout the process. This behavior justifies the higher number of experts kept by DWM-BN, and, therefore, the need for more storage capacity and an execution time 60% higher on average than the other system.

7 Conclusions

The development of this work has allowed an efficient analysis of the IB3 and Bayesian Networks as learning algorithms coordinated by DWM in a scenario of automated negotiation. Three algorithms from the IBL family were presented (IB1, IB2 and IB3). The Bayesian Networks were incorporated into the study due to their high adaptability, as they allow new information to generate drifts in dependencies between concepts and also in the concepts themselves.

Aiming to increase the classification performance and efficiently detect concept drifts, the coordination model DWM was used in the experiments, contributing to increase the performance of learning algorithms. The experiments revolved around the settings DWM-IB3 and DWM-BN. In general, the performances of both systems were similar, detecting drifts quickly, adapting themselves by changing their concept description and recovering their performances in a few iterations. Most of the time, both systems maintained similar performance lines, proving that both algorithms present a satisfactory performance

in detecting drift when coordinated by DWM; however the setting DWM-BN requires massive memory resources and high processing time due to the need to update each network with the classified instance, while the updating is a low cost process for the setting DWM-IB3. These results strongly encourage the use of setting DWM-IB3 for the detection of different types of concept drifts in bilateral and multi-issue negotiations.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based Learning Algorithms. *Machine Learning* 6(1), 37–66 (1991)
2. Coehoorn, R.M., Jennings, N.R.: Learning an Opponent's Preferences to Make Effective Multi-Issue Negotiation Trade-Offs. In: *Proc. of the Sixth International Conference on Electronic Commerce*, pp. 59–68 (2004)
3. Norsys Software Corp., Netica-j manual (2009), <http://www.norsys.com/neticaj/docs/netica#jman.pdf>
4. Enembreck, F., Avila, B.C., Scalabrin, E.E., Barthès, J.-P.: Drifting Negotiations. *Applied Artificial Intelligence* 21(9), 861–881 (2007)
5. Enembreck, F., Tacla, C.A., Barthès, J.P.: Learning Negotiation Policies Using Ensemble-Based Drift Detection Techniques. *International Journal of Artificial Intelligence Tools* 18(2), 173–196 (2008)
6. Pearl, J.: Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann, San Mateo (1988)
7. Hindriks, K., Tykhonov, D.: Opponent modelling in automated multi-issue negotiation using bayesian learning. In: *AAMAS'08: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 331–338 (2008)
8. Kersten, G.E., Michalowski, W., Szpakowicz, S., Koperczak, Z.: Restructurable representations of negotiation. *Manage. Sc.* 37(10), 1269–1290 (1991)
9. Klinkenberg, R., Renz, I.: Adaptive Information Filtering: Learning in the Presence of Concept Drifts. In: *ICML-98*, pp. 33–40 (1998)
10. Kolter, J.Z., Maloof, M.A.: Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift, p. 123. *IEEE Computer Society Press*, Los Alamitos (2003)
11. Littlestone, N., Warmuth, M.: The Weighted Majority algorithm. *Information and Computation* 108, 212–261 (1994)
12. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research* 11, 169–198 (1999)
13. Ruggeri, F., Faltin, F., Kenett, R.: Bayesian Networks. *Encyclopedia of Statistics in Quality & Reliability*. Wiley & Sons, Chichester (2007)
14. Russell, S., Norving, P.: *Artificial Intelligence: A Modern Approach*, 2nd edn. Prentice Hall, Englewood Cliffs (2004)
15. Zeng, D., Sycara, K.: Bayesian learning in negotiation. *International Journal of Human-Computer Studies* 48(1), 125–141 (1998)

Trajectory Based Behavior Analysis for User Verification^{*}

Hsing-Kuo Pao¹, Hong-Yi Lin¹, Kuan-Ta Chen², and Junaidillah Fadlil¹

¹ Dept. of Computer Science & Information Engineering,
National Taiwan University of Science & Technology, Taipei, Taiwan
{pao,M9615061}@mail.ntust.edu.tw, nedijf@gmail.com

² Institute of Information Science, Academia Sinica, Taipei 115, Taiwan
ktchen@iis.sinica.edu.tw

Abstract. Many of our activities on computer need a verification step for authorized access. The goal of verification is to tell apart the true account owner from intruders. We propose a general approach for user verification based on user trajectory inputs. The approach is labor-free for users and is likely to avoid the possible copy or simulation from other non-authorized users or even automatic programs like bots. Our study focuses on finding the hidden patterns embedded in the trajectories produced by account users. We employ a Markov chain model with Gaussian distribution in its transitions to describe the behavior in the trajectory. To distinguish between two trajectories, we propose a novel dissimilarity measure combined with a manifold learnt tuning for catching the pairwise relationship. Based on the pairwise relationship, we plug-in any effective classification or clustering methods for the detection of unauthorized access. The method can also be applied for the task of recognition, predicting the trajectory type without pre-defined identity. Given a trajectory input, the results show that the proposed method can accurately verify the user identity, or suggest whom owns the trajectory if the input identity is not provided.

Keywords: Verification, Behavior analysis, Account security, Trajectory, Dissimilarity measure, Manifold learning, Isomap.

1 Introduction

With the network grows fast, people rely on Internet for daily activities. Someone uses the Internet to do bank transactions, talk with friends in the electronic community, search interesting information, play on-line games, and so on. Many of the activities do not allow anonymous access. The usual login-on process is to provide password or even biometrics like fingerprint, face, iris for personal identification. In this case, the users from both sides of network face the problem that the personal information may be robbed by some unauthorized person such

^{*} Research partially supported by Taiwan National Science Council Grants # 98-2221-E-011-105 and # 98-2221-E-001-017.

as the cracker breaking-in others' on-line game accounts for illegal benefits. Consequently, *verification* plays an important role in *account security*. In this work, we propose a verification scheme based on *user trajectories* to check whether or not the person is the true account owner.

Two-factor authorization is a mechanism that is used to authorize or verify the ownership of an account. Having two factors at the same time to verify identity, the account security is enhanced. For an example, the first factor is that users have to slide the smart card into ATM, and the second factor is for users to type their passwords. Nevertheless, those factors can possibly be faked or stolen by hackers. We suggest analyzing the behavior of users' trajectories so that different patterns from two users can be recognized. Although hackers can design some illegal programs which learn decisions of humans, the behavior patterns are generally difficult to simulate because it is an AI-hard problem. In addition, based on the proposed method, we try not to bring users any extra load to assist the authorization methods to improve the authorization power. That is to say, the users do not need to adjust their ways of account usages. We define our problem as follows:

Definition 1 (Verification). *Given a coordinate trace, verification is to match between a trace and a pre-defined identity, a Yes/No question.*

Proposed Method. As shown in Figure 1, Given an input of coordinate set, we want to extract hidden patterns. We consider a trace $\mathbf{s} = (\mathbf{x}_1, \dots, \mathbf{x}_t, \dots, \mathbf{x}_T)$ which has T coordinates in a 2-D or 3-D space where T is the length of the trace. Working on a trace of short length implies that the method is effective even with such a limited-length input. Given the coordinate information, we want to extract features from the trace and compute a dissimilarity measure for each pair of traces. Afterwards, we combine the pairwise dissimilarities with a manifold learning approach called *Isomap (Isometric feature mapping [1])* for trajectory representation and use *Smooth SVM [2]* in the representation space to classify the patterns into the true user or an intruder. Various trajectory data will be investigated using our method.

2 Related Work

In this section, we discuss some issues which are relevant to our research. There are two topics: account verification and trajectory analysis.

Verification. We need to restrict the account access only for the true account owner. To identify who the user is or at least verify whether the user is the account owner, we discuss several approaches.

Signature. Signature verification is a well known method to verify whether a user is the account owner. The signature verification can be roughly divided into two categories: on-line signature verification [3] and off-line signature verification [4]. In on-line verification, one example is the handwriting trajectory, measured by

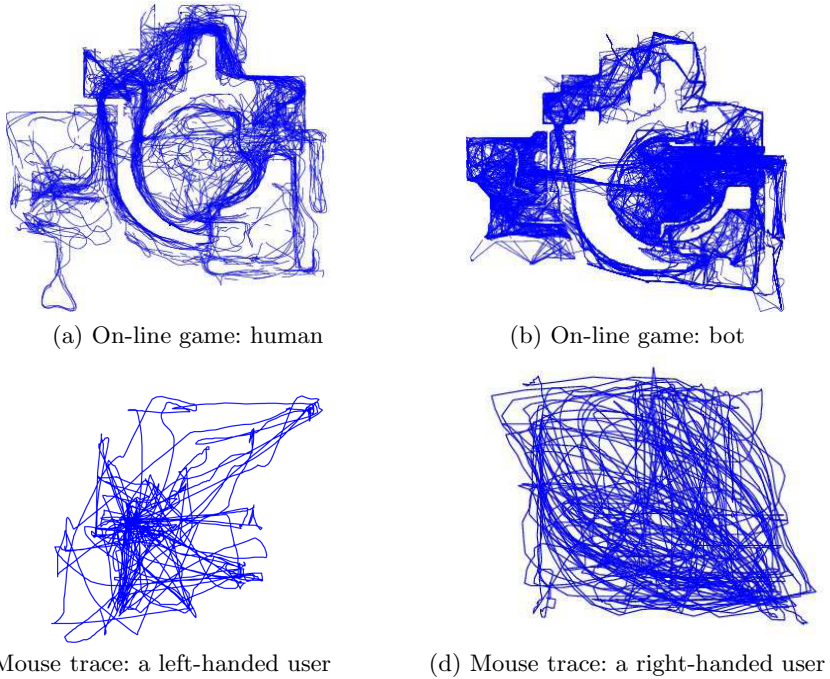


Fig. 1. The different kinds of trajectory input. The (a) and (b) are the traces of bot and human respectively from an on-line game called Quake 2. The (c) and (d) are mouse traces from two individuals. We can vaguely recognize that (c) is from a left-handed user and (d) is from a right-handed user.

time. Mario et al. [5] used a camera to collect such handwriting traces. Richiardi et al. [6] employed the *Gaussian Mixture Models* to verify on-line signatures. They used Gaussian components to represent the features of handwriting, and used the MDL (*Minimum Description Length*) principle to automatically select the signature model. In off-line signature verification, the input is usually a 2-D signature image captured by scanner. The problem therefore becomes an image recognition problem. Usually, we believe that the on-line version rather than the off-line version extracts more information because the time stamps are included in the input.

CAPTCHA. For bot detection, the CAPTCHA (*Completely Automated Public Test to tell Computers and Humans Apart*) [7] is a test that automatically asks users some problems to judge if the user is a human or a bot (automated program). The problem may be a randomly generated string drawn on an image. That is easy for a human, but may be difficult for a bot to tell what the string is. The method is effective, even is still able to be cracked; however, answering the trivial questions can be annoying for human users.

Trajectory Analysis. The pattern recognition for sequential or trajectory data is a wide area of research. We discuss only the ones most related to our work. There are mainly two types of sequential data. One holds temporal relation such as handwriting traces, mouse traces, or avatar traces from on-line games. The other type may not hold the temporal relation, for instance, biological sequences or language texts. For handling traces, the SAX (*Symbolic Aggregate approX-imation*) [8] is a popular method, which has been successfully applied to many applications. One of the key steps of SAX is to discretize the numerical values of input to produce a set of symbols which is an approximation of the original input. Jae-Gil et al. [9] combined the region-based and trajectory-based clustering methods to classify trajectories. They used the MDL principle to partition trajectories, and then searched for specific patterns. Keogh et al. [10] studied parameter-free description of sequential data. Pao et al. [11] considered the distance function between biological sequences. Both followed the study of Li et al. [12], who tried to use the Kolmogorov complexity [13] to describe the “irregularities” in sequential data. Other than the above works, Chen et al. [14], [15] proposed using trajectory inputs for game bot detection.

3 Framework

In this section, we introduce our framework and discuss how we deal with the verification problem based on trajectories. The organization of our framework is as follows. The first step is to extract useful features from trajectories. The second step measures dissimilarities between a pair of sequences. Based on the dissimilarity measure, the last step is to refine the dissimilarities by a manifold learning method called Isomap [1] and use a classification method to classify trajectories into normal users or intruders. We adopt Smooth SVM [2] as the classifier.

3.1 Feature Extraction

Given a trajectory $\mathbf{s} = (\mathbf{x}_1, \dots, \mathbf{x}_t, \dots, \mathbf{x}_T)$ of T seconds, we extract useful features from the trajectory. There are two different features in our study. First, a step is a vector $\mathbf{x}_{t+1} - \mathbf{x}_t$, so Euclidean step size is $\lambda = \|\mathbf{x}_{t+1} - \mathbf{x}_t\|$. Second, an angle θ_t is defined to be the angle between the vector $\mathbf{x}_{t+1} - \mathbf{x}_t$ and the x -axis.

3.2 Dissimilarity Measures

We employ the Markov chain model for dissimilarity measurement. Let $\mathcal{M}(\sigma_\lambda, \sigma_\theta)$ denote the associated model of a trajectory sequence (probably based on the *maximum likelihood* principle), where σ_λ and σ_θ are the transition parameters. The σ_λ describes the standard deviation of step size λ_{t+1} , assumed centered in λ_t ; and the σ_θ describes the standard deviation of angle θ_{t+1} , assumed centered in θ_t . That is,

based on the Markovian properties, between two coordinates in consecutive time stamps \mathbf{x}_t , \mathbf{x}_{t+1} , we assume that

$$P(\lambda_{t+1}|\lambda_t) \sim N(\lambda_t, \sigma_\lambda^2) = \frac{1}{\sqrt{2\pi}\sigma_\lambda} \exp\left(-\frac{(\lambda_{t+1} - \lambda_t)^2}{2\sigma_\lambda^2}\right), \quad (1)$$

$$P(\theta_{t+1}|\theta_t) \sim N(\theta_t, \sigma_\theta^2) = \frac{1}{\sqrt{2\pi}\sigma_\theta} \exp\left(-\frac{(\theta_{t+1} - \theta_t)^2}{2\sigma_\theta^2}\right). \quad (2)$$

Given a model \mathcal{M} , the log-likelihood $\ell(\mathbf{s}; \mathcal{M})$ of a trajectory \mathbf{s} can be written as

$$\ell(\mathbf{s}; \mathcal{M}) = \log L(\mathbf{s}; \mathcal{M}) = \log P(\mathbf{x}_1) + \sum_{t=1} \log \left(P(\mathbf{x}_{t+1}|\mathbf{x}_t) \right), \quad (3)$$

where L is the likelihood function. In our design, the dissimilarity or distance between two trajectories depends on how well a trajectory is described by the model for the other trajectory. First, given the model \mathcal{M} we compute the code length of a trajectory \mathbf{s} as a negative logarithm of the likelihood, such as

$$c(\mathbf{s}|\mathcal{M}) = -\ell(\mathbf{s}; \mathcal{M}) = -\log L(\mathbf{s}; \mathcal{M}). \quad (4)$$

Note that \mathcal{M} does not have to be the associated model of the trajectory \mathbf{s} . We define the dissimilarity between two trajectories \mathbf{s}_1 and \mathbf{s}_2 as

$$d(\mathbf{s}_1, \mathbf{s}_2) = \frac{c(\mathbf{s}_1|\mathcal{M}_2) + c(\mathbf{s}_2|\mathcal{M}_1)}{c(\mathbf{s}_{12}|\mathcal{M}_{12})}, \quad (5)$$

where \mathbf{s}_{12} is the trace concatenating \mathbf{s}_1 and \mathbf{s}_2 one after another, and \mathcal{M}_{12} is the associated model of \mathbf{s}_{12} .

3.3 Trajectory Representation and Labeling

In principle, given the pairwise dissimilarities for trajectories as in Eq. 5, we can simply adopt a simple method such as k nearest neighbors to classify trajectories to be one belonging to the true account owner or one belonging to an intruder. However, we aim at designing a more effective method. We seek an embedding feature space to represent a set of trajectories/sequences; and in the space, we adopt an SVM classifier to label the sequences. In the feature space, two sequences are close (similar) to each other if (1) they have small measure of Eq. 5 or, (2) both of them are close (similar) to a third sequence. The second condition implies that two sequences \mathbf{s}_1 and \mathbf{s}_2 are similar, if they are both similar to some other sequence. In order to achieve the goal, we apply Isomap [1] to find the feature space, that is, the representation of the trajectories. In Isomap, we 1) construct a neighborhood graph by linking each pair of sequences/points that qualify as neighbors; 2) find the length of the shortest path between each pair of points and take it as the approximation of their geodesic distance; and 3) take the pairwise (geodesic) distances as the input and apply Multidimensional Scaling (or MDS) to find the global Euclidean coordinates of the points. Figure 2

Table 1. Data statistics and parameters. The k_{Iso} is the k used in constructing the neighborhood graph for Isomap. For simplicity and a fair comparison, the intrinsic dimensionality is chosen to be 5 in all datasets.

Name	Users	Instances	Trace Length	k_{Iso}	Intrinsic Dim.
Handwriting	11	110	702	7	5
Mouse	8	178	16665	6	5
Game	94	940	1000	8	5

Table 2. (a) The summary of the verification results on various inputs, and (b) the recognition result on the Game dataset, with different lengths of inputs. The table shows the average error rates, in percentage by SSVM classification, with three times of ten-fold cross-validation. In (b), the dataset includes four classes: human, and three types of bots: CR Bot, Eraser Bot, and ICE Bot.

Data Set	Training Error	Test Error	Trace Length	Training Error	Test Error
Handwriting	1.47	2.89	500 seconds	5.48	7.97
Mouse	6.82	8.79	1000 seconds	2.15	2.83
Game	7.59	14.34			

(a)

(b)

shows an example of the 2-D plot after the process of Isomap. The “optimal” dimensionality (called *intrinsic dimensionality*) for separating different kinds of trajectories can be estimated by finding the “elbow” point in the residual variance curve [1]. In the feature space, ideally, we can adopt any classifier to verify a trajectory to be one from the true account owner or one from others. In this study, we use SSVM [2] to evaluate the performance of proposed method.

4 Experiment

Data Description Our experiment datasets are real data including handwriting, mouse traces, and the traces from on-line games. The handwriting dataset (UJI Pen Characters [4]) collects handwriting of digits, lowercase and uppercase letters from 11 humans. We created the mouse movement dataset based on eight users’ daily mouse controlling traces, a total of 178 instances. The game trace data is to collect data from a popular on-line game called Quake 2, a famous FPS (First-Person Shooter) game developed by id Software [5], a trace data lasting about 143.8 hours in total. The dataset consists of some traces from human users, and some others from well-known game bots (automatic programs) such as CR Bot, Eraser Bot and ICE Bot [3]. In summary, the data statistics as well as the parameters used in this work are listed in Table 1.

¹ <http://archive.ics.uci.edu/ml/datasets/UJI+Pen+Characters>

² <http://www.idsoftware.com/>

³ More details can be found in [14], [15].

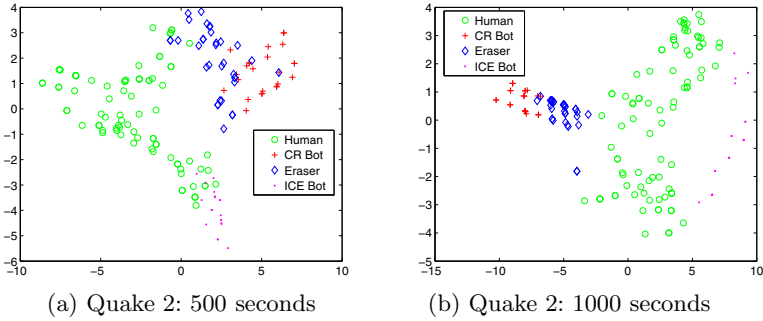


Fig. 2. Given different length of input, the representation of the Quake 2 traces after the projection by Isomap into a 2-D space, where a point represents a trace of a human user (green circle) or from a bot (others). The x - and y -axes are the first and second principal coordinates from Isomap. Classification is worked on a higher dimensional space, known as the space of *intrinsic dimensionality*, shown in Table 1.

We want to verify if a trajectory belongs to the true account owner. We take turns to investigate trajectories from all pairwise individuals. In other words, if there are n users in the dataset, we do C_2^n tests. Table 2(a) shows the performance of proposed method in different kinds of trajectories. Among them, the handwriting dataset gives us the best accuracy (97.11%), followed by the accuracy for mouse trace verification (91.21%), then the accuracy of game traces (85.66%). It follows our intuition. The handwriting traces give us better discriminative power, than mouse traces. On the other hand, game traces are usually in a restricted environment, therefore lack some degree of freedom to show the true identity of trace users.

To further demonstrate the power of proposed method, we use our method to extract different patterns from human traces and three types of bot traces, a multi-class classification problem. The result is shown in Table 2(b). When the trace is 500-second long, our method can reach 92.03% accuracy. If we continue to collect the data up to 1000-second long, the accuracy can further be improved to 97.17%. In Figure 2, we can visualize that the trace has a better presentation when a longer trace is collected. Obviously, we prefer a method that can identify the user behavior as fast as possible, before significant amount of account information or account value got stolen by crackers.

5 Conclusion

In this work, we proposed a novel method for user trajectory verification. The verification scheme is based on a dissimilarity measure, followed by a manifold learning adjustment from Isomap. We have applied our method to various types of trajectories including handwriting, mouse traces, and game traces. The results show that our method is effective in picking the hidden patterns embedded in the trajectory and is plausible to solve related problems, such as recognizing

who the user is without given the account identity. Thus, we believe that the proposed method merits further investigation by the account verification and related research communities.

References

1. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323 (2000)
2. Lee, Y.J., Mangasarian, O.L.: SSVM: A smooth support vector machine for classification. *Comput. Optim. Appl.* 20, 5–22 (2001)
3. Jain, A.K., Griess, F.D., Connell, S.D.: On-line signature verification. *Pattern Recognition* 35, 2963–2972 (2002)
4. Qiao, Y., Liu, J., Tang, X.: Offline signature verification using online handwriting registration. In: *CVPR* (2007)
5. Munich, M.E., Perona, P.: Visual identification by signature tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 200–217 (2003)
6. Richiardi, J., Drygajlo, A.: Gaussian mixture models for on-line signature verification. In: *WBMA '03: Proceedings of the 2003 ACM SIGMM workshop on Biometrics Methods and Applications*, pp. 115–122. ACM, New York (2003)
7. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: Using hard AI problems for security. In: *EUROCRYPT*, pp. 294–311 (2003)
8. Lin, J., Keogh, E.J., Lonardi, S., Chi Chiu, B.Y.: A symbolic representation of time series, with implications for streaming algorithms. In: *DMKD*, pp. 2–11 (2003)
9. Lee, J.G., Han, J., Li, X., Gonzalez, H.: *TraClass*: trajectory classification using hierarchical region-based and trajectory-based clustering. In: *PVLDB*, vol. 1, pp. 1081–1094 (2008)
10. Keogh, E., Lonardi, S., Ratanamahatana, C.A.: Towards parameter-free data mining. In: *KDD '04: Proceedings of the Tenth ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining*, pp. 206–215. ACM, New York (2004)
11. Pao, H.K., Case, J.: Computing entropy for ortholog detection. In: *International Conference on Computational Intelligence*, pp. 89–92 (2004)
12. Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P., Zhang, H.: An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics* 17, 149–154 (2001)
13. Li, M., Vitányi, P.: *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd edn. Springer, New York (1997)
14. Chen, K.T., Liao, A., Pao, H.K.K., Chu, H.H.: Game bot detection based on avatar trajectory. In: *Proceedings of IFIP ICEC 2008* (2008)
15. Chen, K.T., Pao, H.K.K., Chang, H.C.: Game bot identification based on manifold learning. In: *Proceedings of ACM NetGames 2008* (2008)

Discovering Concept Mappings by Similarity Propagation among Substructures

Qi H. Pan, Fedja Hadzic, and Tharam S. Dillon

Curtin University of Technology, Perth, Australia
{Helen.Pan, Fedja.Hadzic, Tharam.Dillon}@curtin.edu.au

Abstract. Concept matching is important when heterogeneous data sources are to be merged for the purpose of knowledge sharing. It has many useful applications in areas such as schema matching, ontology matching, scientific knowledge management, e-commerce, enterprise application integration, etc. With the desire of knowledge sharing and reuse in these fields, merging commonly occurs among different organizations where the knowledge describing the same domain is to be matched. Due to the different naming conventions, granularity and the use of concepts in different contexts, a semantic approach to this problem is preferred in comparison to syntactic approach that performs matches based upon the labels only. We propose a concept matching method that initially does not consider labels when forming candidate matches, but rather utilizes structural information to take the context into account and detect complex matches. Real world knowledge representations (schemas) are used to evaluate the method.

Keywords: concept matching, schema matching, tree mining.

1 Introduction

Increasingly, researches are focusing on data integration which is an essential step in merging heterogeneous data sources because of the needs of merging companies, transforming and integrating data originating from one or multiple legacy applications or information systems into a new one. Semantic heterogeneity is a crucial problem in any data sharing system. There are various tasks involved within the data integration process, such as data transformation, and schema/semantic matching. This paper focuses on the schema concept matching problem and proposes a new method that utilizes structural information to detect the candidate concept mappings which are validated using string similarity measures and an online dictionary. XML subtree mining techniques and semantic similarity measurement are combined to provide a better schema mapping result according to the structural and linguistic matching. The method does not require any user interaction and performs the discovery of the mappings and their validation in a fully automated way. Related work is briefed in the following section. In Section 2, problem definition is provided and the motivation behind the proposed approach is discussed with the aid of an illustrative example. Section 3 provides a high level description of the proposed method. This is followed

by a detailed description of the steps involved in Section 4. Section 5 presents a number of experiments using real world XML schemas which demonstrates the effectiveness of the method. The paper is concluded in Section 6.

1.1 Related Work

In the matching of heterogeneous knowledge representations, the main challenge is that of finding semantically correct matches among the concepts. This problem is analogous to schema matching in databases. Semantic matching takes the schema information as well as the positions of nodes in the conceptual models (graph or tree) into account. The TreeMatch algorithm[1] computes the similarity of contexts in which the two concepts occur in the two schemas. It utilizes schema information and the representative tree structure. A similarity flooding algorithm[2] produces a similarity mapping between the concepts of two graph structures. A string match operator is used to obtain the initial matching concept nodes which propagate the similarity to their adjacent nodes. The Anchor-PROMPT algorithm [3] takes as input a set of similar terms (anchors) and determines the sets of other related terms by analyzing the paths in the subgraph limited by the anchor points. It is based on the notion that if two pairs of concepts in the source ontologies are similar, and there are paths connecting those two concepts, then the concepts in those paths are often similar as well. [4] perform element and structure-level semantic matching among the elements of two graphs. Schema information is used to produce semantic relations among all the concepts and the structure is then traversed to construct the propositional formulas among concepts (equality, overlap, mismatch, granularity).

Extensive surveys and comparisons of some existing approaches to concept matching have been provided in works of [5, 3, 6]. [7] demonstrated a schema matching system. It suggests users the candidate matches for a selected schema element and allows user navigation between the candidates. The candidate matches are chosen with higher ranking of match candidates which is based on lexical similarity, schema structure, element types, and the history of prior matching actions. There are more methods which need manual configuration [8, 9]. In the context of schema concept matching, [10] have argued that full automation is not feasible due to insufficient information currently provided by the database schema, which motivated the use of the additional semantics provided through ontologies in many of the newly developed methods. [11] addresses the issues of supporting ontology-based semantic matching in RDBMS and introduces a set of SQL operators for ontology-based semantic matching. This approach enables users to reference ontology data directly from SQL using the semantic match operators. However, it strongly depends on the assumption that the ontologies are stored in the RDBMS which might not be the most cases of concept matching. [12] presents a concept similarity matching method based on semantic distance, and it considers the inheritance relations and semantic distance relations between concepts, and measures the degree of matching between concepts through semantic similarity. By consulting WordNet, the fragment of the ontology hierarchy concerning concepts can be obtained. However, the problems of this method are the effort made to get the ontology and whether it supports phrase semantic similarity. [13] presents the AgreementMaker system for schema and ontology matching. It combines the structure and semantic similarity to match

elements in schemas. It also requires sophisticated domain experts to be the end users of the system whose needs have driven the design and implementation of the system.

2 Problem Definition and Motivation

The concept term matching problem is the problem of finding a mapping between the concept terms of two (or more) knowledge representations (K_R). Since knowledge representations commonly differ in the amount of specific/general knowledge stored about some aspect of the domain, the number of concept terms will differ among them and the mappings will not just be restricted to one-to-one (1:1) but to many-to-one (M:1), one-to-many (1:M) and many-to-many (M:M). The last case does not occur so often because it can usually be considered as two mappings. Representing it as a 1:M and M:1 match, gives a better indication of the point in the structure (particular concept) where the difference in the knowledge level of detail occurs. A complex match generally indicates that while one K_R uses a singular concept term to represent some aspect of the domain, the second K_R uses multiple concept terms or stores more detailed information about that aspect of the domain. As a simple example, consider Fig. 1 (schemas obtained from www.ontologymatching.org) where the concept with label ‘DeliverTo’ in KR_2 corresponds to two concepts in KR_1 namely ‘DeliverTo’ and ‘Address’ (the same holds for the concept ‘BillTo’). The ‘DeliverTo’ and (‘DeliverTo’, ‘Address’) concept terms both describe one concept or aspect of the domain, which is where to deliver the goods.

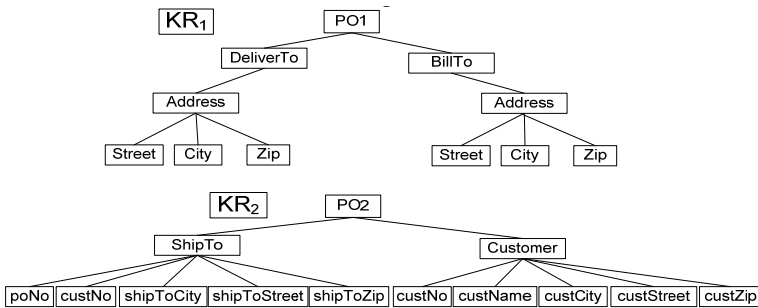


Fig. 1. XML schema structures describing different post order documents

The problem of concept matching can be generally stated as follows.

Definition: Let KR_1 and KR_2 correspond to two knowledge representations, and let $C_1 = \{1c_1, 1c_2, \dots, 1c_n\}$ (where $n = |C_1|$) and $C_2 = \{2c_1, 2c_2, \dots, 2c_m\}$ (where $m = |C_2|$) are complete sets of concept terms used in KR_1 and KR_2 , respectively. The task of concept term matching is to find a set of mappings $M = \{m_1, m_2 \dots m_p\}$ (where $p = |M|$), where each $m \in M$ is a 2-tuple denoted as (e, f) where e and f are sets of 1 or more concepts and $e \subseteq C_1$ and set $f \subseteq C_2$.

Most of the sets e and f from the mappings m will consist of a single concept, but in cases of a complex match, a number of concept terms will be present in one of the

sets. In ideal cases, the set M would contain all concept terms from C_1 and C_2 in all the sets e and f from its elements, respectively. However, due to the many different ways in which knowledge is represented by different organizations, there may remain some elements from C_1 and/or C_2 which are not present in the respective element sets (e and f) of the found mappings (m). This usually occurs in cases where the extent to which the domain knowledge is covered by one organization is much larger, and there are some aspects of the domain totally unaccounted for by the other organization. While the problem has been stated here for two knowledge representations, it is easily extended for three or more knowledge representations by performing the task for each pair separately. While the knowledge representations in general mainly follow a tree or graph structure, in this paper we focus on knowledge representations where the underlying structure is a tree, i.e. no cycles are allowed.

2.1 Motivation of the Proposed Concept Matching Technique

There exist a number of ways and criteria for forming the mappings and many are based on comparing the concept terms using string similarity measures, online dictionaries and thesauruses (eg. WordNet [14]). In this case, the actual string label used for describing a concept of the domain is used as the basis for comparison. In the example given in Fig. 1, the concept terms describing the same aspect of the domain are represented by different labels and there is a different level of granularity amongst the representations. While the WordNet and string similarity measures may detect some of the valid matches by performing label to label comparisons, the nodes with labels 'Street', 'City' and 'Zip' occur twice in the KR_1 in different contexts. Unless the structure in which the concept terms are presented is analysed, it would be very hard to automatically determine the correct matches with respect to the context in which they were used. Generally speaking, approaches based on string similarity and online dictionaries, can work well for some cases, but at the same time they are not always reliable since different naming conventions are used among knowledge representations and the same name may refer to different things at times or be used in different contexts.

In the work presented in this paper, the aim is to initially avoid making comparisons based on string labels, but rather analyze the structure in which the concept terms are presented. We aim to approach the problem in a different manner to most existing approaches in order to see how close to the correct set of mappings one can get without considering the labels. Hence, the main aim is to use the structural information in which concept terms occur in a particular knowledge representation. Taking the position of the concept terms in the representational structure into account, is to some extent a promising approach for taking the context in which the concepts are used into account. Furthermore, there is usually some indication of the possible complex matches which is not so easily obtained by label comparison. The way that these issues are handled will be explained in the next section, but first we will lay the necessary grounds for understanding our approach. The basic idea of the approach is to consider all of the substructures from the given knowledge structure and from an initial match propagate the similarity among other nodes within the substructures of same size and structural properties where both concepts from the initial match have occurred. Therefore the first required step is to extract substructures of all sizes from a

given knowledge representation. Semi-structured documents such as XML, can be effectively modelled using a rooted ordered labelled tree.

A *tree* is a special type of graph where no cycles are allowed. It consists of a set of *nodes* (or *vertices*) that are connected by *edges*. Each edge has two nodes associated with it. A *path* is defined as a finite sequence of edges and in a tree there is a single unique path between any two nodes. A *rooted tree* has its top-most node defined as the *root* that has no incoming edges and for every other node there is a path between the root and that node. A node u is said to be a *parent* of node v , if there is a directed edge from u to v . Node v is then said to be a *child* of node u . The *ancestors* of a node u are the nodes on the path between the root and u , excluding u itself. The *descendants* of a node v can then be defined as those nodes that have v as their ancestor. A tree is *ordered* if the children of each internal node are ordered from left to right.

A rooted ordered labelled tree can be denoted as $T(v_0, V, L, E)$, where (1) $v_0 \in V$ is the root vertex; (2) V is the set of vertices or nodes; (3) L is the set of labels of vertices, for any vertex $v \in V$, $L(v)$ is the label of v ; and (4) $E = \{(x, y) | x, y \in V\}$ is the set of edges in the tree. The problem of frequent subtree mining can be generally stated as:

Given a database of trees, T_{db} , and minimum support threshold (σ), find all subtrees that occur at least σ times in T_{db} . The most commonly mined subtree types are induced and embedded. Given a tree $S = (v_{s_0}, V_S, L_S, E_S)$ and tree $T = (v_{t_0}, V_T, L_T, E_T)$, S is an **ordered induced subtree** of T , iff (1) $V_S \subseteq V_T$; (2) $L_S \subseteq L_T$, and $L_S(v) = L_T(v)$; (3) $E_S \subseteq E_T$; and (4) the left to right ordering of sibling nodes in the original tree is preserved. Given a tree $S = (v_{s_0}, V_S, L_S, E_S)$ and tree $T = (v_{t_0}, V_T, L_T, E_T)$, S is an **ordered embedded subtree** of T , iff (1) $V_S \subseteq V_T$; (2) $L_S \subseteq L_T$, and $L_S(v) = L_T(v)$; (3) if $(v_1, v_2) \in E_S$ then $parent(v_2) = v_1$ in S and v_1 is ancestor of v_2 in T ; and (4) the left to right ordering of sibling nodes in the original tree is preserved

3 Method Overview

This section provides an overview of the proposed method for concept matching. The process as a whole is illustrated in Fig. 2. The method takes as input an initial match and two tree-structured knowledge representations (KRs), represented as rooted ordered labeled trees (see Section 2). The root nodes of the knowledge representations are taken as the initial match. The method starts without considering the labels and utilizes only the structural properties of the KRs to form candidate mappings. We use our previously developed frequent subtree mining algorithm [15] to extract all of the substructures from the given KRs. The process continues by traversing the subtree sets from each KR in which the matched concepts occur. Whenever a pair of subtrees (one from KR_1 and one from KR_2) of same size have the same structure and the matched concepts occur at the same position, the similarity value between the remaining concept terms in those subtrees is increased. Once the similarity value between two concepts exceeds a chosen threshold, they become new matches and the process of subtree traversal and similarity update repeats. This process is repeated until either, all concepts from one of the KRs are present in the formed matches, or the number of iteration exceeds a chosen threshold. The reason for using this approach first is that by utilizing the structural properties, the method can detect

candidate 1:M matches and it is likely to form candidate mappings with the context in which the concepts are used, taken into account. The formed mappings are validated using a string similarity measure and WordNet, and the process repeats by disallowing matches, previously determined to be invalid. In all of our discussion we assume two KR_s, but in cases there are more KR_s, the method will be performed for each pair separately.

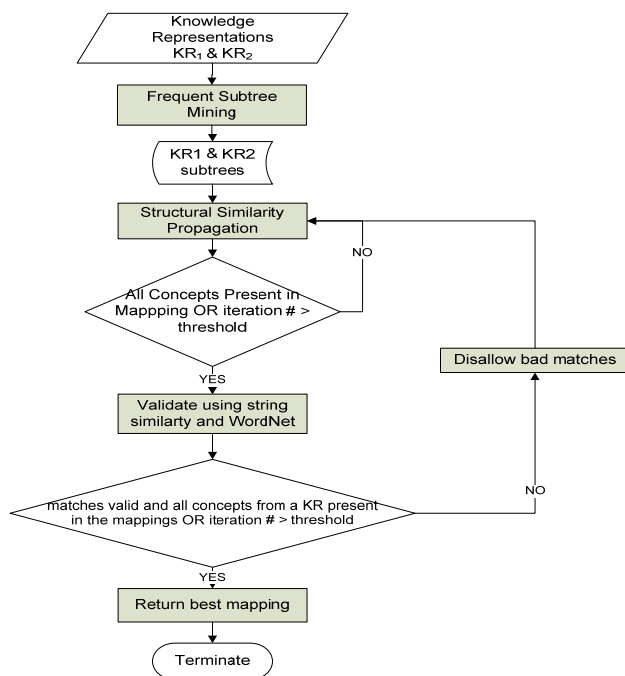


Fig. 2. High level description of the method

4 Detailed Description of the Method

The subtrees extracted from the knowledge representations are of embedded subtree type, and the IMB3 algorithm [15] is used. The reason for choosing to extract embedded subtrees is that when propagating similarity among the substructures, it is important to propagate the similarity not only for all the immediate nodes in the subtrees (i.e. children) but also to all the descendants, as it is well possible that additional nodes may exist in representations, and hence if only induced subtrees were considered, the candidate 1:M matches detected would be limited.

Substructure Similarity Propagation (SSP) Process. The process as a whole can be described by the flowchart represented in Fig. 3. A similarity matrix is set up where all the concept terms from KR₁ and KR₂ are organized into rows and columns in pre-order traversal of the underlying tree structures. There is a corresponding entry for all combinations of possible 1:1 concept term mappings. The process starts by

taking the root nodes of the knowledge representations as the initial match. The process continues by traversing the subtree sets from each knowledge representation in which the matched concepts occur. Whenever a pair of k -subtrees (subtrees consisting of k nodes, one from KR_1 and one from KR_2) have the same structure and the matched concepts occur at the same position, the similarity value between the remaining concept terms in those subtrees is increased in the corresponding entry in the similarity matrix.

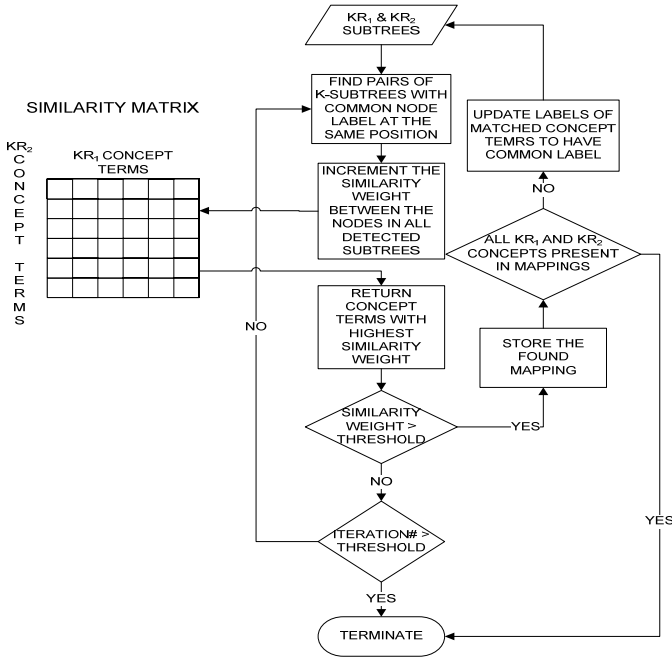


Fig. 3. Substructure Similarity Propagation

Whenever a similarity value between two concept terms exceeds a predetermined threshold value, it becomes a new match and the process is repeated using the newly matched concepts. Furthermore, the possibility of complex matches is checked during the process. For example, these matches occur between the concept terms from KR_1 in the similarity matrix which do not have a sufficiently high similarity value to any of the concept terms in KR_2 to exceed the threshold, but usually have a number of neighboring concept terms from KR_1 that have close similarity values with a particular concept term in KR_2 . When a complex match (e.g. 1:M) is determined, the same process can be applied for similarity update except that the ‘many’ side of a match is considered as a single concept term. Hence, rather than comparing k -subtrees from KR_1 and KR_2 , a k -subtree from KR_1 is compared with a $((k-1)+c_n)$ -subtree from KR_2 , where c_n is equal to the number of concept terms in KR_2 which have been determined to constitute the part of the M in the complex 1:M match. The whole

process repeats until either all the concept terms in the KR_1 and KR_2 are present in the found mappings or a predetermined number of iterations have been exceeded.

Validating the candidate mappings. Once the mappings have been formed, the WordNet online dictionary and string similarity comparison are used to validate the mappings. WordNet allows us to verify the linguistic similarity, while the string similarity compares concept names based on string edit distance, sound similarity, and handles abbreviations and different order of name components.

Each of these validating methods returns a similarity value and the highest one returned is used to validate the candidate match. Hence if the similarity value is below a chosen threshold the candidate match will be considered incorrect and remembered so that when new candidate mappings are formed through structural similarity propagation, this particular match would not be allowed. However, there are exceptions to this, as at times both measures could consider a match incorrect, when it is in fact a correct match. For example, the 'BillTo' and 'Customer' are in fact valid matches, even though both validation methods would consider them as incorrect. To handle such cases, extra logic is integrated in the validating process so that if the children (and some close descendants) of two nodes have been determined to match, and their parent (or ancestor) is considered as a match by the structural similarity propagation method, then it will be considered as a valid match. Another exception to the general rule is when an 1:M match is evaluated. Essentially if the term from one side has a sufficiently high similarity value to only one of the terms from the many side, then the formed match will not be considered as invalid, as it could potentially reflect difference in granularity among the KRs. On the other hand, if the term on one side has a high similarity value to all the terms on the many side, then it will be definitely considered as a true match. When the method terminates because the number of iterations has reached the threshold, mappings formed need to be evaluated. The criteria used are as follows. Given that there is a mapping that contains the highest number of valid matches, then this mapping is returned, and if there are more than one such mappings then we take into consideration the number of invalid matches in the mapping, and hence a further criterion will be the mapping that has least invalid matches. Additional criterion used is for a mapping to contain the least number of unmatched concepts (i.e. concepts not present in any of the formed matches). The mappings that have unmatched concepts where all of those concepts came from 1 knowledge representation would take the priority. Furthermore, we give high priority to mappings containing a 1:M match as opposed to a mapping containing the same number of valid matches, but where the number of concepts present in the matches is smaller. In fact this criterion would be handled by the logic above, since the mapping that has the largest number of concepts on the many side of a valid 1:M match, would also have the least unmatched concepts.

5 Experimental Evaluation

Two pairs of real world XML schemas (<http://www.ontologymatching.org>), namely two Purchase Order schemas as presented in Fig.1 (experiment 1) and Account owner schemas (experiment 2), are used to evaluate the proposed method. For experiment 1 the formed mappings were as follows (refer to Fig. 1): $\{(\{ 'PO1' \}, \{ 'PO2' \})$,

{{'DeliverTo', 'Address'}, {'ShipTo'}}, ({{'BillTo', 'Address'}, {'Customer'}}), ({{'City'}, {'shipToCity'}}), ({{'Street'}, {'shipToStreet'}}), ({{'Zip'}, {'shipToZip'}}), ({{'City'}, {'custCity'}}), ({{'Street'}, {'custStreet'}}), ({{'Zip'}, {'custZip'}}). Please note that the concepts appearing on the left of the knowledge representations (see Fig. 1) are listed first. Since our method initially ignore concept labels and considers only structural positions, these were correctly mapped to the concepts in the right context. In this case all the concept terms from KR_1 are present in the mappings detected, while the unmatched concepts from KR_2 are poNo, custNo, custName and custNo. These could have been represented as part of an 1:M match, as they indicate that there is a difference in granularity at a particular position in the knowledge representation. However, these were not detected by our method because the concepts from KR_1 were all present in the detected mapping and the similarity propagation process halted at that stage. It is necessary to mention that the mapping pairs, ({{'DeliverTo', 'Address'}, {'ShipTo'}}) and ({{'BillTo', 'Address'}, {'Customer'}}) found in this experiment, are 2:1 matches. For the same data the S-Match[16] method detects all the same matches as shown above except for these two 2:1 matches. S-Match finds only the ({{'DeliverTo'}, {'ShipTo'}}) and ({{'BillTo'}, {'Customer'}}) matches since the structural matching of their child nodes is not taken into account to consider them as complex matches. One could argue that the matches of ({{'DeliverTo', 'Address'}, {'ShipTo'}}) and ({{'BillTo', 'Address'}, {'Customer'}}) are preferred as they indicate the exact point in which granular difference occurs among the knowlege representations.

Table 1. Detected mapping statistics for account owner schema pairs

Mapping No.	No. of Matches	No. of Valid Matches	% of Valid Matches	No. of Invalid Matches	Avg. of Similarity
1	10	7	70%	3	0.63
2	8	6	75%	2	0.68
3	8	6	75%	2	0.68
4	9	8	89%	1	0.73
5	9	7	78%	2	0.65
6	9	8	89%	1	0.73
7	8	8	100%	0	0.82

In regards to experiments 2, in Table 1 we present the detailed measurement of the mappings detected showing the number of matches, number of valid matches, percentage of valid matches, number of invalid matches and average of the semantic similarity in each mapping. By observing the columns 4 and 6 from Table 1, one can see that as new candidate mappings are formed the % of valid matches and the corresponding average of semantic similarity are mostly increasing. For example the 3 mappings with the highest percentage of valid matches (i.e. mapping no 4, 6 and 7), consistently have high average of semantic similarities, which reflects reliable mappings. Additionally, it is necessary to mention that the validation part of this method does not detect the match between *AccountOwner* from Left.xml and *Customer* from Right.xml, because WordNet and string similarity measure do not return a high similarity value for these two elements. However, two of their children, *Name* and *Address* from Left.xml, match with, *Cname* and *CAddress* from Right.xml.

Thus, *AccountOwner* and *Customer* are considered as a valid match according to the structural information and the fact that their children form valid matches, as was explained in Section 4.

6 Conclusion

A new concept matching technique, which effectively combines structural information, linguistic/semantic and string similarity measures to arrive at a more reliable set of mappings, was presented. It demonstrates application of tree mining algorithms for the purpose of taking contextual information into account and detecting complex matches. Experimental results confirm the effectiveness of the approach by using the real world data set.

References

1. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic Schema Matching with Cupid. In: Proceedings of the International Conference on very Large Data Bases (2001)
2. Melnik, S., Molina-Garcia, H., Rahm, E.: Similarity flooding: a versatile graph matching algorithm. In: Proceedings of ICDE-02 (2002)
3. Noy, N.F.: Semantic integration: a survey of ontology-based approaches. *SIGMOD Record* 33(4), 65–70 (2004)
4. Giunchiglia, F., Shvaiko, P.: Semantic matching. In: Ontologies and Distributed Systems Workshop, IJCAI (2003)
5. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. *Journal on Data Semantics IV* (2005)
6. Doan, A., Halevy, A.: Semantic integration research in the database community: A brief survey. *AI Magazine* (2005)
7. Bernstein, P.A., Melnik, S., Churchill, J.E.: Incremental schema matching. In: Proc. of the 32nd Int'l Conf. on Very Large Data Bases, pp. 1167–1170 (2006)
8. Drumm, C., Schmitt, M., Do, H.: QuickMig - Automatic Schema Matching for Data Migration Projects. In: Proc. ACM CIKM, Lisbon (November 2007)
9. Amarintrarak, N., Runapongsa, S.K., Tongsima, S., Wiwatwattana, N.: SAXM: Semi-automatic XML Schema Mapping. In: The 24th International Technical Conference on Circuits/Systems, Computers and Communications, ITC-CSCC (2009)
10. Sheth, A., Larson, J.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comp. Surveys* 22(3), 183–230 (1990)
11. Das, S., Chong, E.I., Eadon, G., Srinivasan, J.: Supporting ontology-based semantic matching in RDBMS. In: Proc. of 13th VLDB Conf., pp. 1054–1065 (2004)
12. Ge, J., Qiu, Y.: Concept Similarity Matching Based on Semantic Distance. In: Proc. of the 2008 4th Int'l Conf. on Semantics, Knowledge and Grid, SKG (2008)
13. Cruz, I.F., Antonelli, F.P., Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologie. In: Proceedings of VLDB, Demo (2009)
14. Miller, G.A.: WordNet: A Lexical Database for English. *Comm. of the ACM* 38(11) (1995)
15. Tan, H., Dillon, T.S., Hadzic, F., Feng, L., Chang, E.: IMB3-Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding. In: Proc. of the 10th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (2006)
16. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: an Algorithm and an Implementation of Semantic Matching. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) *ESWS 2004. LNCS*, vol. 3053, pp. 61–75. Springer, Heidelberg (2004)

Clustering and Visualizing SOM Results

José Alfredo F. Costa

Department of Electrical Engineering, Federal University, Natal, RN, Brazil
alfredo@ufrnet.br

Abstract. Self-organizing maps (SOM) had been used for input data quantization and visual display of data, an important property that does not exist in most of clustering algorithms. Effective data clustering using SOM involves two or three steps procedure. After proper network training, units can be clustered generating regions of neurons which are related to data clusters. The basic assumption relies on the data density approximation by the neurons through unsupervised learning. This paper presents a gradient-based SOM visualization method and compares it with U-matrix. It also discusses steps toward clustering using SOM and morphological operators. Results using benchmark datasets show that the new method is more robust to choice of parameters in the filtering phase than the conventional method. The paper also proposes an enhancing method to map visualization taking advantage of the neurons activity, which improve cluster detection especially in small maps.

Keywords: Data clustering, visualization, self-organizing maps.

1 Introduction

Data mining had been established as a major research and application areas, as the number, complexity and size of databases grow daily [1]. A number of processes have been developed to deal with multivariate data. Supervised methods are used when the desired outcomes associated to a set of inputs are available. Tasks such as rules generation, classification and some data reduction methods use the desired response to model internal parameters and optimize the algorithm to the specific function. However, in many cases, labeling the samples may be costly or even impossible. In this case, unsupervised methods are applied, which work directly with the input data. By establishing a similarity criterion, raw data can be grouped in clusters. Advantages include data reduction and modeling. In many cases it is possible to derive rules and general profile of clusters, which may aid future unlabeled data to be assigned to existing classes.

Data clustering has been an active research and application field in data mining. The objective is to find a convenient and valid organization (to identify “natural” groupings) of multivariate data, based on similarities among the patterns. Clustering algorithms can be roughly grouped in five main groups: partitioning methods; hierarchical methods; density-based clustering; grid-based and model-based algorithms [2]. Conventional methods, such as k-means, may impose a structure on the data rather than finding it.

A wide range of neural networks had been proposed, including competitive learning models. In general, a number of units receive an information vector and units compete each other to quantize the data. The winner unit has its weight vector updated in the direction of the input data. Self-organizing maps (SOM) are one of best known models in unsupervised (or competitive) paradigm [3]. SOM had been used for input data quantization as well as visualization of data, an important property that does not exist in most of clustering algorithms.

However, instead of using the same number of neurons as the expected number of data clusters, effective clustering using SOM involves two or three steps procedure. After proper network training, contiguous regions of the map can be associated to complex shaped geometries in the input space. Visual inspection from distance matrix also had been widely used. However, grid segmentation must be done. The basic assumption relies on the data density approximation by the neurons through unsupervised learning.

This paper presents a gradient-based SOM visualization method and compares it with conventional methods, such as *U-matrix* [4]. It also discusses steps toward clustering using SOM and morphological operators. Results show that the new method is more robust than the conventional method regarding the choice of filtering parameters, which occurs prior the application of watershed transform. Section 3 discusses the new approach and also presents an enhancing method to distance matrices visualization by taking advantage of the neurons activity, which improve SOM cluster detection especially in small maps. Results are described in section 4 after using benchmark datasets. Section 5 concludes the paper with some final remarks and possible future extensions of this work.

2 A Brief on Data Clustering Using SOM

A number of methods for visualizing data relations in a trained SOM have been proposed, including multiple views of component planes, mesh visualization using projections and 2D and 3D surface plots of distance matrices [5,6]. Some former papers dealing with SOM and clustering used 1D or 2D small networks in a one-neuron per cluster fashion [7]. This approach makes the SOM works like k-means, enabling only discovery of hyperspherical shaped clusters.

The basic SOM output information for an input pattern is the index of the winner neuron. Neuron activity, the number of associated patterns to a neuron, usually is also taken in consideration. SOM visualization is traditionally performed using the *U-matrix* [4], which enables the visualization of inter neuron's distances in a 3-D surface landscape or a monochromatic image. It had been used as an interactive aid for exploration of cluster's borders, which appears as high values in the *U-matrix*. Similar neighboring neurons will present small inter distances and therefore will appear as valleys in the landscape. However in some cases *U-matrix* visualizations may be noisy and cluster borders not clear. Some affecting factors are the complexity of map embedding in high dimension, interpolating units and other factors, such as dimension mismatch in input data and SOM topology. Many literature papers described manual labeling of the SOM after unsupervised mapping. Some papers describe the usage of

hits histogram or matrix distances between neurons just as a visualization tool to indicate cluster tendencies.

Lampinen and Oja [8] described a two layer SOM for clustering. The task of the second layer is analogous to clustering of the SOM by k-means algorithm. The desired number of clusters is required and the proposed method is only feasible for hyper-spherical-shaped clusters. Murtagh [9] proposed an agglomerative contiguity-constrained clustering method on the SOM. The method groups the output from SOM based on a minimal distance criterion to merge the neighboring nodes together. The algorithm presented by Kiang [10] extends the Murtagh's approach by using the minimal variance criterion. However, both algorithms need to recalculate the centre after every merging of two clusters and are appropriate only for or hyper-ellipsoidal shaped clusters.

Given that SOM provides a good approximation to the input space, attempting to preserve topology and approximate the data density, it is possible to estimate information about the original data clusters by analyzing the geometric relations of the neurons after the proper training of the net. Recently, many two-level approaches have been described to effectively use SOM as a clustering tool. The basic idea is to use a clustering algorithm after SOM training, as a post-processing technique [11-15]. These algorithms aim to automatically (or semi-automatically) interpret the neurons of a trained SOM. In this approach, usually, the number of neurons is higher than the expected number of clusters. The objective is to group neighboring neurons and obtain regions of units which approximate the geometry of the clusters [11].

Costa and Netto [11-14] proposed the automated segmentation of SOM using morphological analysis of the U-matrix. The algorithm applies mathematical morphology operations, such as filtering and watershed transform, to segment the U-matrix. Image markers are found by scale methods, either using the largest plateau of stability between the number of connected regions versus inter-neuron distance threshold or using modified validity cluster indexes. After image homotopy modification and watershed application, segmented regions are labeled as well as the corresponding neurons. Variations of this algorithm had been used in a number of applications, e.g., in satellite image segmentation [13, 14], clustering schema elements for semantic integration of heterogeneous data sources [16], determining the number of operational modes in baseline multivariate SPC data [17] and clustering of EEG-segments [18].

Vesanto and Alhoniemi [15] described both hierarchical agglomerative clustering and partitioning clustering using k-means for clustering of the SOM. The experiments indicated that clustering the SOM instead of directly clustering the data is a computationally effective approach. However, hierarchical clustering algorithms in [9] used only inter-cluster distance criterion to cluster the output neurons and, in a similar way that algorithm presented [9]. In the case of batch k-means algorithm, described by authors, it is required the desired number of clusters in advance and is only feasible for hyper-spherical-shaped clusters. The authors reported usage of many runs with different parameters to obtain cluster indexes over different values of k in order to find the feasible value. However, the approach was not able to properly recognize nonspherical clusters and in some applications, small clusters had merged in one large cluster.

Wu and Chow [19] proposed method to clustering of the SOM uses the clustering validity index locally to determine which pair of clusters to be merged. Compared with others classical clustering methods applied to the SOM, the algorithm utilizes more information about the data in each cluster in addition to inter-cluster distances. Experimental results were described on four data sets presented, two synthetic and two real data (iris and wine) and results were coherent with expectative. In the case of iris data, only two clusters were automatically detected.

Brugger et al. [20] also addressed the problem of automatic cluster detection in trained maps. The authors described the Clusot algorithm, a two-step procedure. The first step involves the computation of the Clusot surface which is used in the second step to detect the clusters. Instead of using only neuron distances, such as in the U-Matrix derived methods, Clusot uses also neuron frequency to obtain the Clusot surface. Local maxima in the surface indicate a cluster. To enable cluster detection, two algorithms were described, a Gradient-Based Cluster Detection and a Recursive Flooding Cluster Detection. In the first approach, the authors reported that there was a problem with finding an appropriate mapping for the SOM information. The second approach resembles the watershed algorithm, which was used by Costa and Netto [11-12]. When compared with conventional data clustering methods applied to SOM [15] the Clusot presented inferior or equivalent results. Also, both methods described in refs. [15] and [20] were not capable of automatically detecting complex structures such as the chainlink data [4], which was possible using the regularized watershed on U-matrix [21].

3 Gradient-Based SOM Matrix with Applications to Data Clustering and Visualization

The *U-matrix* enables to visualize the resulting SOM mapping [4] by computing distances between adjacent neurons. For a map with size $X \times Y$ neurons, the U-matrix will have size $(2X-1) \times (2Y-1)$. Let k be a neuron on the map, $NN(k)$ be the set of immediate neighbors on the map, $w(k)$ the weight vector associated with neuron k , then

$$U(k) = \sum_{m \in NN(k)} \|w_k - w_m\| \quad (1)$$

where $\|\cdot\|$ is a distance measure similar to used to train the map. The U-matrix is a inter-neuron distance image, $U(k)$. Its values can be visualized as an image or a three dimensional landscape. Interpolation takes place in calculating intermediate $(2x-1, 2y-1)$ positions (x and y as map coordinates), whereas mean or median values are obtained from calculated distances. High values on the *U-matrix* encode dissimilarities between neurons, associated to cluster borders [4, 5, 11-15].

3.1 A New Gradient-Based Visualization Matrix

A visualization matrix can be derived from SOM components gradient. Given the data dimension, D , which is the same number of map components, the approach is obtain the gradient of D components in the directions (O output dimensions) of the map. In most of cases, the output dimension is two. For a neuron F with coordinates (i, j) , the

value associated to its height is the root sum of square components' gradient, given, for 2-D output grid (x and y directions) by:

$$U(i, j) = \left\{ \sum_{k=1}^D \left[\left(\frac{\partial F_k(i, j)}{\partial x} \right)^2 + \left(\frac{\partial F_k(i, j)}{\partial y} \right)^2 \right] \right\}^{\frac{1}{2}} \tag{2}$$

For a better visualization and image analysis, bilinear interpolation can be undertaken. For higher output grid dimensions, it can be added new terms, generating, for example, for 3-D output grid, $U(i, j, l)$ with a derivative term with l inside the sum brackets. Higher output grid analysis of SOM can be derived, in a similar way of [21].

3.2 Weights to Improve Cluster Visualization in U-Matrix

Weights derived from neuron activities that can be used directly to enhance distance or gradient matrices. Consider a dataset S obtained from a uniform distribution, i.e., absence of clusters. It is expected that an $X \times Y$ map will spread its neurons, trying to approximate the data density. Avoiding neuron border effects (to simplify in a first analysis), it is supposed that the expected neuron activity mean \hat{h} to be roughly the cardinality of data set divided by the total number of neurons.

$$\hat{h} = |S| / \prod_{i=1}^O d_i \tag{3}$$

Where d_i , $i = 1, \dots, O$, are the map output dimensions. In the uniform density generated map, each neuron activity h_i and their mean \bar{h} are expected to have value near to \hat{h} . The standard deviation of all neuron activities, sd , is also expected to have a low value. Therefore, in this particular case, mean interneuron distances and activities are expected to be roughly similar, with low standard deviation. A different situation occurs when clusters occur and are captured in regions of the map. Activity within clusters is high than some other areas. Interneuron distances in cluster regions are smaller than between clusters. The map try to capture the data manifold and it is expected that some neurons act as linking units (some with null or low activity). A weight set derived from neuron activity might improve SOM visualization by enhancing cluster borders (low activity areas, higher neighboring neuron distances) and attenuate distances in high activity areas. A basic approach could be to find at least three categories of neurons based on their activities: weak, normal and strong activity neurons. One simple way to do this is based on mean of map activity and standard deviation. The logical matrix that indicates strong neurons, S (with elements s_i , whereas i is the neuron index), could be derived from:

$$s_i = h_i > \bar{h} + \alpha \cdot sd, \text{ for all } i \text{ neurons} \tag{4}$$

Where \bar{h} and sd are the neuron activity mean and standard deviation. Parameter α can be used to modulate how far from mean the neuron activity must exceed to belong to S . Otherwise, weak neurons set, K , could be regarded to neurons with activity bellow map activity mean minus βsd .

$$k_i = h_i < \bar{h} - \beta \cdot sd, \text{ for all } i \text{ neurons} \tag{5}$$

In the case of $\bar{h} - \beta \cdot sd$ present a negative value, zero is considered instead. All other neurons not marked in the two logical sets, S and K , can be considered normal activity neurons, N .

As we desire to enhance distances from set K and decrease within distances from set S , one simple way is to obtain a weighted sum from the sets, considering higher weights to K and lower to S . The set P' can be obtained from:

$$P' = \sigma_1 \cdot S + \sigma_2 \cdot N + \sigma_3 \cdot K \quad (6)$$

Where it is expected that $\sigma_1 \leq \sigma_2 \leq \sigma_3$. In all experiments in this paper, the values used to σ_i , $i = 1, \dots, 3$, were 0.5, 1.0 and 1.5, respectively. Values for α and β were set to 1. As U -matrix has almost twice the size of map, bilinear interpolation is used to generate P from P' . Therefore, U_e , the enhanced matrix, can be derived from array multiplication (element-by-element product) from the distance or gradient matrix, U and the interpolated weights, $U_e = P \cdot U$. P and U_e matrices can also be scaled to $[0, 1]$ range prior calculi.

4 Results

A dataset with a mixture of bivariate Gaussian densities was designed from description in ref. [22]. Five classes, with some overlap, were generated each containing 300 samples. The five populations were generated from the mean vectors (0.0) (1.1), (1, -1), (-1, -1), (-1, 1). The covariance matrices are diagonal. For the first class, $\Sigma_1 = \text{diag}(0.2, 0.2)$. The other covariance matrices were obtained using $\Sigma = \text{diag}(0.05, 0.3)$ rotated with angle $\pm\pi/4$.

Some functions from SOM toolbox were used to train maps [23]. Figure 1 present grid and neuron activity for both final neighborhood (FN) 1 and 0, for a 12x12 map, trained with batch algorithm and 250 epochs. Although quantization factors are better in final zero neighborhood, it is seen from figure 2 that a smooth U-matrix is obtained from FN 1.

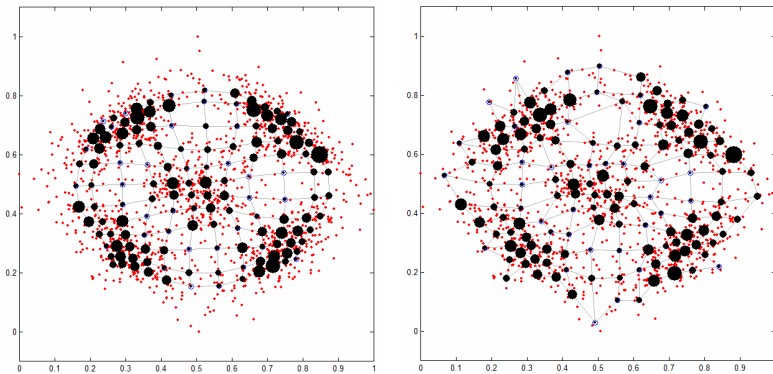


Fig. 1. Grid (12x12 map) and unit activity for FN 1 (left) and 0 (right)

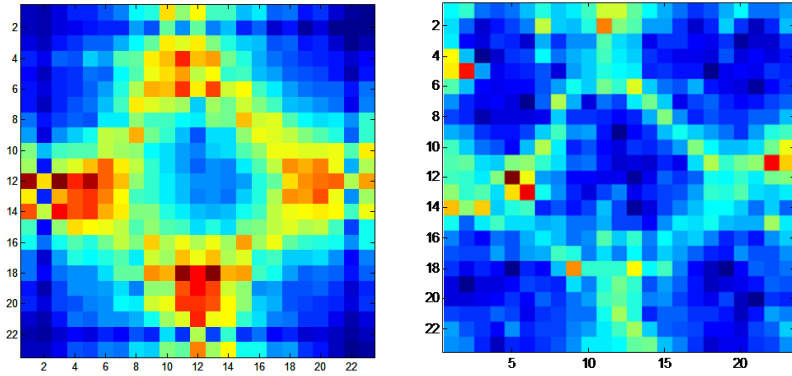


Fig. 2. U-matrices derived from FN 1 (left) and 0 (right)

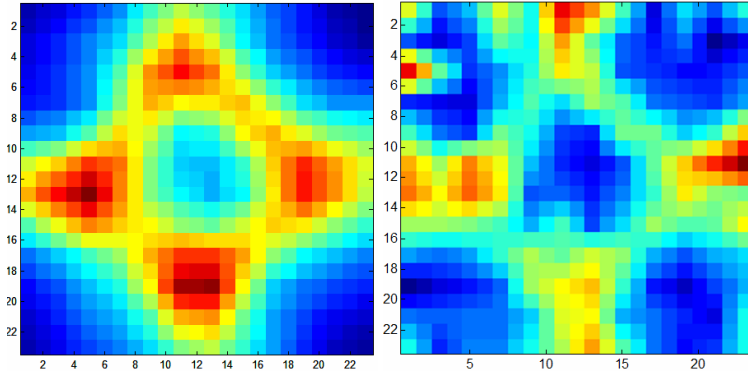


Fig. 3. Gradient matrices derived from FN 1 (left) and 0 (right)

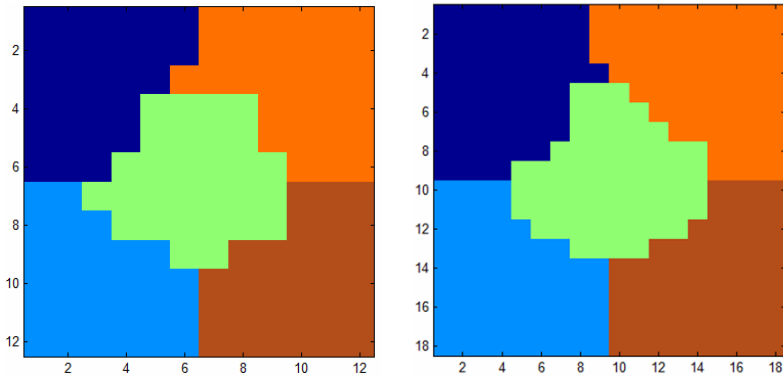


Fig. 4. Segmented SOM after U-matrix and watershed algorithm (SL-SOM), for 12x12 (left) and 18x18 map (right)

Figure 3 show the gradient matrices for FN 1 and 0. Results from two-step SOM clustering using watershed transform [11-14] is shown in figure 4.

Tables 1 and 2 show accuracy for cluster recovery for different map sizes and FN 1 and 0, respectively. U and G represent U-matrix and gradient-matrix. U' and G' represent weighted versions of U and G. In table 1, for a SOM 6x6, (a) means that the algorithm found the number of clusters as 4, different from the real number of clusters, 5. It was possible to obtain the right number of clusters only in the weighted U-matrix. For a 25x25 map, label (b) the algorithm was also in doubt between 2 and 5 clusters. Only using an improved SL-SOM algorithm, using a modified CDbw cluster index, it was possible to automatically obtain 5 clusters, with accuracy 95.5%. In the case of FN equal to 0, for 18x18 map the number of 3 clusters was the most probable for a range of *area_open* parameter [24], used to filter distance matrices prior marker detection. However, for a range from 6 to 10 in this parameter, the number of five clusters was automatically found and the accuracy of 95.0% was obtained. In general, accuracy cluster recovery results are similar, but in many simulations use of weights aided the decision of watershed marker, by widening the stability of number of clusters versus threshold parameter (distance), see [21].

Table 1. Accuracy for cluster recovery for different SOM sizes, FN = 1

SOM size	U	G	U'	G'
6x6	(a)	(a)	90,7%	(a)
12x12	94,6%	94,0%	94,0%	94,7%
18x18	95,0%	95,2%	95,1%	94,7%
25x25	95,7%	95,7%	(b)	95,5%

Table 2. Accuracy for cluster recovery for different SOM sizes, FN = 0

SOM size	U	G	U'	G'
6x6	94,7%	94,7%	94,7%	94,7%
12x12	94,9%	94,9%	94,9%	94,9%
18x18	95% (c)	95,2%	94,3%	94,8%
25x25	94,3%	94,8%	94,7%	95,4%

Relating to this parameter, an experimental sensitivity analysis was conducted, considering different map sizes and stability regions for number of clusters using U and G matrices filtered with morphological operations such as *area_open* and *area_close* [24]. It is shown that the correct number of clusters, 5 in this dataset, appears in stable regions for a larger portion of distance threshold versus filtering parameter, as can be seen from comparing figure 5b (derived from gradient matrix, left size, red area) with 5a (from U-matrix, right side, light green area).

Results for other clustering algorithms, such as *k*-means and EM were also conducted. In the *k*-means case, using a range of *k* from 2 to 10 and using Davies-Bouldin index, only 6 in 20 independent runs found the correct number of clusters. The best accuracy result, comparison of original class with discovered clusters, when forced *k* to 5, real number of densities, was 94.9%. For expectation-maximization

algorithm (EM) [1-3], using Gaussian models, tested for a range of k from 2 to 8, the method was capable to correct identify the number of clusters. The best accuracy result for 20 independent runs was 96.5%. It was expected higher accuracy values since the clustering model matches with dataset.

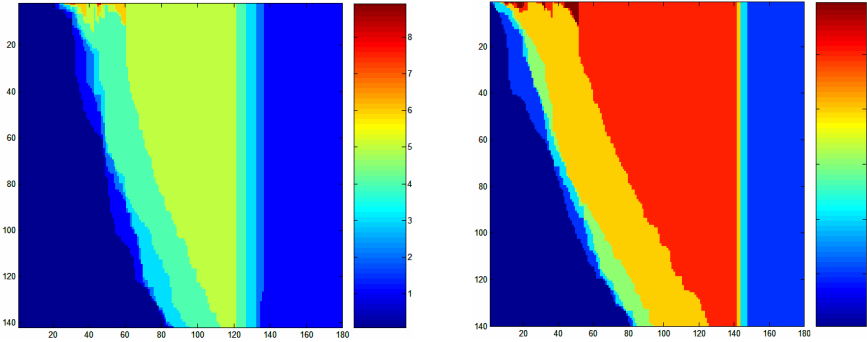


Fig. 5. Stable regions for number of clusters versus distance threshold and versus filtering parameters derived from U-matrix (left) and G-matrix (right), 18x18 map

5 Conclusions and Final Remarks

Cluster analysis aims to discover the hidden structure of data or the generating model of data. Recent literature had shown great interest in algorithms for data clustering and visualization using SOM as well applications from these techniques. However, the automation of knowledge discovery in SOM is not straightforward.

A gradient-based matrix visualization was described as well a technique to enhance matrix visualization, by strengthening distances from neighboring neurons with low activity and decreasing them on the contrary way. Results were shown comparing gradient and U-matrices, with and without weights. Accuracy for cluster recovery and stability for morphological parameter used in first step of process, area open and area close filtering was shown. Results indicate better stability for broader range of filtering parameters for the gradient-based matrix. Results for Iris, Wine, and other benchmark datasets were also conducted, not shown due to space reasons. Future research will regard hierarchy of maps [21] and topology preservation indexes [2-3].

Acknowledgment. Part of this research was conducted with financial support from Brazilian Science Council (CNPq), grants 480043/2008-6 and 201382/2008-3.

References

1. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (June 2005)
2. Xu, R., Wunsch, D.: Clustering. IEEE Press, Los Alamitos (2009)
3. Yin, H.: The Self-Organizing Maps: Background, Theories, Extensions and Applications. Computational Intelligence: A Compendium, 715–762 (2008)

4. Ultsch, A.: Self-Organizing Neural Networks for Visualization and Classification. In: Opitz, O., et al. (eds.) *Information and Classification*, pp. 301–306. Springer, Berlin (1993)
5. Kohonen, T.: *Self-Organizing Maps*, 3rd edn. Springer, Berlin (2001)
6. Vesanto, J.: Using SOM in Data Mining. Licentiate's Thesis, Department of Computer Science and Engineering, Helsinki University of Technology, Espoo, Finland (2000)
7. Curry, B., Davies, F., Phillips, P., Evans, M., Mouthino, L.: The Kohonen self-organizing map: an application to the study of strategic groups in the UK hotel industry. *Expert Systems* 18(1), 19–31 (2001)
8. Lampinen, J., Oja, E.: Clustering properties of hierarchical self-organizing maps. *J. of Math. Im. and Vis.* 2, 261–272 (1992)
9. Murtagh, F.: Interpreting the Kohonen self-organizing feature map using contiguity-constrained clustering. *Pattern Recognition Letters* 16, 399–408 (1995)
10. Kiang, M.Y.: Extending the Kohonen self-organizing map networks for clustering analysis. *Computational Statistics and Data Analysis* 38, 161–180 (2001)
11. Costa, J.A.F., Netto, M.L.A.: Estimating the Number of Clusters in Multivariate Data by Self-Organizing Maps. *Intl. Journal of Neural Systems* 9, 195–202 (1999)
12. Costa, J.A.F., Netto, M.L.A.: Clustering of complex shaped data sets via Kohonen maps and mathematical morphology. In: Dasarathy, B. (ed.) *Proceedings of the SPIE, Data Mining and Knowledge Discovery*, vol. 4384, pp. 16–27 (2001)
13. Goncalves, M.L., de Andrade Netto, M.L., Costa, J.A.F., Zullo, J.: Data Clustering using Self-Organizing Maps segmented by Mathematic Morphology and Simplified Cluster Validity Indexes: An application in remotely sensed images. In: 2006 IEEE Intl. Joint Conf. on Neural Networks, Vancouver, BC, Canada, pp. 4421–4428 (2006)
14. Goncalves, M., Netto, M., Zullo, J., Costa, J.A.F.: A new method for unsupervised classification of remotely sensed images using Kohonen self-organizing maps and agglomerative hierarchical clustering methods. *Intl. Journal of Remote Sensing* 29(11), 3171–3207 (2008)
15. Vesanto, J., Alhoniemi, E.: Clustering of the Self-Organizing Map. *IEEE Trans. on Neural Networks* 11(3), 586–602 (2000)
16. Zhao, H., Ram, S.: Combining schema and instance information for integrating heterogeneous data sources. *Data and Knowledge Engineering* 61(2), 281–303 (2007)
17. Zhang, H., Albin, S.: Determining the number of operational modes in baseline multivariate SPC data. *IIE Trans.* 39(12), 1103–1110 (2007)
18. Sommer, D., Golz, M.: Clustering of EEG-Segments Using Hierarchical Agglomerative Methods and Self-Organizing Maps. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) *ICANN 2001. LNCS*, vol. 2130, pp. 642–649. Springer, Heidelberg (2001)
19. Wu, S., Chow, T.W.S.: Clustering of the Self-organizing Map using a clustering validity index based on inter-cluster and intra-cluster density. *Pattern Recog.* 37, 175–188 (2004)
20. Brugger, D., Bogdan, M., Rosenstiel, W.: Automatic Cluster Detection in Kohonen's SOM. *IEEE Transactions on Neural Networks* 19(3), 442–459 (2008)
21. Costa, J.A.F.: *Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis*. PhD Thesis. UNICAMP (December 1999) (in Portuguese)
22. Hamad, D., Firmin, C., Postaire, J.: Unsupervised pattern classification by neural networks. *Mathematics and Computers in Simulation* 41, 109–116 (1996)
23. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: Self-Organizing Map in Matlab: the SOM Toolbox. In: *Proc. Matlab DSP Conf. 1999, Finland*, pp. 35–40 (1999)
24. Dougherty, E.R., Lotufo, R.A.: *Hands-on Morphological Image Processing*. SPIE Publications (2003)

A Hybrid Evolutionary Algorithm to Quadratic Three-Dimensional Assignment Problem with Local Search for Many-Core Graphics Processors

Piotr Lipinski

Institute of Computer Science,
University of Wrocław, Wrocław, Poland
lipinski@ii.uni.wroc.pl

Abstract. This paper concerns the quadratic three-dimensional assignment problem (Q3AP), an extension of the quadratic assignment problem (QAP), and proposes an efficient hybrid evolutionary algorithm combining stochastic optimization and local search with a number of crossover operators, a number of mutation operators and an auto-adaptation mechanism. Auto-adaptation manages the pool of evolutionary operators applying different operators in different computation phases to better explore the search space and to avoid premature convergence. Local search additionally optimizes populations of candidate solutions and accelerates evolutionary search. It uses a many-core graphics processor to optimize a number of solutions in parallel, which enables its incorporation into the evolutionary algorithm without excessive increases in the computation time. Experiments performed on benchmark Q3AP instances derived from the classic QAP instances proposed by Nugent et al. confirmed that the proposed algorithm is able to find optimal solutions to Q3AP in a reasonable time and outperforms best known results found in the literature.

1 Introduction

The quadratic three-dimensional assignment problem (Q3AP), [7], is an extension of the quadratic assignment problem (QAP), [1], [7], which has recently gained more and more popularity due to its application in the hybrid automatic repeat request (Hybrid-ARQ, HARQ) method in wireless communication systems, among others, in the HS-DPA and HSUPA transmission protocols in mobile phone networks [2].

However, in contrast to the well-known QAP, there are not many approaches to Q3AP proposed in the literature and only a few reports with computational results, [2], [3], [4]. One of the reasons may be the much higher computation time required and the much larger number of $n! \times n!$ feasible solutions to Q3AP compared to the $n!$ feasible solutions to QAP.

This paper proposes an efficient hybrid evolutionary algorithm combining stochastic optimization and local search with a number of crossover operators, a number of mutation operators and an auto-adaptation mechanism [5]. Auto-adaptation manages the pool of evolutionary operators applying different operators in different computation phases to better explore the search space and to avoid premature convergence. Local

search additionally optimizes populations of candidate solutions and accelerates evolutionary search.

Combining evolutionary computation with local search is often a tempting technique [5]. However, the computational cost of such hybrid approaches is usually very high, mainly because of the high cost of local search employed for many candidate solutions in the population. On the other hand, such techniques usually lead to very good results, significantly accelerating evolutionary search by quickly moving the population to promising regions of the search space and consequently leading it to efficient solutions that may be hard to obtain without local search and overlooked by classic evolutionary search.

In recent years, the development of graphical processing units (GPUs) enables new horizons of computing. Advantages of GPU computing perfectly suit such hybrids of evolutionary computation and local search.

This paper is structured in the following manner: Section 2 defines Q3AP. Section 3 introduces the hybrid evolutionary algorithm. Section 4 discusses the experiments. Finally, Section 5 concludes the paper.

2 Quadratic Three-Dimensional Assignment Problem

The first definition of the quadratic three-dimensional assignment problem (Q3AP) was introduced by William P. Pierskalla [7]. It is usually formulated as a minimization problem

$$\min \sum_{i=1}^n \sum_{j=1}^n \sum_{p=1}^n b_{ijp} x_{ijp} + \sum_{i=1}^n \sum_{j=1}^n \sum_{p=1}^n \sum_{k=1}^n \sum_{l=1}^n \sum_{q=1}^n c_{ijpklq} x_{ijp} x_{klq}, \tag{1}$$

where $x_{ijp} \in \{0, 1\}$ and

$$\sum_{j=1}^n \sum_{p=1}^n x_{ijp} = 1, \quad \sum_{i=1}^n \sum_{p=1}^n x_{ijp} = 1, \quad \sum_{i=1}^n \sum_{j=1}^n x_{ijp} = 1, \tag{2}$$

with given $n \in \mathbb{N}$, $b_{ijp} \in \mathbb{R}$ and $c_{ijpklq} \in \mathbb{R}$.

3 Hybrid Evolutionary Algorithm to Q3AP

3.1 Objective Function and Search Space

Each feasible solution to the problem defined in the previous section may be equivalently represented by a pair of permutations $\mathbf{p} = (p_1, p_2, \dots, p_n) \in \Pi_n$ and $\mathbf{q} = (q_1, q_2, \dots, q_n) \in \Pi_n$, where Π_n denotes the set of permutations of the set $\{1, 2, \dots, n\}$ (in such a way that the elements x_{ijp} from the previous section equal 1 if and only if $j = p_i$ and $p = q_i$). Hence, the objective function may be equivalently represented by

$$F(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n b_{ip_i q_i} + \sum_{i=1}^n \sum_{j=1}^n c_{ip_i q_i j p_j q_j}, \tag{3}$$

with the same $n \in \mathbb{N}$, $b_{ijp} \in \mathbb{R}$ and $c_{ijpklq} \in \mathbb{R}$ as in the previous section. Q3AP may be equivalently formulated as a problem of minimizing the objective function $F(\mathbf{p}, \mathbf{q})$ over the search space $\Pi_n \times \Pi_n$.

3.2 Overview of Algorithm

Algorithm 1 presents an overview of the Hybrid Evolutionary Algorithm to Q3AP (HEA-Q3AP). It uses a number n_C of crossover operators and a number n_M of mutation operators (listed in Table 1), an auto-adaptation mechanism for managing them and local search for improving the candidate solutions. It starts with initializing the probabilities of usage of particular crossover operators (with $1/n_C$ for each operator) and particular mutation operators (with $1/n_M$ for each operator), generating a random initial population \mathcal{P}_0 consisting of N candidate solutions, where each candidate solution is a pair $[\mathbf{p}, \mathbf{q}]$ of random permutations \mathbf{p} and \mathbf{q} , as well as evaluating it.

Afterwards, the evolution starts by creating an offspring population \mathcal{O}_t consisting of M offspring solutions (assuming that M is even). In order to produce a pair of offspring solutions, two parent solutions, $[\mathbf{p}_1, \mathbf{q}_1]$ and $[\mathbf{p}_2, \mathbf{q}_2]$, are selected from the current population \mathcal{P}_t using the well-known roulette wheel method [5]. With a crossover probability p_C , a crossover operator, randomly drawn with probabilities \mathbf{p}_C , is applied to the parent solutions (in fact, it is applied twice, once for the first and once for the second permutation) producing two new solutions, $[\tilde{\mathbf{p}}_1, \tilde{\mathbf{q}}_1]$ and $[\tilde{\mathbf{p}}_2, \tilde{\mathbf{q}}_2]$. With the opposite probability $1 - p_C$, the new solutions are copies of the parent solutions. With a mutation probability p_M , each new solution is processed by a mutation operator, randomly drawn with probabilities \mathbf{p}_M , and added to the offspring population. Next, single local search improves the offspring population and $\kappa \cdot M$ best offspring solutions replace their parent solutions if only they outperform them. Finally, auto-adaptation updates the probabilities of usage of evolutionary operators, multiple local search improves the new population \mathcal{P}_{t+1} and the evolution repeats until a termination condition is held (normally, after a certain number of iterations).

3.3 Auto-Adaptation

Auto-adaptation aims at choosing the most efficient crossover or mutation operator for the current phase of the evolutionary algorithm. Crossover and mutation operators have assigned probabilities of usage $p_1^C, p_2^C, \dots, p_{n_C}^C$ and $p_1^M, p_2^M, \dots, p_{n_M}^M$, respectively, forming probability vectors \mathbf{p}_C and \mathbf{p}_M . HEA-Q3AP starts with equal probabilities of usage, $1/n_C$ for crossover operators and $1/n_M$ for mutation operators, and updates them during evolution. In each iteration, evolutionary operators gain scores for producing offspring solutions that outperform their parents (one point for one offspring solution that outperforms at least one of its two parents). Scores are cumulated during evolution in such a way that scores from the current iteration are added to scores gathered earlier multiplied by 0.75 (in order to avoid a continuous domination of some operators). Probabilities of usage are proportional to scores with a constraint that they cannot be lower than 0.05 (in order to avoid a continuous elimination of some operators). Auto-adaptation works independently for crossover and mutation operators.

3.4 Parallel Local Search

HEA-Q3AP applies single local search to improve the offspring population. For each offspring solution $[\tilde{\mathbf{p}}, \tilde{\mathbf{q}}]$, it checks all possible transpositions in $\tilde{\mathbf{p}}$ and in $\tilde{\mathbf{q}}$, considering

Algorithm 1. Hybrid Evolutionary Algorithm to Q3AP (HEA-Q3AP)

```


$\mathbf{p}_C = \mathbf{1}/n_C$ ;  $\mathbf{p}_M = \mathbf{1}/n_M$ ;  

 $\mathcal{P}_0 = \text{Random-Population}(N)$ ;  

 $t = 0$ ;  

while not Termination-Condition( $\mathcal{P}_t$ ) do  

     $\mathcal{O}_t = \emptyset$ ;  

    while  $|\mathcal{O}_t| < M$  do  

         $([\mathbf{p}_1, \mathbf{q}_1], [\mathbf{p}_2, \mathbf{q}_2]) = \text{Parent-Selection}(\mathcal{P}_t)$ ;  

        if Random-Value(0, 1) <  $p_C$  then  

            Crossover-Operator = Crossover-Operator-Selection( $\mathbf{p}_C$ );  

             $([\tilde{\mathbf{p}}_1, \tilde{\mathbf{q}}_1], [\tilde{\mathbf{p}}_2, \tilde{\mathbf{q}}_2]) = \text{Crossover-Operator}([\mathbf{p}_1, \mathbf{q}_1], [\mathbf{p}_2, \mathbf{q}_2])$ ;  

        else  

             $([\tilde{\mathbf{p}}_1, \tilde{\mathbf{q}}_1], [\tilde{\mathbf{p}}_2, \tilde{\mathbf{q}}_2]) = ([\mathbf{p}_1, \mathbf{q}_1], [\mathbf{p}_2, \mathbf{q}_2])$ ;  

        end if  

        if Random-Value(0, 1) <  $p_M$  then  

            Mutation-Operator = Mutation-Operator-Selection( $\mathbf{p}_M$ );  

            Mutation-Operator( $[\tilde{\mathbf{p}}_1, \tilde{\mathbf{q}}_1]$ );  

        end if  

        if Random-Value(0, 1) <  $p_M$  then  

            Mutation-Operator = Mutation-Operator-Selection( $\mathbf{p}_M$ );  

            Mutation-Operator( $[\tilde{\mathbf{p}}_1, \tilde{\mathbf{q}}_1]$ );  

        end if  

         $\mathcal{O}_t = \mathcal{O}_t \cup \{[\tilde{\mathbf{p}}_1, \tilde{\mathbf{q}}_1], [\tilde{\mathbf{p}}_2, \tilde{\mathbf{q}}_2]\}$   

    end while  

    Single-Local-Search( $\mathcal{O}_t$ );  

     $\mathcal{P}_{t+1} = \text{Population-Selection}(\mathcal{P}_t, \mathcal{O}_t)$ ;  

    Crossover-Operator-Assessment( $\mathcal{P}_t, \mathcal{O}_t, \mathbf{p}_C$ );  

    Mutation-Operator-Assessment( $\mathcal{P}_t, \mathcal{O}_t, \mathbf{p}_M$ );  

    Multiple-Local-Search( $\mathcal{P}_{t+1}$ );  

     $t = t + 1$ ;  

end while  

return  $\mathcal{P}_t$ ;


```

Table 1. Evolutionary Operators

operator	type
PMX (PartialMapped Crossover)	crossover
OX (Order Crossover)	crossover
CX (Cycle Crossover)	crossover
PBX (Position Based Crossover)	crossover
OBX (Order Based Crossover)	crossover
PPX (Precedence Preservative Crossover)	crossover
LCSX (Longest Common Subsequence Crossover)	crossover
LOX (Linear Order Crossover)	crossover
a random transposition of \mathbf{p}	mutation
a random transposition of \mathbf{q}	mutation
swapping \mathbf{p} and \mathbf{q}	mutation
replacing \mathbf{p} with a random permutation	mutation

Table 2. Hardware Platform Specification (NVidia GeForce GTX 280)

number of multiprocessors	30
number of registers per multiprocessor	16384
maximum number of threads per block	512
number of threads per warp	32
shared memory per multiprocessor	16 kB
constant memory	64 kB
local memory per thread	16 kB
maximum number of active blocks per multiprocessor	8
maximum number of active warps per multiprocessor	32
maximum number of active threads per multiprocessor	1024

$n \cdot (n - 1)$ possible transpositions in total, and selects the transposition leading to the largest reduction in objective function values. Single local search repeats such an optimization until no improvement is obtained.

HEA-Q3AP applies multiple local search to improve the main population. For each solution $[\mathbf{p}, \mathbf{q}]$, it starts with single local search, which transforms the initial solution $[\mathbf{p}, \mathbf{q}]$ into a local optimum $[\hat{\mathbf{p}}, \hat{\mathbf{q}}]$, and mutating the local optimum $[\hat{\mathbf{p}}, \hat{\mathbf{q}}]$ by some random transpositions. Next, single local search transforms the new solution into a new local optimum $[\bar{\mathbf{p}}, \bar{\mathbf{q}}]$. If the new local optimum outperforms the old one, i.e. $F(\bar{\mathbf{p}}, \bar{\mathbf{q}}) < F(\hat{\mathbf{p}}, \hat{\mathbf{q}})$, $[\bar{\mathbf{p}}, \bar{\mathbf{q}}]$ replaces $[\hat{\mathbf{p}}, \hat{\mathbf{q}}]$. Otherwise, with a probability proportional to $\exp(F(\hat{\mathbf{p}}, \hat{\mathbf{q}}) - F(\bar{\mathbf{p}}, \bar{\mathbf{q}}))$, the new solution $[\bar{\mathbf{p}}, \bar{\mathbf{q}}]$ either remains or is replaced with the local optimum $[\hat{\mathbf{p}}, \hat{\mathbf{q}}]$. Finally, the new solution $[\bar{\mathbf{p}}, \bar{\mathbf{q}}]$ is again mutated and optimized with single local search. Multiple local search repeats such an optimization a number of times.

Single local search uses a many-core graphics processor to optimize a number of solutions in parallel, which enables its incorporation into the evolutionary algorithm without excessive increases in the computation time. In experiments, single local search was run on NVidia GeForce GTX 280 (a many-core graphics processor with 240 cores) with Compute Unified Device Architecture (CUDA), which is a parallel computing architecture, with a specific parallel programming model and instruction set architecture, for many-core GPUs.

Each solution was processed in a separated thread. Threads were organized in blocks. Blocks were run in parallel by multiprocessors, in such a way that a warp of 32 threads was processed at the same time, while the remaining warps of the same blocks were waiting active in a queue. The number of threads per block depended on the problem size, because only 16 kB of shared memory was available for the entire block (thus, in practice, the maximum number of threads per block was approximately equal to 16 kB divided by the size of the solution). Details of the hardware platform specification are presented in Table 2.

4 Experiments

In order to validate the proposed approach, a number of experiments on benchmark data were performed. Benchmark Q3AP instances were derived from the classic QAP instances proposed by Nugent et al. [6], downloaded from the QAPLIB repository [1],

in such a way that the same flow and distance matrices, A and B , were used and the objective function was

$$F(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n \sum_{j=1}^n A_{p_i p_j} \cdot A_{q_i q_j} \cdot B_{ij}^2. \tag{4}$$

Populations of $N = 3840$ candidate solutions were used and offspring populations had the same size, i.e. $M = N$. The crossover probability was $p_C = 0.95$ and the mutation probability was $p_M = 0.05$. Single local search was performed in 60 blocks of 64 threads. Multiple local search used 6 random transpositions to mutate a solution and repeated the optimization 25 (for NUG8, NUG10, NUG12) or 50 (for NUG13, NUG14, NUG15, NUG20, NUG30) times. Population replacement compared $0.75 \cdot M$ best offspring solutions against their parents.

Figure 1 presents a comparison of HEA-Q3AP (a) and its modification without multiple local search (b) and without single and multiple local search (c) on the NUG13 benchmark. Plots correspond to the lowest, the average and the highest value of the objective function in successive iterations for a typical case. HEA-Q3AP found the global minimum (1912) after 242 iterations in 3737 seconds (in the worst of 8 runs of the algorithm, see Table 3), while its two modifications stopped on two local minima (1996 and 2486) found after 82 and 6761 iterations in 29 and 436 seconds, respectively.

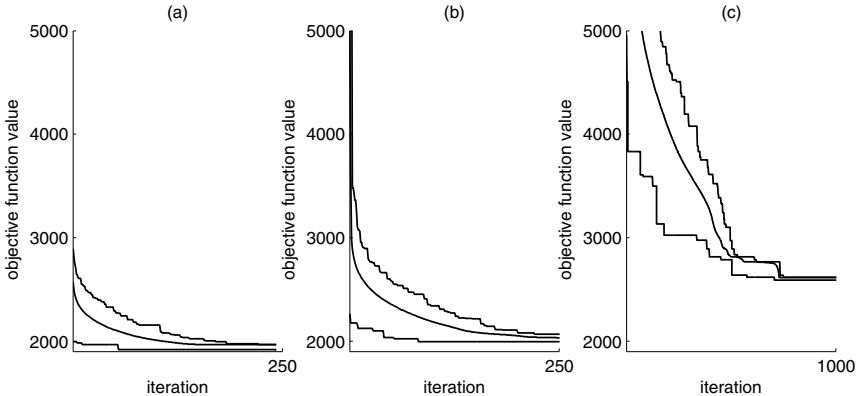


Fig. 1. Comparison of HEA-Q3AP (a) and its modification without multiple local search (b) and without single and multiple local search (c) on the NUG13 benchmark (plots correspond to the lowest, the average and the highest value of the objective function in successive iterations for a typical case)

It is easy to see that applying local search in HEA-Q3AP enables avoiding premature convergence to local minima and discovering the global minimum. Certainly, the computation time per iteration significantly differs in these algorithms (15.44, 0.35 and 0.06 seconds, respectively) and Figure 1 does not reflect it. Results for the remaining benchmarks are comparable, so due to size constraints, they are not presented in detail.

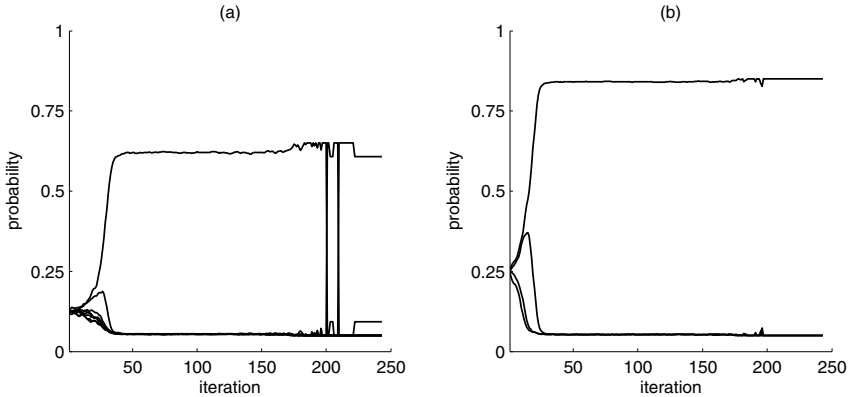


Fig. 2. Probabilities of usage of evolutionary operators in successive iterations of HEA-Q3AP run on the NUG13 benchmark: (a) crossover operators, (b) mutation operators

Table 3. Summary of results and their comparison to the best known results found in the literature

	NUG8	NUG10	NUG12	NUG13	NUG14	NUG15
best found result	134	430	580	1912	2320	2230
best time to reach optimum (s)	< 1	1	4	93	535	1087
avg. time to reach optimum (s)	< 1	5	90	732	764	2032
worst time to reach optimum (s)	< 1	11	152	3737	1154	1634
best known result	134	430	580	1912	2320	2230
known avg. objective function value in 1h	—	—	589	1918	—	2399
known avg. time to reach optimum (s)	—	—	2162	3002	—	—

Figure 2 presents the probabilities of usage of evolutionary operators in successive iterations of HEA-Q3AP run on the NUG13 benchmark. Although, in a typical case, after a number of iteration one operator dominates, like in the case of mutation operators (b), auto-adaptation sometimes enables promoting an other, more efficient, operator, like in the case of crossover operators (a).

Different operators were efficient in different computation phases and in different experiments. For instance, for the NUG13 benchmark, the most efficient crossover operator was LCSX (the average probability of usage equal to 0.3383) and the most efficient mutation operator was applying a random transposition to \mathbf{p} (the average probability of usage equal to 0.5953). However, for the NUG12 benchmark, the most efficient crossover operators were PMX, OBX and LOX, while the most efficient mutation operators were applying a random transposition to either \mathbf{p} or \mathbf{q} .

Table 3 presents a summary of results and their comparison to the best known results found in the literature [3]. For each benchmark, HEA-Q3AP was run independently 8 times and stopped after finding the global optimum (found each time). Beside the worst run for the NUG13 benchmark, the computation time did not exceed 1 hour. Solutions to larger problems, such as NUG20 (7750 in less than 12 hours) and NUG30 (28706 in less than 12 hours), are not compared due to the lack of reported results in the literature.

Results prove that the proposed approach is promising and enabled to find the global optimum of the objective function. Although, the computation time is long, in some cases exceeding 1 hour, compared to other research, [2], [3], [4], it seems to be reasonable. However, it is a common problem to compare the computation time from the parallel and the sequential approach as well as different approaches on different hardware configurations.

5 Conclusions

Preliminary results, compared to the best known results found in the literature, confirmed that HEA-Q3AP is able to find optimal solutions to Q3AP in a reasonable time and outperforms other approaches, mainly due to parallel local search on many-core graphics processors.

However, further research is necessary to investigate the efficiency of local search and the influence of auto-adaptation. Further optimization of the prototype implementation of the parallel local search may also improve results.

References

1. Burkard, R., Karisch, S., Rendl, F.: QAPLIB - a quadratic assignment problem library. *Journal of Global Optimization* 10, 391–403 (1997)
2. Guignard, M., Hahn, P., Ding, Z., Kim, B.-J., Samra, H., Stutzle, T., Kanthak, S.: Hybrid ARQ Symbol Mapping in Digital Wireless Communication Systems Based on the Quadratic 3-Dimensional Assignment Problem (Q3AP). In: *Proceedings of the 2005 NSF Design, Service and Manufacturing Grantees and Research Conference (2005)*
3. Hahn, P., Kim, B.-J., Stutzle, T., Kanthak, S., Hightower, W., Samra, H., Ding, Z., Guignard, M.: The Quadratic Three-Dimensional Assignment Problem: Exact and Approximate Solution Methods. *European Journal of Operational Research* 184, 416–428 (2008)
4. Loukil, L., Mehdi, M., Melab, N., Talbi, E.-G., Bouvry, P.: A Parallel Hybrid Genetic Algorithm–Simulated Annealing for Solving Q3AP on Computational Grid. In: *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing*, pp. 1–8. IEEE Computer Society, Los Alamitos (2009)
5. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Heidelberg (1994)
6. Nugent, C., Vollman, T., Ruml, J.: An Experimental Comparison of Techniques for the Assignment of Facilities to Locations. *Operations Research* 16, 150–173 (1968)
7. Pierskalla, W.: *The Multi-Dimensional Assignment Problem*, Technical Memorandum No. 93, Operations Research Department, CASE Institute of Technology (1967)

Evolution Strategies for Objective Functions with Locally Correlated Variables

Piotr Lipinski

Institute of Computer Science,
University of Wrocław, Wrocław, Poland
lipinski@ii.uni.wroc.pl

Abstract. This paper proposes an improvement of Evolutionary Strategies for objective functions with locally correlated variables. It focusses on detecting local dependencies among variables of the objective function on the basis of the current population and transforming the original objective function into a new one of a smaller number of variables. Such a transformation is updated in successive iterations of the evolutionary algorithm to reflect local dependencies over successive neighborhoods of optimal solutions. Experiments performed on some popular benchmark functions confirm that the improved algorithm outperforms the original one.

1 Introduction

Evolutionary Algorithms (EAs), [3], [1], [10], are an increasingly popular technique of solving optimization problems, which try to find an optimum of an objective function on a search space by maintaining a population of candidate solutions, representing data points in the search space, and moving it towards more and more promising regions using some evolutionary operators.

Many practical applications concern optimization problems with a multi-variable objective function with correlated variables, among others, evolutionary financial decision support systems, [5], [7], [8], and evolutionary portfolio optimization systems [6], where the dimension of the search space often exceeds 300. Discovering such correlations between variables provides an opportunity to reduce the dimensionality of the optimization problem and its complexity.

Although there are numerous techniques of detecting global dependencies among variables over the entire search space, usually in a preprocessing phase, such as the Principal Component Analysis (PCA) [4], its non-linear variations, the Linear Discriminant Analysis (LDA) or the Generative Topographical Mapping (GTM) [2], there has been little research on local dependencies over neighborhoods of optimal solutions and detecting them during runtime.

Some recent EAs, called Estimation of Distribution Algorithms (EDAs) [9] try to regard the population of candidate solutions as a data sample with a probability distribution approximating the probability distribution describing optimal solutions. A similar approach is presented in this paper to detect correlations among variables – it treats the current population as the data sample from a neighborhood of optimal solutions and endeavours to discover dependencies among variables in such a neighborhood.

This paper is structured in the following manner: Section 2 defines the problem and presents the motivation to the research. Section 3 introduces the improvement of Evolutionary Strategies for objective functions with locally correlated variables. Section 4 discusses the experiments. Finally, Section 5 concludes the paper.

2 Problem Definition

Evolutionary Strategies (ESs), [1], [10], deal with the problem of minimizing (or maximizing) a specific objective function $F : \mathbb{R}^n \rightarrow \mathbb{R}$ of n variables over the n -dimensional search space \mathbb{R}^n .

In many practical cases, although the objective function F is formally a function of n variables, some of them are correlated, either globally over the entire search space \mathbb{R}^n or locally over a certain subspace $A \subset \mathbb{R}^n$ containing a global minimum (or maximum) of the objective function F , so the original objective function F may be replaced with another function $G : \mathbb{R}^k \rightarrow \mathbb{R}$ of k variables and a mapping $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^k$, where $k \leq n$, which maps n -dimensional vectors $\mathbf{x} = (x_1, x_2, \dots, x_n)$ into k -dimensional vectors $\mathbf{y} = (y_1, y_2, \dots, y_k)$ in such a way that $F(\mathbf{x}) = G(\mathbf{y})$.

Therefore, the problem of minimizing (or maximizing) the objective function F over the search space \mathbb{R}^n may be reduced to the problem of minimizing (or maximizing) the objective function G over the search space \mathbb{R}^k , where $k \leq n$, and then finding n -dimensional vectors \mathbf{x}_0 corresponding to the k -dimensional vector \mathbf{y}_0 being a minimum (or maximum) of the objective function G (i.e. finding \mathbf{x}_0 for which $\Phi(\mathbf{x}_0) = \mathbf{y}_0$).

Figure 1 presents the example of an objective function $F : \mathbb{R}^2 \rightarrow \mathbb{R}$, defined by

$$F(x_1, x_2) = \frac{\cos(2\pi(x_1 - 2x_2))}{|x_1 - 2x_2| + 1}, \tag{1}$$

which is formally a function of two variables, x_1 and x_2 , but the variables are correlated (for the sake of simplicity, the correlation is global), and the objective function F may be presented as

$$F(x_1, x_2) = G(y_1), \tag{2}$$

where

$$G(y_1) = \frac{\cos(2\pi y_1)}{|y_1| + 1}, \tag{3}$$

and

$$y_1 = x_1 - 2x_2, \tag{4}$$

which reduces the two-dimensional optimization problem with the objective function F of two variables to a one-dimensional optimization problem with the objective function G of one variable.

This paper proposes an improvement of ESs for objective functions with locally correlated variables, which is capable of discovering local dependencies among variables of the objective function and locally reducing it to another objective function of a smaller number of variables, which leads to a reduction in the dimensionality of the optimization problem and consequently also in the computation time.

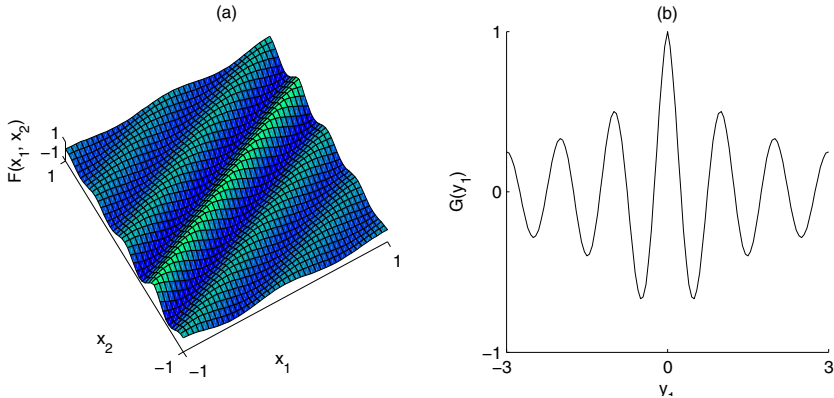


Fig. 1. (a) The objective function $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ of two correlated variables, defined by (1), (b) its reduction to the objective function $G : \mathbb{R} \rightarrow \mathbb{R}$ of one variable, defined by (3)

3 Evolution Strategies with Internal Dimensionality Reduction

Algorithm 1 presents an overview of the Evolution Strategy with Internal Dimensionality Reduction (ESIDR) for an objective function $F : \mathbb{R}^n \rightarrow \mathbb{R}$ with locally correlated variables. It tries to replace, in each iteration t , the original objective function F with another objective function $G_t : \mathbb{R}^{n_t} \rightarrow \mathbb{R}$, a linear mapping $\Phi_t : \mathbb{R}^n \rightarrow \mathbb{R}^{n_t}$ (the reducing mapping) and a linear mapping $\Psi_t : \mathbb{R}^{n_t} \rightarrow \mathbb{R}^n$ (the restoring mapping), where $n_t \leq n$, in such a way that, for each candidate solution $\mathbf{x} \in \mathbb{R}^n$ from the current population \mathcal{P}_t , $G_t(\Phi_t(\mathbf{x})) \approx F(\mathbf{x})$ and $\Psi_t(\Phi_t(\mathbf{x})) \approx \mathbf{x}$. ESIDR applies the Principal Component Analysis (PCA), [4], to the data sample defined by the current population \mathcal{P}_t , and defines the two linear mappings, Φ_t and Ψ_t , by eigenvectors corresponding to n_t largest eigenvalues of the correlation matrix of the data sample, taking the smallest n_t for which the sum of n_t largest eigenvalues exceeds $0.95 \cdot n$. Once Φ_t and Ψ_t defined, ESIDR defines G_t as a composition of Ψ_t and F .

ESIDR runs with a population \mathcal{P}_t composed of N individuals, representing candidate solutions to the optimization problem with the objective function G_t , where each individual $[\mathbf{x}, \boldsymbol{\sigma}]$ consists of a real-number vector $\mathbf{x} = (x_1, x_2, \dots, x_{n_t}) \in \mathbb{R}^{n_t}$ representing an element of the search space \mathbb{R}^{n_t} , i.e. an argument of the objective function G_t , and a real-number vector $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \dots, \sigma_{n_t}) \in \mathbb{R}^{n_t}$ containing mutation parameters which impact on the evolution of the individual, as in classic ESs [10].

ESIDR starts with $n_0 = n$, Ψ_0 being the identity function and $G_0 = F$. It creates an initial population \mathcal{P}_0 at random, in such a way that each gene x_i , for $i = 1, 2, \dots, n_0$, of each chromosome \mathbf{x} is drawn with the standard gaussian distribution $\mathcal{N}(0, 1)$ and each gene σ_i , for $i = 1, 2, \dots, n_0$, of each chromosome $\boldsymbol{\sigma}$ is drawn in the same manner. Once created, the population is evaluated according to the objective function G_0 .

Afterwards, the evolution starts by creating an offspring population \mathcal{O}_t consisting of M offspring individuals (assuming that M is even). In order to produce a pair

Algorithm 1. Evolution Strategy with Internal Dimensionality Reduction (ESIDR)

```

 $n_0 = n; \Psi_0 = \mathbf{1}; G_0 = F;$ 
 $\mathcal{P}_0 = \text{Random-Population}(N);$ 
 $\text{Population-Evaluation}(\mathcal{P}_0, G_0);$ 
 $t = 0;$ 
while not Termination-Condition( $\mathcal{P}_t$ ) do
   $\mathcal{O}_t = \emptyset;$ 
  while  $|\mathcal{O}_t| < M$  do
     $([\mathbf{x}_1, \boldsymbol{\sigma}_1], [\mathbf{x}_2, \boldsymbol{\sigma}_2]) = \text{Parent-Selection}(\mathcal{P}_t);$ 
     $[\tilde{\mathbf{x}}_1, \tilde{\boldsymbol{\sigma}}_1] = \text{Global-Intermediary-Recombination}([\mathbf{x}_1, \boldsymbol{\sigma}_1], [\mathbf{x}_2, \boldsymbol{\sigma}_2]);$ 
     $[\tilde{\mathbf{x}}_2, \tilde{\boldsymbol{\sigma}}_2] = \text{Uniform-Crossover}([\mathbf{x}_1, \boldsymbol{\sigma}_1], [\mathbf{x}_2, \boldsymbol{\sigma}_2]);$ 
     $\text{Mutation}([\tilde{\mathbf{x}}_1, \tilde{\boldsymbol{\sigma}}_1]);$ 
     $\text{Mutation}([\tilde{\mathbf{x}}_2, \tilde{\boldsymbol{\sigma}}_2]);$ 
     $\mathcal{O}_t = \mathcal{O}_t \cup \{[\tilde{\mathbf{x}}_1, \tilde{\boldsymbol{\sigma}}_1], [\tilde{\mathbf{x}}_2, \tilde{\boldsymbol{\sigma}}_2]\};$ 
  end while
   $\mathcal{P}_{t+1} = \text{Population-Selection}(\mathcal{P}_t, \mathcal{O}_t);$ 
   $\text{Restoring-Mutation}(\mathcal{P}_{t+1});$ 
   $t = t + 1;$ 
   $(n_t, \Psi_t, G_t) = \text{Dependency-Mining}(\mathcal{P}_t);$ 
   $\mathcal{P}_t = \text{Population-Transformation}(\mathcal{P}_t, n_t, \Psi_t, G_t);$ 
   $\text{Population-Evaluation}(\mathcal{P}_t, G_t);$ 
end while

```

of offspring individuals, two parent individuals, $[\mathbf{x}_1, \boldsymbol{\sigma}_1]$ and $[\mathbf{x}_2, \boldsymbol{\sigma}_2]$, are selected from the current population \mathcal{P}_t using the well-known roulette wheel method [10]. The global intermediary recombination operator, [10], produces the first offspring individual, $[\tilde{\mathbf{x}}_1, \tilde{\boldsymbol{\sigma}}_1]$, in such a way that

$$\tilde{x}_{1i} = \frac{x_{1i} + x_{2i}}{2}, \quad \tilde{\sigma}_{1i} = \frac{\sigma_{1i} + \sigma_{2i}}{2}, \quad (5)$$

for $i = 1, 2, \dots, n_t$. The uniform crossover operator, [10], produces the second offspring individual, $[\tilde{\mathbf{x}}_2, \tilde{\boldsymbol{\sigma}}_2]$, in such a way that the offspring chromosomes inherit each gene from a randomly chosen parent (drawn for each gene independently). Once created, the first offspring individual $[\tilde{\mathbf{x}}_1, \tilde{\boldsymbol{\sigma}}_1]$ undergoes mutation. First, mutation modifies the chromosome $\tilde{\boldsymbol{\sigma}}_1$ in the following way

$$\tilde{\sigma}_{1i} = \tilde{\sigma}_{1i} \cdot \exp(\varepsilon_i + \varepsilon_0), \quad (6)$$

for $i = 1, 2, \dots, n_t$, where ε_i is a real number generated randomly using the gaussian distribution (drawn for each gene independently) with mean 0 and variance τ^2 , where τ is the algorithm parameter, and ε_0 is a real number generated randomly using the gaussian distribution with mean 0 and variance τ_0^2 , where τ_0 is the algorithm parameter. Next, mutation modifies the chromosome $\tilde{\mathbf{x}}_1$, using the vector $\tilde{\boldsymbol{\sigma}}_1$ included in the offspring individual, as modified in (6), in the following way

$$\tilde{x}_{1i} = \tilde{x}_{1i} + \varepsilon_i, \quad (7)$$

for $i = 1, 2, \dots, n_t$, where ε_i is a real number generated randomly using the gaussian distribution (drawn for each gene independently) with mean 0 and variance $\tilde{\sigma}_{1i}^2$.

Next, the second offspring individual $[\tilde{x}_2, \tilde{\sigma}_2]$ undergoes mutation in the same manner. Population selection, based on the tournament selection, [10], focusses on randomly choosing a number of individuals from the union of the current and offspring populations, selecting the best of them to the new population and repeating the process N times until the adequate number of individuals is selected.

Once \mathcal{P}_{t+1} selected, EAIDR applies restoring mutation to it. It aims to extend the current search space \mathbb{R}^{n_t} and the current objective function G_t to the original ones (i.e. set $n_t = n$, Ψ_t being the identity function and $G_t = F$), transform the current population \mathcal{P}_{t+1} to the original search space \mathbb{R}^n and mutating, with a low probability, some candidate solutions in the original search space as in [7].

Finally, EAIDR applies PCA to the data sample defined by the new population \mathcal{P}_t : normalizes the data sample, computes eigenvalues and eigenvectors of its correlation matrix, selects the smallest n_t for which the sum of n_t largest eigenvalues exceeds $0.95 \cdot n_t$, defines the two linear mappings, Φ_t and Ψ_t , by eigenvectors corresponding to n_t largest eigenvalues ($\Psi_t = \Phi_t^T$) and G_t as a composition of Ψ_t and F . After transforming the new population \mathcal{P}_t to the new search space \mathbb{R}^{n_t} and evaluating it with the new objective function G_t , the evolution repeats until a termination condition is held (normally, after a certain number of iterations).

4 Experiments

In order to validate the proposed approach, a number of experiments were performed with the aim of comparing the original ESs to their extensions for objective functions with locally correlated variables. Experiments referred to a few benchmark functions widely applied in comparing and testing of various EAs [11]. The first and second benchmark functions, $f_1 : \mathbb{R}^k \rightarrow \mathbb{R}$ and $f_2 : \mathbb{R}^k \rightarrow \mathbb{R}$, are derived from the spherical model,

$$f_1(x_1, x_2, \dots, x_k) = \sum_{i=1}^k x_i^2, \quad f_2(x_1, x_2, \dots, x_k) = \sum_{i=1}^k \text{floor}(x_i + 0.5)^2. \quad (8)$$

The third benchmark function, $f_3 : \mathbb{R}^k \rightarrow \mathbb{R}$, is the Ackley’s function

$$f_3(x_1, x_2, \dots, x_k) = -c_1 \exp(-c_2 \sqrt{\frac{1}{k} \sum_{i=1}^k x_i^2}) - \exp(\frac{1}{k} \sum_{i=1}^k \cos(c_3 x_i)) + c_1 + e, \quad (9)$$

where $c_1 = 20$, $c_2 = 0.2$, $c_3 = 2\pi$. The last benchmark function, $f_4 : \mathbb{R}^k \rightarrow \mathbb{R}$, is the Fletcher-Powell’s function

$$f_4(x_1, x_2, \dots, x_k) = \sum_{i=1}^k (A_i - B_i)^2, \quad (10)$$

where

$$A_i = \sum_{j=1}^k (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j), \quad B_i = \sum_{j=1}^k (a_{ij} \sin x_j + b_{ij} \cos x_j), \quad (11)$$

and $a_{ij}, b_{ij} \in [-100, 100]$, $\alpha_j \in [-\pi, \pi]$ are chosen randomly.

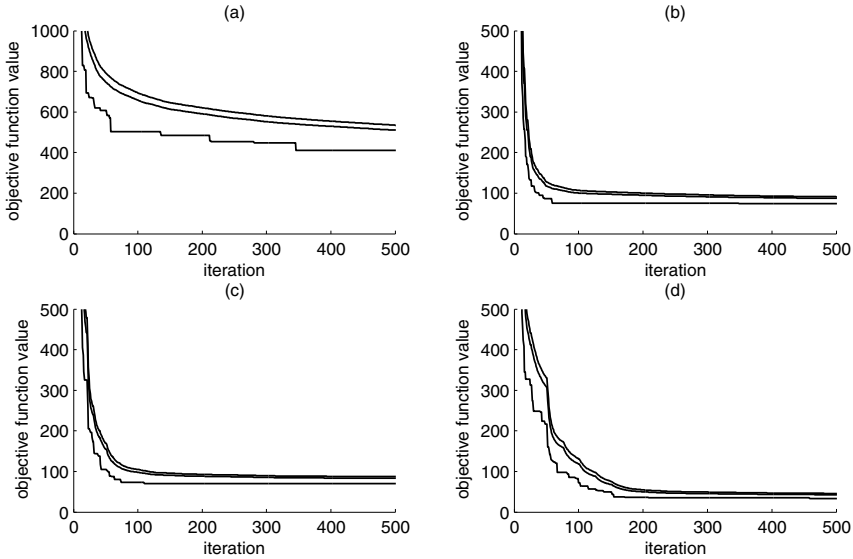


Fig. 2. Comparison of the classic ES (a) with the ESIDR (b) - (d) with dimensionality reduction run in each 5, 10 and 25 iterations, respectively, on one experiment concerning a 500-dimensional objective function F derived from the 125-dimensional sum of squares function f_1 (plots correspond to the lowest, the average and the highest value of the objective function in successive iterations for a typical case)

Each benchmark function was extended by a randomly chosen linear mapping $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^k$, for $k < n$, so that the final objective function $F : \mathbb{R}^n \rightarrow \mathbb{R}$ was a composition of Ψ and f_i in such a way that

$$F(x_1, x_2, \dots, x_n) = f_i(\Psi(x_1, x_2, \dots, x_n)), \quad (12)$$

for $i = 1, 2, 3$ or 4 . It is easy to see that variables of the final objective function F were correlated. Although the objective function F was formally a function of n variables, the real dimensionality of the optimization problem was $k < n$.

Populations of $N = 500$ individuals were used and offspring populations had the size $M = 2 \cdot N = 1000$. Mutation parameters were $\tau = 0.05$ and $\tau_0 = 0.01$. Tournament selection randomly chose 20 individuals and selected the best of them. Dimensionality reduction was run in each 5, 10 or 25 iterations (depending on the dimensionality of the original optimization problem and the expected number of iterations to reach optimal solutions), taking the previous dimension and the previous objective function in the other iterations.

Figure 2 presents a comparison of the classic ES with the ESIDR on one experiment concerning a 500-dimensional objective function F derived from the 125-dimensional sum of squares function f_1 : (a) refers to the classic ES, (b) - (d) refer to ESIDR with dimensionality reduction run in each 5, 10 and 25 iterations, respectively. Plots

Table 1. Comparison of the original ES and its extensions with internal dimensionality reduction

$i = 1$				$i = 3$			
n	k	ES	ESIDR	n	k	ES	ESIDR
20	10	33	21	20	10	38	26
40	20	49	36	40	20	59	41
80	40	77	54	80	40	86	62
250	50	-	1293	250	50	-	1937
250	125	-	2873	250	125	-	3091
500	125	-	3842	500	125	-	4820

$i = 2$				$i = 4$			
n	k	ES	ESIDR	n	k	ES	ESIDR
20	10	34	22	20	10	45	28
40	20	54	37	40	20	68	48
80	40	79	59	80	40	99	74
250	50	-	1384	250	50	-	2135
250	125	-	2947	250	125	-	3643
500	125	-	4013	500	125	-	5247

correspond to the lowest, the average and the highest value of the objective function in successive iterations for a typical case. It is easy to see that the ESIDR significantly outperformed the classic ES.

Table 1 presents a summary of results for four basis benchmark functions, f_1 , f_2 , f_3 and f_4 , transformed into the final objective function F by a randomly chosen linear mapping Ψ . For each configuration of k and n (6 different configurations were considered), 50 random linear mappings Ψ were drawn, leading to $4 \times 6 \times 50 = 1200$ different final objective functions F , and each optimization problem was solved independently 8 times by the classic ES and ESIDR. For each optimization problem, the average number of iterations to reach the optimal solution is reported.

It is easy to see that the proposed algorithm significantly outperformed the original one, leading to a large reduction in the number of iterations necessary to reach the optimal solution and consequently also in the computation time. It worth noticing that ESIDR succeeded in finding the optimal solution (with a certain accuracy) for all the optimization problems, while the classic ES was not able to find it for objective functions with larger number of variables.

5 Conclusions

Experiments performed to validate the proposed approach concerned some popular benchmark functions, such as the spherical model, the step function, the Ackley’s function and the Fletcher-Powell’s function [11]. The proposed algorithm, ESIDR, significantly outperformed the classic ES and dependency mining proposed in this paper succeeded in the detection of dependencies among correlated variables and enabled a significant reduction in the dimensionality of the optimization problem and consequently also in the computation time.

Further research on more complex models of dependencies, including non-linear dependencies, using more advanced methods, mainly generative approaches, such as the Mixtures of Probabilistic Principal Component Analyzers (MPPCA), the Bayesian Principal Component Analysis (BPCA), the Variational Principal Component Analysis (VPCA) or recent techniques of topographical mappings, may lead to the further improvement in the proposed approach.

References

1. Back, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York (1995)
2. Bishop, C., Svensen, M., Williams, C.: GTM: the Generative Topographic Mapping. *Neural Computation* 10, 215–234 (1998)
3. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Reading (1989)
4. Jolliffe, I.T.: *Principal Component Analysis*. Springer, Heidelberg (1986)
5. Korczak, J., Lipinski, P.: Evolutionary Building of Stock Trading Experts in a Real-Time System. In: *Proceedings of the 2004 Congress on Evolutionary Computation, CEC 2004*, Portland, USA, pp. 940–947 (2004)
6. Korczak, J., Lipinski, P., Roger, P.: Evolution Strategy in Portfolio Optimization. In: Collet, P., Fonlupt, C., Hao, J.-K., Lutton, E., Schoenauer, M. (eds.) *EA 2001*. LNCS, vol. 2310, pp. 156–167. Springer, Heidelberg (2002)
7. Lipinski, P.: Clustering of Large Number of Stock Market Trading Rules. In: *Proceedings of the 16th Symposium on Computational Statistics, CompStat 2004*, pp. 1397–1404. Springer, Heidelberg (2004)
8. Lipinski, P.: Dependency Mining in Large Sets of Stock Market Trading Rules. In: Pejas, J., Piegat, A. (eds.) *Enhanced Methods in Computer Security, Biometric and Intelligent Systems*, pp. 329–336. Kluwer Academic Publishers, Dordrecht (2005)
9. Larranaga, P., Lozano, J.A.: *Estimation of Distribution Algorithms*. Kluwer Academic Publishers, Dordrecht (2002)
10. Schwefel, H.-P.: *Evolution and Optimum Seeking*. John Wiley and Sons, Chichester (1995)

Neural Data Analysis and Reduction Using Improved Framework of Information-Preserving EMD

Zareen Mehboob and Hujun Yin

The University of Manchester
zareen.mehboob@postgrad.manchester.ac.uk,
hujun.yin@manchester.ac.uk

Abstract. This paper presents several improvements to the framework of information-preserving empirical mode decomposition (EMD). The basic framework was presented in our previous work [1]. The method decomposes a non-stationary neural response into a number of oscillatory modes varying in information content. After the spectral information analysis only few modes, taking part in stimulus coding, are retrieved for further analysis. The improvements and enhancement have been proposed for the steps involved in information quantification and modes extraction. An investigation has also been carried out for compression of retrieved informative modes of the neural signal in order to achieve a lower bit rate using the proposed framework. Experimental results are presented.

1 Introduction

Continuous neural recordings of electric and magnetic field potentials, such as LFP (local field potentials), EEG (electrocardiographs), and MEG (magnetoencephalographs), are usually analyzed in different frequency bands to understand a brain state or activity of neuronal populations in that area. In the previous work, we presented a method based on the empirical mode decomposition (EMD) to extract the oscillations taking part in stimulus coding, i.e. the informative intrinsic mode functions. EMD offers many advantages over many existing non-linear, non-stationary signal processing methods [2]. Although it is considered as a form of wavelet decomposition [3], the basis functions of the EMD are derived from the data itself, thus making it adaptive and data-driven and suitable for analysis of neural responses. The details and advantages of the method can be found in [2-4].

This paper suggests a few steps to improve the modes information quantification and extraction process. The proposed modifications and their effect on the results are presented in the following section. An initial study has been conducted to use the retrieved informative oscillations for data compression. The compression algorithm and results are presented and discussed in section 3, followed by the conclusion.

2 Information Preserving Empirical Mode Decomposition (EMD)

As a data-driven method, EMD [4] can adaptively decompose a nonlinear neural response into a number of zero-mean AM/FM functions called intrinsic mode functions (IMFs). It considers signal oscillations at local level and can reveal important temporal information that is not achievable by using the Fourier and wavelet transforms [4]. In our previous work, it was shown that extracted IMFs have varying information content and not all extracted IMFs are information carriers. In the initial framework [1], an information coupled EMD was proposed to quantify the information content in original recordings and their corresponding IMFs and to retrieve only a small number of IMFs that are taking part in stimulus coding.

2.1 Extraction of Informative Modes

Several modifications are proposed on the previous framework [1]. The experiments on various data sets have confirmed that the suggested modifications make the information estimation more accurate and improve the modes extraction process. The steps involved in the enhanced extraction framework are summarized in Fig. 1. The steps can be numbered according to the flow from step 1 to step 5.

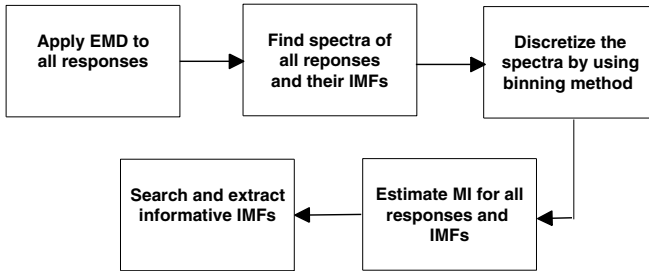


Fig. 1. Steps for the extraction of informative modes using EMD

In the first step, neural responses are decomposed by using the EMD. The second step involves the spectral analysis of the responses and their subsequent modes using the multitaper method (MTM) [5].

The third block in Fig. 1 involves the discretization of spectra, which are then used in step 4 for probability estimation of mutual information (MI) and analysis of stimuli and responses. The last step consists of a filtering process for retrieving information carrying modes. The proposed improvements are for steps 3 and 5.

In most experiments designed for neuronal population analysis, the neural code and information analysis is usually carried out in the frequency domain [1, 6, 7]. In order to determine how well the power (pwr) in a response (R), at a certain frequency (f), encodes the stimulus features, the mutual information

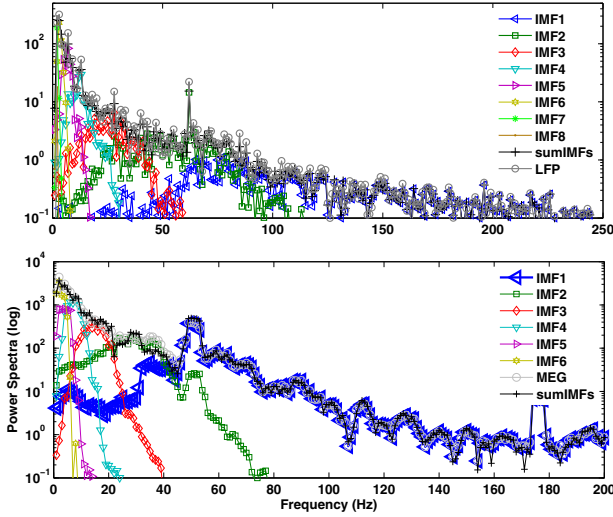


Fig. 2. Spectral distribution of an LFP (top) and MEG and their corresponding IMFs. The black like with + marker is the spectral sum of all IMFs which is equivalent to the spectra of original recording.

$MI(S; R_{pwr}(f))$ is calculated between the stimuli (s) and the power (R_{pwr}) at frequency (f). This is given by:

$$MI(S; R_{pwr}(f)) = \sum_s P(s) \sum_{R_{pwr}(f)} P(R_{pwr}(f)|s) \log_2 \left(\frac{P(R_{pwr}(f)|s)}{P(R_{pwr}(f))} \right) \quad (1)$$

where $P(s)$ is the probability of the stimulus or stimulus window s . $P(s)$ is equal to the inverse of the total number of stimuli if the stimuli are discrete, otherwise the total number of stimulus windows in case of continuous stimuli recordings. $P(R_{pwr}(f)|s)$ is the probability of observing a power $R_{pwr}(f)$ at frequency f , in response to a single trial to stimulus s and $P(R_{pwr}(f))$ is probability of power $R_{pwr}(f)$ across all trials in response to any stimulus. $MI(S; R_{pwr}(f))$ quantifies the reduction of the uncertainty about the stimulus that can be gained from observing the power at frequency f in one trial. For base 2 logarithms, $MI(S; R_{pwr}(f))$ is expressed in units of bits. 1 bit of information means that observation of the neuronal response in one trial reduces the observer's stimulus uncertainty by a factor of 2.

The MI calculation is based on the probability distribution of $R_{pwr}(f)$ across all the stimuli and trials. The probability estimate is carried out by means of a discretization method known as binning [8]. The binning method is of particular importance since the MI estimation is affected by it and the results may be subject to biasing due to the errors in probability estimation [9]. More information about binning strategies can be found in [8-10]. The spectra from an LFP and MEG recording are shown in Fig. 2.

As shown in figure, the spectra of subsequent IMFs lie in different frequency bands but their total sum is equal to the spectra of the original recording R . We represent the MI of the original recording R by MI_R and the MI of its decomposed IMFs by $MI_IMF_n, n = 1, 2, \dots, N$. It has been shown previously [1] that if N IMFs are extracted from a recording R (Eq. 2) then each IMF varies in information content and only few IMFs need to be retrieved whose summed information will be roughly equivalent to the information contained in the original recording/s R (Eq. 2).

$$R(t) = \sum_{n=1}^N IMF_j(t) + r_N(t)$$

where $r_N(t)$ is the residual which gives the trend of signal.

$$MI_bestIMFs \approx MI_R \tag{2}$$

However in the previously proposed algorithm, for MI estimation, the binning was applied individually to spectra of each IMF and in few cases can result in biased MI estimation. This can be seen clearly in the top plots of Fig. 3 shown by green dotted line where MI of the extracted IMFs exceeds the MI of the original response. From further analysis, it is found that the biasing can be effectively minimized by applying binning across spectra of all IMFs and then estimating the MI for each IMF individually. The bottom two plots in Fig. 3 are the new MI estimations for the same set of LFP data and they clearly show that the information from the extracted IMFs has matched that of the original LFP and the bias has been significantly reduced.

Further improvement has also been made to step 5 of the framework, Fig. 1. In the previously proposed method, the filtration of informative IMFs was based on the cross correlation measure between the MI_R and $MI_IMF_n, n = 1, 2, \dots, N$. Cross correlation is suitable for finding linear similarity among two time series. Due to this reason, It was also observed previously [1] that only one or two IMFs were retrieved in most cases which lacked in information preservation and it was anticipated that using a nonlinear similarity measure could improve the extraction procedure and result in extraction of more informative IMFs which may otherwise be missed by using the correlation.

For this reason, we propose to replace the linear similarity measure with a nonlinear, more localized similarity measure called the cross correlation entropy measure or CorrEntropy (CorrE) [11, 12]. CorrE is a localized similarity measure based on information theoretic learning (ITL). It finds the similarity between two random processes [12] and can detect nonlinear characteristics in a given time series. Our experiments showed that using the CorrE, significantly improved the informative IMF extraction process. Examples of the results are shown in Table 2. The CorrE between two random variables can be found by:

$$V(X, Y) = \frac{1}{L} \sum_{k=1}^L G(x_k - y_k) \tag{3}$$

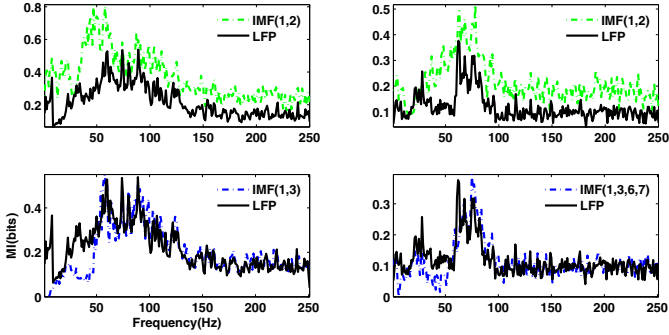


Fig. 3. The comparison of MI estimation by binning spectra of each IMF separately (top) and by binning spectra of all IMs collectively and then estimating MI for individual IMFs (bottom)

where G is the Gaussian kernel of size σ and $\sigma = 0.9 \times \min\left(AL^{-\frac{1}{5}}\right)$. The kernel width works as a window to find the similarity in the given area. L is the data length and A stands for the minimum of the empirical standard deviation of data and the data interquartile range scaled by 1.34 as defined in Silverman's rule. The details of the algorithm are given in Table 1. The algorithm has been tested on several LFP, EEG and MEG datasets. Some of the results obtained by using the previous approach and the improved approach are shown in Fig. 3 and Table 2 for a comparison. The first column of the table shows the IMFs extracted using the previous method and the third column shows the IMFs extracted using the enhanced framework presented in Table 1. The second column shows the initial correlation between the MI_R and MI of first extracted IMF ($MI_{bestIMF_1}$) and the final correlation between MI_R and all the extracted IMFs ($MI_{bestIMFs}$). The fourth column shows the initial and final CorrE between the MI_R and $MI_{bestIMFs}$. The MI correlation and MI correntropy plots for the first two rows are shown in Fig. 3.

3 Data Compression Using Informative Modes

The storage and analysis of neural data recordings is not a trivial task as the volume of neural data recordings are often huge and the experiments are usually carried out for different stimuli and repeated several (10-100) times. The archival storage and exchange of these datasets requires an efficient compression algorithm. The criteria for the compression algorithm designs of neural data recordings would be based on discarding the data points that are not carrying any information or contributing to stimulus coding so that the information loss in the compressed dataset will be minimal or negligible.

In order to address these considerations and achieve a low bit rate compression for neural recordings, we have investigated an approach that can make use of the information preserving EMD and a recently proposed EMD based audio

Table 1. Algorithm for extraction of informative modes from neural signals

-
1. Apply EMD to all the responses/ recordings in a given dataset. Fix the stopping criteria of sifting process and extract N IMFs for each trial. The number of N depends on the length of recoding.
 2. Divide the original recording R into stimulus windows for continuous stimulus case. For discrete case go to step 5.
 3. For continuous stimulus recordings, divide each IMF into suitable stimulus windows. For discrete case go to step 5.
 4. Calculate power spectrum density of all responses R using the multitaper method.
 5. Calculate power spectrum density of all IMFs using MTM.
 6. Using the binning method, discretize the power spectra of all responses for each frequency into equi-spaced bins and calculate MI for the responses using Eq. 1, store it as MI_R .
 7. Take the power spectra of all IMFs and discretize the spectra at each frequency into equi-spaced bins. Calculate MI for each IMF using Eq. 1. This gives $MI_IMF_n, n = 1, 2, \dots, N$.
 8. Take each $MI_IMF_n, (n = 1, \dots, N)$ and compare its CorrE between MI of that IMF with MI_R by using Eq. 3.
 9. Choose the best informative IMF that has the maximum MI CorrE with the response R and store it to a set of $MI_bestIMF$.
 10. Choose the next informative IMF by using the following step:
 Take each of the remaining MI_IMF one by one and add it to the MI of selected IMFs and compute the CorrE between their MI and MI_R . If the resultant CorrE is greater than the previous value+0.05, it means that this IMF is contributing significant amount of information. Choose this IMF as the next best IMF and update the collection of best IMFs ($MI_bestIMF$). Otherwise quit.
-

Table 2. Comparison of the results obtained from previous proposed framework and after implementing the suggested modifications

IMF extraction using correlation [1]		IMF extraction using algorithm in Table 1	
IMFs	MI Correlation	IMFs	MI CorrE
1,2	0.69-0.76	1,3	8.08-8.91
1,2	0.65-0.73	1,3,6,7	10.04-11.97
1,5	0.56-0.64	1,3	6.43-7.48
1,4	0.64-0.74	1,3,4,5,6	9.75-12.03
1	0.79	1,3,4	9.44-11.56

compression [13]. The idea is to apply compression on extracted informative IMFs only. The compression algorithm is based on following simple steps:

1. Extract the IMFs for the given set of recordings.
2. Find the information preserving IMFs using algorithm given in Table 1.
3. Find the extrema for all the extracted IMFs.
4. Quantize and encode the extrema.

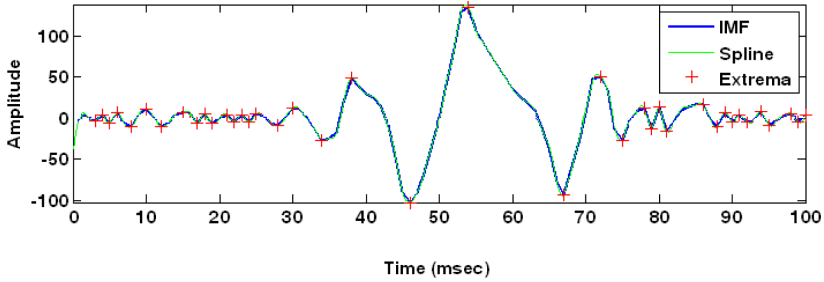


Fig. 4. An original IMF and its reconstruction by spline interpolation

The compressed IMFs and recordings can be reconstructed by interpolating IMFs and summing them. We used the Huffman coding for encoding purpose. With the spline interpolation of extrema of an IMF, a perfect reconstruction can be obtained (e.g. Fig. 4) so in case a user decides to use all the IMFs for compression then it will be a lossless coding, it would be similar if the user wants to compress only the information carrying IMFs. The algorithm can be improved by introducing a thresholding step for reducing the number of extrema in each IMF. But consideration should be made to observe the effect in the information level. In the initial study we have tested the approach by using all the extrema in compression. Using the above approach, a compression ratio of 0.18 or lower has been achieved. It is anticipated that further investigation could yield to more promising results that could help in handling and standardizing compression techniques for large neural datasets.

4 Conclusion

This study proposes several advances that can improve the EMD-based extraction procedure of information carrying modes from the continuous neural field potentials. The modifications are shown to help obtain more precise or less biased information quantification. The initial analysis of the approach adopted for compression has also shown some promising results. Further investigation will be conducted to achieve better compression rates with minimal information loss.

References

1. Mehboob, Z., Yin, H.: Information preserving empirical mode decomposition for filtering field potentials. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 226–233. Springer, Heidelberg (2009)
2. Rilling, G., Flandrin, P., Goncalves, P.: On empirical mode decomposition and its algorithms (2003), <http://perso.ens-lyon.fr/patrick.flandrin/nsip03.pdf>
3. Flandrin, P., Rilling, G., Goncalves, P.: Empirical mode decomposition as a filter bank. IEEE Sig. Proc. Lett, 112–114 (2004)

4. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. Royal Soc.* 454, 903–995 (1998)
5. Percival, D.B., Walden, A.T.: *Spectral analysis for physical applications: multi-tapper and conventional univariate techniques*. Cambridge University Press, Cambridge (1993)
6. Mandic, D.P., Souretis, G., Leong, W.Y., Looney, D., Van Hulle, M.M., Tanaka, T.: Complex empirical mode decomposition for multichannel information fusion. In: *Signal Processing Techniques for Knowledge Extraction and Information Fusion*, pp. 243–260. Springer, Heidelberg (2008)
7. Hämäläinen, M., Hari, R., Ilmoniemi, R.J., Knuutila, J., Lounasmaa, O.V.: Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Rev. Mod. Phys.* 65(2), 413–497 (1993)
8. Belitski, A., Gretton, A., Magri, C., Murayama, Y., Montemurro, M., Logothetis, N., Panzeri, S.: Low frequency local field potentials and spikes in primary visual cortex convey independent visual information. *J. Neurosci.* 28(22), 5696–5709 (2008)
9. Rodrigo, Q.Q., Panzeri, S.: Extracting information from neuronal populations: information theory and decoding approaches. *Nat. Rev. Neurosci.* 10(3), 173–185 (2009)
10. Magri, C., Whittingstall, K., Singh, V., Logothetis, N., Panzeri, S.: A toolbox for the fast information analysis of multiple-site lfp, eeg and spike train recordings. *BMC Neuroscience* 10(81), 1–24 (2009)
11. Gunduz, A., Principe, J.C.: Correntropy as a novel measure for nonlinearity tests. *Signal Processing* 89, 14–23 (2009)
12. Liu, W., Pokharel, P.P., Principe, J.C.: Correntropy: A localized similarity measure. In: *IJCNN*, pp. 4919–4924 (2006)
13. Khaldi, K., Boudraa, A.O., Turk, M., Chonave, T., Samaali, I.: Audio encoding based on empirical mode decomposition. In: *EUSIPCO*, pp. 924–928 (2009)

Improving the Performance of the Truncated Fourier Series Least Squares (TFSL) Power System Load Model Using an Artificial Neural Network Paradigm

Shonique L. Miller, Gary L. Lebby, and Ali R. Osareh*

Green Renewable Energy and Advanced Technology Transfer
North Carolina Agricultural and Technical State University
1601 East Market Street, Greensboro, North Carolina, USA
{s1mille2,lebby,osareh}@ncat.edu
<http://www.mitss.ncat.edu>

Abstract. Power System Load models have a wide range of application in the electric power industry including applications involving: (i) load management policy monitoring; (ii) assisting with the generator commitment problem; (iii) providing short term forecasts; (iv) aiding with system planning by providing long term forecasts. A method that has been utilized in the power systems planning community involves modeling the power system load (PSL) utilizing a truncated Fourier series. Presented herein is an innovative method based upon analyzing nine weeks of data and generating an optimum number of Fourier series terms included in model structure from a set of preselected heuristic basis functions for prediction. The resulting PSL model capable of providing high quality middle-long term forecasts and retain the shape prediction of the load curve out in time.

Keywords: load models, neural network, shape modeling.

1 Introduction

Public electric utilities must always be prepared to serve the peak demand load reliably and must remain knowledgeable of the minimum demand load so as to coordinate maintenance or energy storage procedures. This important feature of any public electric utility is a major reason as to why power system planning, modeling and forecasting is a critical step in strategic planning for the utility [1]. Human behavior and temperature of the region along with the type of load are some of the major factors inherent in the shape and pattern of any power system load. The objective of most companies would be to ensure it has

* Thanks to Dominion Power of Virginia and ULTURNAGEN, LLC of North Carolina for supporting the work of the Institute for Green Renewable Energy and Advanced Technology Transfer (GREATT) at North Carolina Agricultural and Technical State University.

the ability to continually provide its services to its customers. Strategic planning must be done to ensure that the power industry maintains its ability to provide secure and reliable services [1]. Consequently, study of the growth and behavior of the load becomes imperative and this study must be accurate and concise.

Artificial Neural Networks are neurologically inspired systems consisting of highly interconnected elementary computational units (neurons) [2]. Studies have shown that Artificial Neural Networks (ANN) have been instrumental in modeling power system loads. Their ability to model multivariate problems without making complex dependency assumptions among input variables are unmatched and they do not rely on human experience, but attempt to draw links between sets of input data and observed outputs. Despite their ability and robustness, there are several issues that can not be addressed by ANN's alone. Some of these include: (i) modeling load deviations caused by effects of weather, (ii) prediction of special days such as Mondays, Fridays and holidays, (iii) incorporation of operator knowledge in the forecasting model, (iv) failure to capture all information during training [3].

It can be concluded that the ANN's failure to capture all information during training diminishes the accuracy of the PSL model and forecast developed. It is in this regard we propose that if the peak and valley data can be captured and modeled separately and then this data inserted at corresponding points within the model, the model would reflect the actual load data more accurately and the mean absolute percentage error would be reduced, which will improve the performance of the ANN.

The modeling architecture used in this document is the Optimal Linear Associative Memory (OLAM) and it will be used to model the weekly average power system load provided by the Randolph Electric Membership Corporate,¹ between 1991 and 1995.

2 The Truncated Fourier Series Least Squares PSL Model

The characteristics used to model the power system load in this study are determined by plotting and observing obvious trends in the historical average weekly load data over a period of approximately four years. There are noticeable characteristics such the winter heating peaks and summer cooling peaks occurring at approximately six-month intervals. In electric power utilities located in the southeastern part of the United States, the peaks are due to the winter heating load during the cold months and the summer cooling load during the hot months [4]. This phenomena occurs in every year of figure 1, thus it is labeled the seasonal load effect. Traditionally, the seasonal weather load is modeled as a truncated Fourier series [5]. The load model used in this document is given as follows:

$$PSL_w = Base + Growth * w + Seasonal_w \quad (1)$$

¹ Randolph Electric Membership Corporate is an Electric Power Cooperate operating in the area surrounding Asheboro, North Carolina.

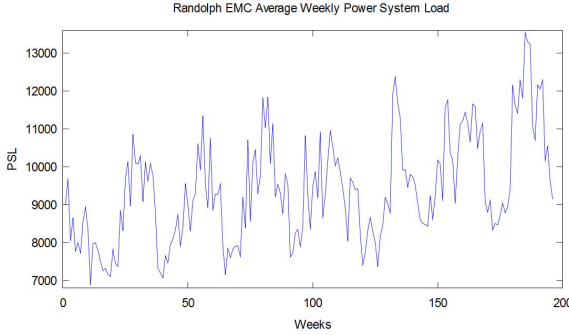


Fig. 1. Untouched Average Weekly Power System Load

$$Seasonal_w = \sum_{i=1}^N A(i)\cos(\omega_0 * i * w) + B(i)\sin(\omega_0 * i * w) \tag{2}$$

$$\omega_0 = \frac{\pi}{26} \tag{3}$$

(We will assume that the base and growth are linear functions of the week variable (w).)

The direct application of solving the normal equations for least square problems become cumbersome as the number of unknown constants in a problem grows. Generally we can express our problem as:

$$y = w_0x_0 + w_1x_1 + \dots + w_nx_n \tag{4}$$

If we are looking over many discrete events, T , and we assume that x_0 is defined to be one we can write as:

$$\tilde{y} = \tilde{X} \bullet \tilde{w} \tag{5}$$

The above expression of our general problem is exactly the mathematical architecture of the OLAM and thus the truncated Fourier series least squares modeling technique can be approximated by the OLAM.

3 Optimal Linear Associative Memory (OLAM)

In 1984 professors Teuvo Kohonen and Mikko J Ruohonen developed the OLAM based upon earlier work in 1973 on correlation matrix memories. The weights of the OLAM guarantee perfect retrieval of stored memories given that the columns of both the x and y fields are linearly independent. It is assumed that the columns of X and Y are both linearly independent. Both the X and Y fields are separated into training set and test set. If this is the case then the forward OLAM weights may be calculated as [6]:

$$\hat{w} = (x\tilde{trn}^T x\tilde{trn})^{-1} x\tilde{trn}^T * \tilde{Y} \tag{6}$$

For this particular model, the output \tilde{Y} is replaced with the power system load training data set as shown in equation (2).

$$\hat{w} = (x\tilde{r}n^T x\tilde{r}n)^{-1}x\tilde{r}n^T * PSL_w \tag{7}$$

We will assume that the base and growth are linear functions of the week variable (w) and the power system model used is shown in equation (1).

4 Hartley Test

The extent to which the Fourier series will be truncated (i.e. the value of N) is to be determined by the significance of the regression coefficients. We will initially begin with the number of harmonics being seven (N=7 (chosen randomly)). From a statistical point of view, the computed Fourier coefficients are merely estimates of the true values which would have been obtained had there been no contamination by noise factors. The presence of noise (and this is true in any real data) implies that the computed Fourier coefficients will often not equal zero when coefficients of the underlying signal is actually zero. For example, a square wave in sine phase has only odd harmonic components but when noise is added to the square wave the computed Fourier coefficients will not be zero for the even harmonics. To create a faithful model, it may become necessary to omit coefficients that may arise from the effects of noise.

In performing a Fourier analysis (the constant term has been absorbed by the base term, and the base and growth terms have been factored out of the current analysis) a random variable of interest is the power in the k^{th} harmonic, which is one-half the square of the (polar) amplitude as follows[7]:

$$p_k = \frac{1}{2}(A_k^2 + B_k^2) \tag{8}$$

The A_k and B_k coefficients can be shown to have Gaussian characteristics and from probability theory we know that if a random variable is Gaussian then the square of that random variable is distributed as a chi-squared random variable. The Hartley Statistic, invented by Ralph Vinton Lyon Hartley, is a method for testing the significance of power in the k^{th} harmonic by assuming all of the other harmonics are noise.[7]:

$$H = \frac{p_k}{\frac{1}{R} \sum_{j=1}^R p_j} \sim F(2, 2R) \tag{9}$$

H is F-distributed with two degrees of freedom in the numerator and $2R$ degrees of freedom in the denominator. The test of whether the coefficients (taken in pairs) are significant, boils down to determining that the signal power in the k^{th} harmonic is zero if H is smaller than the tabulated value of $F(2, 2R)$ [7]. If H is larger than the tabulated value, reject the null hypothesis that the signal power in this harmonic is zero. R was chosen to be 7 and from the F-table, $F(2, 2R)$ is 3.74 for a 95 percent confidence integral.

5 Goodness of Fit: R^2 Test

The R^2 statistic gives information concerning the goodness of fit of a model. In regression, the R^2 coefficient of determination is a statistical measure of how well the regression line approximates the real data points. An R^2 of 1.0 indicates that the regression line perfectly fits the data. R^2 is the percentage of variation in the actual data that is explained by your model. R^2 is calculated by:

$$R^2 = 1 - \frac{\sum_{w=1}^N (PSL_w - P\hat{S}L_w)^2}{\sum_{w=1}^N (PSL_w - P\bar{S}L)^2} \quad (10)$$

6 Durbin-Watson Test

After the power system load was modeled, a test on the residuals had to be done to determine whether they were random noise. The residuals must be uncorrelated (random) and the Durbin-Watson Test will be used to determine this. The residuals have to follow a normal distribution and exhibit homoscedasticity (equal variances). Usually the results of a regression will give residuals that are mean zero. Sometimes this simple test will uncover the fact of erroneous modeling and programming upfront. The Durbin-Watson (DW) statistic (named after James Durbin and Geoffrey Watson) is used to detect the presence of autocorrelation in the residuals resulting from a regression analysis, and is calculated by [7]:

$$DW = \frac{\sum_{w=2}^N (R_w - R_{(w-1)})^2}{\sum_{w=1}^N R_w^2} \quad (11)$$

If DW is approximately 2, then this is an indication that no auto correlation exists. If DW is substantially less than 2, there is evidence of positive serial correlation. There is a reason for concern if DW is less than 1.0, there may be cause for alarm. Small DW indicates successive error terms are, on average, close in value to one another (i.e. positively correlated.) Large DW indicates successive error terms are, on average negatively correlated.

7 Homoscedasticity

In regression analysis, homoscedasticity means a situation in which the variance of the dependent variable is the same for all the data (aka equal variances). We test for equal variances by using the F-test. In order to perform this test, F_{STAT} was determined by [7]:

$$F_{STAT} = \frac{\text{largesamplevariance}}{\text{smallsamplevariance}} \quad (12)$$

$$F_{STAT} = \frac{\hat{\sigma}_1^2}{\hat{\sigma}_2^2} \quad (13)$$

The calculated F_{STAT} is then compared to a F_{CRIT} .

8 PSL Modeling Using the OLAM

Using the fore-mentioned model, we generated an input vector (X) consisting of 196 rows and 16 columns, with column 1 representing the bias, column 2 the growth variable and columns 3 – 16 7 successive harmonics of the Fourier series. The output vector Y will consist of the 196 average weekly original Randolph data.

The Hartley Test was performed on the 7 harmonics of the Fourier Series and it was observed that only the first harmonic is significantly different from zero with a calculated H value of 3.75 which is greater than the F-table delimiter of 3.74. The Fourier series will then be truncated to this number of harmonics, $N = 1$. This determination justifies the winter cooling and summer peaks observed each year in the southeastern region.

Now that the Fourier series has been truncated, the model can be inserted into the OLAM. The resulting model is shown in the figure 2.

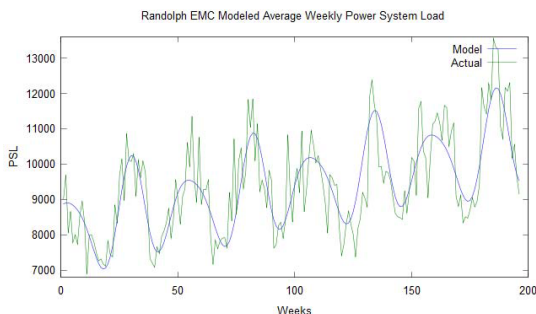


Fig. 2. Model of Average Weekly Power System Load using OLAM

The Durbin-Watson test was done on the entire model and raw power system load. When calculated, the Durbin-Watson test was 1.4439. This value although closer to 2 than 1 still needs to be investigated as to whether or not it is safe to say the residuals are uncorrelated.

The calculated F_{STAT} is then compared to a F_{CRIT} . From the f-table, this value was found to be 2.784. For 1000 iterations, random large and small samples were retrieved from the residuals and F_{STAT} calculated. Out of the 1000 iterations, 964 were less than F_{CRIT} and 32 were greater. It can be concluded that the residuals have equal variances. The mean absolute error for this model is 2464 with the mean absolute percentage error associated with the model and actual load is 8.16 percent. The goodness of fit of the model (R^2) was determined to be 0.5116. This value shows that although much of the variance of the load data has been explained by the model, there is still some variance within the parameters that have not been captured by the model, hence our model's fit to the original data can be improved.

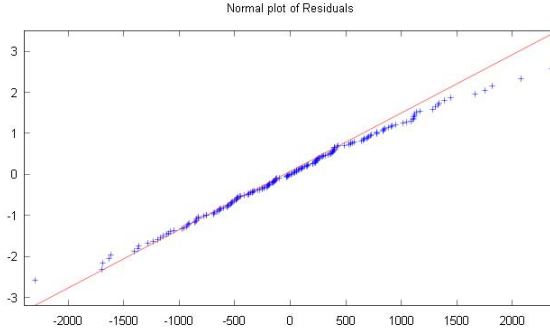


Fig. 3. Normal plot of the Residuals

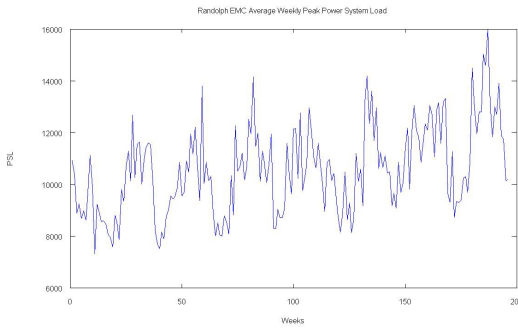


Fig. 4. Untouched Average Weekly Peak Power System Load

9 Improving the Performance of the OLAM

As stated earlier, information can be lost during the training process of ANN's. Looking at figure (3), it is observed that the model is not accurately reflecting the peak and valley load data, thus some key data about the peak and valley load of the PSL may have been omitted from the model that could lead to erroneous forecasting and inadequate futuristic preparation of a power utility.

It is in this regard that it is proposed that if the peak and valley data are modeled separately, knowing the period of the data, the original model can be "turned off" as it peaks and troughs and the modeled peak and valley data can be inserted into the corresponding points in the original model, the model error would decrease and the R^2 term improved.

The peak and valley load were extracted from the original load data by simply finding the maximum and minimum weekly load as provided by the Randolph Corporation historical data and are shown in figures 4 and 5.

The peak and valley load data are of similar characteristic to the original load data and thus a similar OLAM model can be used to model them respectively as shown in figures 6 and 7.

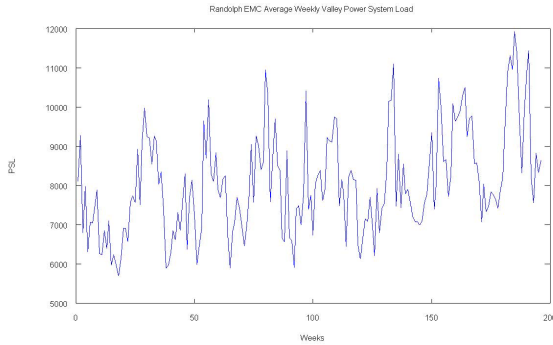


Fig. 5. Untouched Average Weekly Valley Power System Load

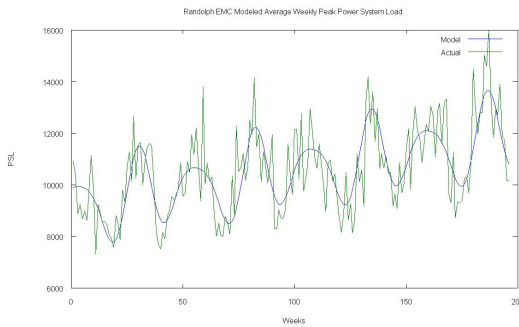


Fig. 6. Model Average Weekly Peak Power System Load

Now that the peak and valley load have been modeled, the original average weekly model was then turned off when the original model reached a peak point and when it reached a valley point. When looking at the model load, it seen that within one period there is one peak and one valley. The period of the model is:

$$T = \frac{52}{2\pi} \tag{14}$$

Knowing this the model was turned off at a $\frac{1}{4}$ of each period and then again at $\frac{3}{4}$ and the peak and valley model parameters were inserted at these points respectively. The resulting model is shown in figure 8 and before any statistic measurements are performed, by observation it is seen that the model’s trace of the power system load has improved exceptionally as compared to figure 2.

The Durbin-Watson test was calculated to be 1.5421 an increase from the precious value and the normal plot of the residuals are, displayed figure 9 demonstrate that the residuals from the improved model are even more normal than previously observed.

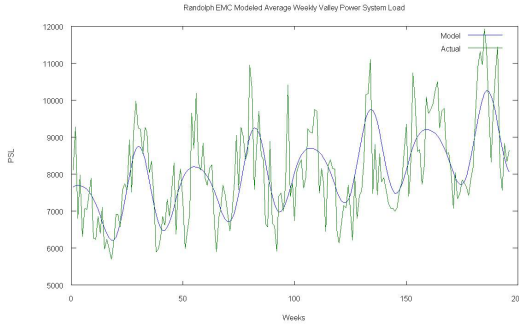


Fig. 7. Model Average Weekly Valley Power System Load

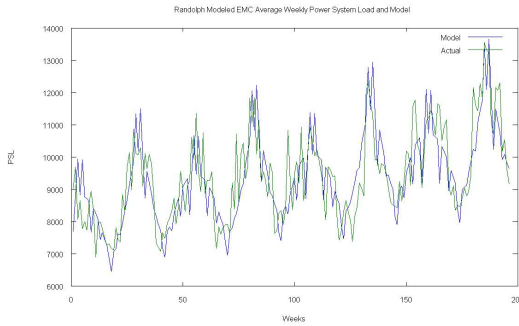


Fig. 8. Improved Model Average Weekly Power System Load with peak and valley load model parameters added

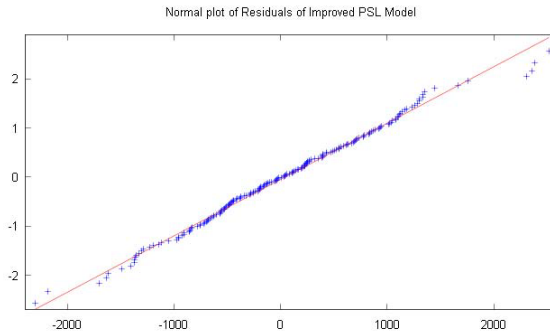


Fig. 9. Normal plot of the Residuals

The calculated F_{STAT} is then compared to a F_{CRIT} . From the f-table, this value was found to be 2.784. For 1000 iterations, random large and small samples were retrieved from the residuals and F_{STAT} calculated. Out of the 1000 iterations, 968 were less than F_{CRIT} and 32 were greater. Again it can be generalized that that the residuals have equal variances.

The mean absolute error for this model decreased to 605.06 with the mean absolute percentage error associated with the model and actual load decreasing to 6.41 percent. The goodness of fit of the model (R^2) increased to 0.699. This value shows that the injecting of modeled peak and valley information has increased the goodness of fit of the model data.

10 Conclusions

In conclusion, ANN's are excellent modeling tools. Their ability to model multivariate problems without making complex dependency assumptions among input variables are unmatched and they do not rely on human experience, but attempt to draw links between sets of input data and observed outputs prove them to be very robust tools in load modeling. However, there are several issues that can not be addressed by ANN alone. Some of these include: (i) modeling load deviations caused by effects of weather, (ii) prediction of special days such as Mondays, Fridays and holidays, (iii) incorporation of operator knowledge in the forecasting model, (iv) information can be lost during training [3]. Due to these disadvantages tools and ideas must be implemented to improve the performance of the ANN's. It was observed that The OLAM, just one type of ANN, was able to model the Randolph powers system load data with a fair degree of accuracy and the mean absolute percent error of 8.16 percent error with a goodness of fit measurement of 0.5116. When observing the overlay of the model to the original data it was observed that the peaks and valleys were not sufficiently captured by the model. It was then proposed that if the peak and valley data were modeled separately and then inserted at the corresponding points in the model, this will decrease model error and improve the goodness of fit. The results show that this added technique alone has improved the performance of the model with the mean absolute percentage error decreasing to 6.41 percent and the goodness of fit increasing to 0.699. Further consideration and enhancement of the model is obtained by modeling the residuals of the model and actual load data and these values also added to the model parameters. These results are suggestive that the performance of ANN's in power system load modeling can be improved and enhanced without any over generalization of data or underlying phenomena of the nature of the load.

References

1. Lebby, G., Stevenson, K., Shi, G.: Power system load modeling using a RBFGRNN with self starting centers. In: PowerCon: Blackout. IEEE, Los Alamitos (2003)
2. Khotanzad, A.: On-line load forecasting services (2007)
3. ul Asar, A., ul Hassnain, S.R., Khattack, A.U.: A multi-agent approach to short term load forecasting problem. International Journal of Intelligent Control And Systems (2005)

4. Darmand, A.B., Leiby, G.L., Williams, F.R., Stevenson, K.M., LaPrade, J.A.C.: Power system load characterization of a southern electric power cooperate using neural networks and statistical methods. In: International Conference on Power and Energy Systems, Iasted Europes (2001)
5. Leiby, G.L.: Load shape modeling in southeastern utility systems. Ph.D. dissertation, Clemson University (1985)
6. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River (1999)
7. Leiby, G.L.: Modelling of the power system load using the least squares method. Power System Planning Class Notes (2009)

An Efficient Approach to Clustering Real-Estate Listings

Maciej Grzenda¹ and Deepak Thukral²

¹ Warsaw University of Technology,
Faculty of Mathematics and Information Science,
00-661 Warszawa, Pl. Politechniki 1, Poland

M.Grzenda@mini.pw.edu.pl

² TESOBÉ Music Pictures Ltd.
Osloer Strasse 16/17, D-13359 Berlin, Germany
deepak@tesobe.com

Abstract. World Wide Web (WWW) is a vast source of information, the problem of information overload is more acute than ever. Due to noise in WWW, it is becoming hard to find usable information. Real-estate listings are frequently available through different real estate agencies and published on different web sites. As a consequence, differences in price and description can also be observed. At the same time, a potential buyer or renter may prefer to get the entire description of a property of interest based on the data available on different portals and if possible track the changes in price. This problem can be considered as an illustration of a wider class of problems with integrating the data from numerous semi-structured web data sources.

The paper investigates the way clustering algorithms can be used to identify individual real estate properties described on different portals. Clustering algorithms have been used to group the records acquired from different web sources. Both standard clustering methods have been evaluated, and a method using new distance function combining similarity of semi-structured and unstructured data has been proposed. The latter approach has allowed substantial improvement in clustering results.

1 Introduction

The huge amounts of information available on the Internet have lead to extensive research on information extraction techniques. Most of the algorithms concentrate on keyword-based searching. At the same time, a growing number of automatically generated web pages is observed. In the latter case, the content of such web pages follows templates that are filled in with the data typically coming from database management systems. As a consequence, such web resources can be considered as semi-structured data sources [1-4]. In a growing number of cases, these data sources describe partly the same subject matter. Examples include, but are not limited to books, IT products, real-estate listings or hotels. Not surprisingly, an increasing demand for uniform methods of accessing the

data available on the Internet, regardless of the differences in data structures and content of individual data sources is observed [2].

The primary objective of our work is to analyze the way that data captured from semi-structured web data sources can be integrated, taking into account the lack of unique identifiers and incompleteness of the data. The proposed approach is evaluated by analyzing a real life data set retrieved from a group of portals publishing real-estate listings.

Real-estate listings are published on multiple portals to gain audience, but this has led to a common problem called information overload. Moreover most of the times data attributes like price, floor etc. are different or missing, which makes the task of clustering more complicated. In addition, it is not unusual to find the same property being described many times on the same portal, in some cases with varying descriptions. This provides an interesting example of multiple heterogeneous data sources containing partly overlapping data both at an intra-source and inter-source level.

The methods, described in this work aim to automatically cluster individual real-estate listings coming from different web data sources in order to provide a global view of the data, eliminate redundancies and integrate the data describing every property that may not be fully available in any single data source. Moreover, by identifying individual properties being described in a number of property listings and tracking the changes in the source systems on regular bases, one can track price changes as well. In addition, the time between offering the property for sale in any of the source systems and removing it from the source systems can be calculated. Thus, the applications of the proposed approach may go far beyond integrating the data of individual properties.

The remainder of this paper is organized as follows:

- Section 2 presents related work in the field, both in terms of integration of semi-structured data and clustering of incomplete data.
- In section 3 the overall architecture of an integrated system is described. This provides the basis for the presentation of the data set acquired from monitored real-estate portals.
- New clustering methods using a weighted distance function and combining the similarity of structured and unstructured content have been proposed next.
- Finally, the comparison of results has been presented. This is followed by conclusions in section 5.

2 Background and Related Work

2.1 Web Content Mining and Information Integration

Rapid growth of web data in the recent years offers unprecedented opportunities for web content mining. This relatively new research area attempts to address a number of problems. An overview of web mining research areas with particular attention to data extraction issues can be found in [5].

In the analyzed problem, three popular real-estate portals are considered. Thus, the emphasis has been placed on *data integration* [5]. The data integration techniques address a growing demand for uniform methods of accessing the data irrespective of the actual web data sources [2, 3]. In general two approaches to data integration can be considered [2]. In the first approach, the data from numerous data sources is collected in a central location and offered through global views. In the second approach, the systems capable of transforming user requests to queries made on-the-fly to individual data sources are attempted [2]. An advantage of the first approach is that the data collected over some period of time can be further investigated. Thus, not only can the current integrated data set be queried, but also the past content can be analyzed and examined.

The need for integrating and visualizing the real estate property information has been addressed by T. Hong [1]. Grammar induction techniques have been proposed to automatically parse different web documents and collate all of the real estate listings together. Afterwards, geographically-based visualization has been implemented. The latter work and most other works concentrate on wrapper generation techniques and query planning and execution [1]. Limited attention has been paid to the incompleteness of the data and the need for identifying individual real-estate properties.

Thus, two distinctive factors suggest further research in the field. First of all, the same real-property might be described numerous times on the same or different portals. Secondly, a critical problem is that some objects from various sources might have missing values. In our crawl in 48% of cases the same property advertised on different web portals either has missing or different values.

2.2 Dealing with Incomplete Data

Data imputation [6], where missing values are estimated and marginalization, where missing values are ignored are two common methods used to handle missing data. As imputation should not be considered as reliable as the actual data, marginalization has been considered to be the better option. Most previous works with marginalization were focused on supervised methods such as Neural Networks [7].

As far as unsupervised clustering is concerned, Fuzzy c-means (FCM) [8] clustering provides a base to cluster items with missing datum. Geometry of a missing value in vector is a line or in high dimensionality is a plane, e.g. (2, ?) is a line where all x is 2. All the techniques discussed above produce good results when low number of features is missing with respect to data dimensionality. In the analysed case in some records most features have been missing.

3 Proposed Approach

3.1 System Architecture

The application created as a part of this study consists of the two core modules. The first module is a crawler engine containing manually developed wrappers

used to extract the data from monitored real-estate portals. It is accompanied by a web application placing the records captured from individual data sources in a database, clustering them and displaying the global view of real-estate clusters.

3.2 Feature Selection and a Reference Data Set

Data from 3 different sources being different real estate portals with over 20 000 entries has been harvested over one year of time span. The following features have been collected: **Type**, **Description**, **Price**, **Area**, **Number of rooms**, **Year of construction**, **Floor**, **Total number of floors in a building**, **Street**, **District**, **City**, **Coordinates** (longitude and latitude) and **URI**. Geographical coordinates have been calculated by openly available geocoding webservices basing on the city and street name.

The features are divided into **Description** being an unstructured text, the set of mandatory attributes $F_s = \{\text{Type, City, Price, Area}\}$, and the remaining attributes F_m , which are allowed to be empty. Database records x have been created only for $\tilde{x} : \forall i \in F_s : \tilde{x}_i \neq ?$, where $?$ stands for a missing value. The idea is to cluster items over F_s considering partial distance over F_m .

To compare our approach with other approaches a reference dataset X consisting of 1000 randomly selected records has been created. All the data has been manually inspected by the author to correctly cluster the records describing the same property. The manual process has been based on the investigation of not only the attributes of every record, but also unstructured content being the text description and photos accompanying most records. The latter content in many cases precisely decides whether different records describe the same property or not. In this way a real-life reference data set has been created. This provides basis for the comparison of clustering methods discussed below.

3.3 Distance Functions

Basic Distance Function. Two different distance functions: one without unstructured text and the second one with unstructured text have been used. In the first distance function the distance has been calculated using a relative weight vector w . The normalization of data enables relative importance of some features in datum. For example, while clustering, **Type** feature should be the same that is, all apartments should be clustered together. So, this importance is done via weights $w_i \in [0, 1]$. At the same time, all the weights $w_i, i = 1 \dots \text{card}(F_s) + \text{card}(F_m)$ can be set by the system administrator to fully control the clustering process and define the notion of similarity required for the two records x, y to be placed in the same data cluster.

Let us consider a weight vector $w = [w_1, w_2, \dots, w_{\text{card}(F_s) + \text{card}(F_m)}]$. The distance between two real-estate listings can be computed as $d_B(x, y)$:

$$d_B(x, y) = \sum_{i=1}^{\text{card}(F_s)} w_i \phi_i(x_i, y_i) + \sum_{j=\text{card}(F_s)+1}^{\text{card}(F_s) + \text{card}(F_m)} w_j \phi_j(x_j, y_j) \quad (1)$$

$\phi_j()$ is the distance function calculated for a pair of potentially missing attributes which is defined as

$$\phi_j(x_j, y_j) = \begin{cases} 0 & \text{if } (x_j = ? \vee y_j = ?) \\ \tilde{\phi}_j(x_j, y_j) & \text{otherwise} \end{cases} \tag{2}$$

The distance function $\tilde{\phi}_i()$ can be different for every feature i . Still every function follows the same template i.e.

$$\tilde{\phi}_i(a, b) = \begin{cases} \infty & \text{if } |a - b| > \eta_i \\ |a - b| & \text{otherwise} \end{cases} \tag{3}$$

In other words, η_i defines allowable difference between two values of feature i i.e. the difference justified by the ambiguity of an attribute. For instance, floor number can differ by one ($\eta_i = 1$), because two-floor apartment can be described using the upper or lower floor number. The $d_B(x, y)$ function will be referred to as the basic distance function (BDF), as it does not take into account the similarity of unstructured text describing a property.

Combined Distance Function. In the second distance function, the same distance function has been combined with *Jaccard distance* considering unstructured text of a **Description** feature. The Jaccard distance is defined as:

$$d_J(x, y) = 1 - J(x, y) = \frac{|W(x) \cup W(y)| - |W(x) \cap W(y)|}{|W(x) \cup W(y)|} \tag{4}$$

where $W(x)$ and $W(y)$ stand for the sets of words used in the **Description** attribute of x and y , respectively. $J(A, B)$ is a value between 0 and 1. Lower value implies closer data points or in other words relatively similar data points.

Let $d_2(x, y) = \delta d_B(x, y) + (1 - \delta)d_J(x, y)$. In the analyzed case, the description of a real-estate listing is an unstructured text. Computing distance using Jaccard similarity in such cases would be very expensive, because of N-dimensional data. Locality-sensitive hashing (LSH) [9] performs probabilistic dimension reduction of high dimensional data. Hence, BDF has been combined with LSH to utilize unstructured information and minimize computation overhead. Since $Prob[h(x) = h(y)] = J(x, y)$,

$$d_C(x, y) = \delta d_B(x, y) + (1 - \delta)(1 - Prob[h(x) = h(y)]) \tag{5}$$

The $d_C()$ function will be referred to as Combined Distance Function (CDF). In this case, the distance between two records representing two different property listings is calculated using both mandatory attributes of F_s , optional and potentially empty attributes of F_m and unstructured text description.

3.4 Weighted Clustering Method

An important aspect of the analysis is time complexity of clustering algorithms. Considering the number of records acquired from different data sources and the

need to recalculate clusters on regular basis in order to reflect dynamic changes in property listings, the computational complexity should be possibly low. Thus, a combination of Canopy clustering [10] being is a less computationally expensive version of k-means with weighted distance function has been used. The algorithm runs in $O(n \log(n))$ and there is an evidence that it performs better than k-means clustering [10]. In weighted clustering, Canopy clustering using weight vectors w and a BDF function has been applied. The method is referred to as Weighted Clustering Method (WCM). Distance functions $\phi_i()$ used to evaluate the similarity of individual features have been involved, too.

3.5 Weighted and LSH Clustering Method

The same Canopy clustering algorithm has been used with a combined distance function $d_C()$. While the same ϕ_i functions have been used, the difference is that LSH-based distance calculation for unstructured part has been performed as well. In this case $\delta = 0.85$ has been used for the experiments. It means 15% importance has been given to unstructured text. In general, the similarity of the structured data has to play the key role in the clustering process in order to minimise the risk of placing records with contradictory attributes in the same cluster. The method is referred to as Weighted and LSH Clustering Method (WLSHM). It requires a separate study to investigate the impact of different ϕ_i settings on the clustering process. Still, improvement resulting from the use of WLSHM has been shown.

4 Results

In order to evaluate the proposed clustering algorithms, rand index $R()$ has been applied:

$$R(M) = \frac{TP(M) + TN(M)}{TP(M) + TN(M) + FP(M) + FN(M)} \quad (6)$$

$TP()$, $TN()$, $FP()$, and $FN()$ stand for true positive, true negative, false positive and false negative and are used to evaluate the quality of clusters by comparing them against the reference set of clusters.

The techniques proposed in this work have been compared with k-means, Min-Hash (LSH) [11], FCM and a combination of FCM and LSH clustering algorithm. The comparison has been performed for two benchmark problems: Exact Matching Problem (EMP) and Near-Neighbour Matching Problem (NNMP). In EMP exact real-estate listings having the same sets of non-empty features are examined. In NNMP near listings based on user thresholds η_i and potentially missing features are considered. The clusters created by all algorithms for the two problems have been compared against the clusters manually created for a reference data set X . Two cluster sets $S_E = E_1 \cup E_2 \cup \dots \cup E_{192}$ and $S_N = S_1 \cup S_2 \cup \dots \cup S_{168}$ have been identified manually for the data set X and denote a correct solution of EMP and NNMP, respectively. The results of the simulations have been summarized in Table 1.

Table 1. Results summary

Method	EMP		NNMP	
	Time(sec.)	$R(M)$	Time(sec.)	$R(M)$
k-means	4.2	0.80	5.6	0.66
FCM	3.1	0.87	4.8	0.66
LSH	1.2	0.81	1.4	0.60
FCM+LSH	2.4	0.89	3.2	0.73
WCM	1.6	0.97	1.9	0.84
WLSHM	2.0	0.98	2.2	0.90

Search for an apartment

Cluster ID 3816 (7)

Average price	Price per sqm	Median listing price	Median per sqm price
392 142 PLN	6 127 PLN / sqm	399 000 PLN	6 234 PLN / sqm

Properties in this group

[New lower price, beautiful 3-room apartment, 64m2, price 382 000](#)

Upper street, Western district
382 000 PLN 64m2 3 rooms 4th floor Total number of floors 4
 Published 5 days, 11 hours ago

[64m2 in the city centre, perfect fittings, low housing, quiet surroundings - see it!](#)

Upper street, Western district
395 000 PLN 64m2 3 rooms 4th floor Total number of floors 4
 Parking place
 Published 2 months, 1 week ago

[Great 3-room apartment, 64m2, priced 400 000](#)

Upper street, Western district
400 000 PLN 64m2 3 rooms 4th floor Total number of floors 4
 Published 3 months, 2 weeks ago

Fig. 1. Sample clustering result shown in a web application

LSH turns out to be the fastest algorithm. Still, weighted approach (WCM) results in significant improvement over standard k-means and FCM clustering algorithms in terms of the rand index value. In other words, WCM provides clusters that are much closer to the objective of identifying correct solution both for EMP and NNMP. Moreover, by investigating the similarity of unstructured text through LSH algorithm, even further improvement can be achieved. This results in $R(M) = 0.98$ and $R(M) = 0.90$ when WLSHM is used for EMP and NNMP, respectively. What is important, the additional overhead due to the processing of unstructured text is negligible.

Fig. 1 shows sample cluster displayed in a web application being a part of the solution (English names have been used instead of the original text attributes). Multiple listings referring to the same property have been correctly identified. Moreover, price changes in consecutive listings can be observed.

5 Conclusions

A novel technique of clustering real-estate listings from WWW has been proposed. This technique can be used to cluster data of different categories, composed of semi-structured and unstructured components. The challenge behind good clustering technique for real-estate listings has been discussed. The tests have shown that both weighted approach and its LSH variant is capable of detecting duplicates even if missing data or minor differences in feature values are present. Therefore, the proposed technique can be applied not just to collect a sum of records, but also to identify the subject matter and cluster the data describing it.

References

1. Hong, T.: Visualizing real estate property information on the Web. In: Proceedings of IEEE International Conference on Information Visualization, pp. 182–187 (1999)
2. Lim, S.: An automated integration approach for semi-structured and structured data. In: The Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications, CODAS 2001, pp. 12–21 (2001)
3. Shaker, M., et al.: A Framework for Extracting Information from Semi-Structured Web Data Sources. In: Third International Conference on Convergence and Hybrid Information Technology, ICCIT '08, pp. 27–31 (2008)
4. Phong Bao Vung, L., Gao, X., Zhang, M.: Data Extraction from Semi-structured Web Pages by Clustering. In: IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006, pp. 374–377 (2006)
5. Pol, K., et al.: A Survey on Web Content Mining and Extraction of Structured and Semistructured Data. In: First International Conference on Emerging Trends in Engineering and Technology, ICETET '08, pp. 543–546 (2010)
6. Zawistowski, P., Grzenda, M.: Handling Incomplete Data Using Evolution of Imputation Methods. In: Kolehmainen, M., et al. (eds.) ICANNGA 2009, LNCS, vol. 5495, pp. 22–31. Springer, Heidelberg (2009)
7. Tresp, V., Neuneier, R., Ahmad, S.: Efficient methods for dealing with missing data in supervised learning. *Advances in Neural Information Processing Systems*, 689–696 (1995)
8. Hathaway, R., Bezdek, J.: Fuzzy c-means clustering of incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 31(5), 735–744 (2001)
9. Andoni, A., Indyk, P.: Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions. *Communications of the ACM* 51(1) (2008)
10. McCallum, A., Nigam, K., Ungar, L.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 169–178. ACM, New York (2000)
11. Cohen, E., et al.: Finding interesting associations without support pruning. *IEEE Transactions on Knowledge and Data Engineering* 13(1), 64–78 (2001)

Incremental Update of Cyclic Association Rules

Eya Ben Ahmed

Higher Institute of Management of Tunis, Tunisia
eya.benahmed@gmail.com

Abstract. A promising challenge of data mining, especially for association rules technique, is the incremental mining of association whatever is the trend of the association rules. Recently, some researches were devoted to incremental update of temporal association rules problem. In this paper, we focus on *cyclic association rules*, a class of temporal association rules. Thus, we introduce a new approach called IUPCAR dedicated to maintaining incrementally the cyclic association rules already extracted. Based on the carried out experimental study, we point out the efficiency of our proposal.

Keywords: Cyclic association rules, Incremental maintenance of cyclic association rules, incremental update of cyclic association rules.

1 Introduction

Through time, the volume of information increases, the databases must be updated with the new amounts of data. Considering that an association rule generates explicitly reliable knowledge according to an explored database at an accurate time. So that, each update of the database radically overwhelms the already stored patterns. A projection of the database's changes must be drawn on extracted association rules. Since then, several proposals to solve this problem appeared [3,4]. Parallel to those efforts, cyclic mining of association rules was also investigated on several studies. Such investigations can be found in [1,2]. The problem of cyclic association rules mining consists in generation of association rules from articles characterized by regular cyclic variation over time. In [1], Ozdon et al. presented the first strategies of cyclic association rules extraction. Then, as a response to the anomalies characterizing the already proposed approaches, a more efficient algorithm was introduced by Ben Ahmed et al. [2]. It can be seen that the research community has proposed separate solutions for the incremental mining and the cyclic association rules mining problems. In this paper, we present a new algorithm called IUPCAR (Incremental Update of Cyclic Association Rules) for incremental mining of cyclic association rules. This algorithm provides the benefits of fast incremental mining and efficient cyclic association rules extraction. The rest of the paper is organized as follows: section 2 studies the fundamental bases on which our proposal is built. Section 3 presents a formal description of the problem. Section 4 details our proposal. Carried out experiments stressing on the efficiency of our proposal are sketched in section 5. Finally, section 6 concludes the paper and points out avenues for future work.

2 Foundations of the Proposed Algorithm

This section presents the previous theoretical foundations which form the bases on which our proposed algorithm is built. These bases include the theoretical foundations of the two problems of cyclic association rules mining and incremental mining of association rules. Therefore, we briefly discuss the cyclic association rules. Finally, we describe the incremental mining problem.

2.1 Cyclic Association Rules

We present the basic concepts related to cyclic association rules that will be of use in the remainder.

Time unit. Considering the temporal aspect, the first considered measure is the time unit. Firstly, it was introduced by Odzen *et al* [1].

Definition 1. *Given a transactional database DB, each time unit u_i corresponds to the time scale on the database [1].*

Example 1. *Let the following example be highlighted in table 1.*

Table 1. Initial database DB

Transaction ID	Items	Transaction ID	Items	Transaction ID	Items
1	B	2	A, B	3	A, B, C, D
4	A, B, C	5	C	6	A

The transactions illustrated in table 1 are extracted hourly. So that, the corresponding time unit is the hour.

Cycle. The concept of cycle was primarily introduced by Odzen *et al* [1].

Definition 2. *A cycle c is a tuple (l, o) , such that l is the length cycle, being multiples of the unit of time; o is an offset designating the first time unit where the cycle appeared.*

Thus, we conclude that $0 \leq o < l$.

Example 2. *If we consider a length of cycle $l = 2$ and the corresponding offset is 1. So that, the cycle $c = (l, o) = (2, 1)$.*

Approaches addressing the issue of cyclic association rules are the following: (i) The **Sequential Approach**: is a two-phase based algorithm. The key idea is: (i) to generate the large itemsets and to extract straightforwardly the corresponding association rules. (ii) to detect the cycles of rules. So those are cyclic will be kept and the remainder is pruned; (ii) The **Interleaved Approach**: is a three-phase based algorithm. Thanks to cycle pruning, we generate the potential cycles. For every unit of time, we apply the cycle skipping to extract the itemsets and we

count their support thanks to the cycle elimination; (iii) **The PCAR Approach:** Radically, it is based on the segmentation of the database in a number of partition fixed by the user. The browse of the database is done sequentially partition by partition. This latter is scanned to generate the frequent cyclic itemsets. Achieving the last one, we obtain the set of frequent cyclic itemsets. Hence, we extract from them the cyclic association rules.

2.2 Incremental Mining of Association Rules

In the incremental context, several streams of approaches were reported to update incrementally the discovered association rules. We start by introducing the most well-known ones. Initially, the key idea of incremental update was proposed by Cheung et al by introducing FUP for incrementally updating frequent itemsets [3]. Inspired from the APRIORI algorithm, the key idea of FUP is that by adding the new transactions db to the initial database *DB*, some previously frequent itemsets will remain frequent and some previously infrequent itemsets will become frequent (these itemsets are called winners). At the same time, some previously frequent itemsets will become infrequent (these itemsets are called losers). The BORDERS Algorithm developed by Thomas *et al.* [4] and Feldman *et al.* [5], is another approach using the concept of "negative border" introduced by Toivonen [6] aiming to indicate if it is necessary or not to check any candidate against the initial database. the algorithm maintains information about the support of frequent itemsets in the original database along with the support of their negative border. If any itemset becomes a winner in the updated database, it follows that some itemset formerly in the negative border will also become a winner.

3 Incremental Mining of Cyclic Association Rules

Along this section, we present a description of the tackled problem. First, we formally define the problem of cyclic association rules. Then, we present the basic notions.

3.1 Formal Problem Description

Regarding cyclic association rules, we stress on rules that are repeated in a cyclical way. Indeed, given a length of cycle, we extract itemsets that appear sequentially in the database. Let consider *X* and *Y* two itemsets appearing in *DB* at transaction number *i* and sequentially at the transaction number $i + \text{LENGTH OF CYCLE}$ until the end of the database (Table 2). According to given support threshold, we prune the non frequent cyclic itemsets. Thus, we generate the cyclic association rules based on minimum confidence threshold.

To summarize, the cyclic association rules mining problem can be reduced to extraction of frequent cyclic itemsets, because once we have frequent cyclic itemsets set, cyclic association rules generation will be straightforward.

Table 2. Cyclic itemsets in DB

Transaction ID	Items
i	X , Y
...	...
i+length of cycle	X , Y
...	...
i+(length of cycle*k)	X , Y

After several updates of *DB*, an increment *db* of $|db|$ transactions is added to *DB*. The problem of incremental maintenance of cyclic association rules is to compute the new set of the frequent cyclic itemsets in $DB' = DB \cup db$ according to a support threshold *MinSup*.

In order to extract cyclic association rules from databases, the only plausible solution is to rerun one of the classical algorithms dedicated to the generation of cyclic association rules *i.e.*, SEQUENTIAL, INTERLEAVED or PCAR. As a result, two drawbacks are quoted: (i) If the original database is large, much computation time is wasted in maintaining association rules whenever new transactions are generated; (ii) Information previously mined from the original database, provides no help in the maintenance process.

3.2 Basic Notions

We start this subsection by presenting the key settings that will be of use in the remainder.

Frequent Cyclic itemset. This concept refers to cyclic itemsets having supports exceeding the considered threshold. The formal definition is as follows.

Definition 3. Let *XY* be an itemset, the $sup(XY)$ is the support of the itemset in the database, reminding that only cyclic occurrences are considered on the support computing, and the minimum support threshold reminding *MinSup*. The itemset *XY* is considered as **Frequent Cyclic** denoted **FC** if the cyclic occurrences of the itemset *XY* are greater or equal to the given support threshold otherwise if $sup(XY) \geq MinSup$.

Example 3. We consider the context shown by table 1, the *Minsup* equal to 2 and the length of cycle is 2. The binary sequence representing the itemset *AB* is 011100 so $sup(AB) = MinSup = 2$ then *AB* is called **Frequent Cyclic itemset FC**.

Frequent Pseudo-Cyclic itemset. The frequent pseudo-cyclic concept is presented as follows.

Definition 4. Let *XY* be an itemset, the $sup(XY)$ is the support of the itemset in the database, the *MinSup* the minimum support threshold. The itemset *XY* is considered as **frequent pseudo-cyclic** denoted **FPC** if its support is less than *MinSup*. Simultaneously, its support is greater than a given threshold called **MinFPC** : $MinFPC \leq sup(XY) < MinSup$.

Example 4. Given the previous context, we consider $MinSup$ equal to 2, the $MinFPC$ is 0.2 and the length of cycle is 2. The binary sequence representing the itemset AD is 000100 so $sup(AD) = 1 < MinSup=2 \geq MinFPC=0.2$ then AD is called **Frequent Pseudo-Cyclic itemset FPC**.

Minimum FPC threshold. According to this measure, we classify the remainder of the itemsets after $MinSup$ pruning on hopeful cyclic itemsets that are not frequent in the initial database but are more likely to move to this status in the increment database.

Definition 5. The **Minimum FPC threshold**, denoted by **MinFPC**, refers to a threshold dedicated to prune the none hopeful itemsets. It is computed according to this formula: $MinFPC = \frac{MinSup}{|DB|+|db|} + MinSup$

Example 5. Continuing with the same database DB considered as initial one, let the database db containing 4 transactions be the increment one. In addition we fix the $MinSup$ to 2. Then the $MinFPC$ is computed as follows: $MinFPC = \frac{2}{|6|+|4|} + 2 = 0.2$

Non Frequent Cyclic Itemset. This concept refers to cyclic itemsets that are not both frequent cyclic itemsets and frequent pseudo-cyclic itemsets.

Definition 6. Let XY be an itemset, the $sup(XY)$ is the support of the itemset in the database and the $MinSup$ the minimum support threshold. The itemset XY is considered as **non frequent cyclic itemset** denoted **NFC** if the support of XY is less than the given $MinFPC$ threshold otherwise if $sup(XY) < MinFPC$.

Example 6. Given the previous context, we consider $MinSup$ equal to 4, the **MinFPC** is 2 and the length of cycle is 2. The binary sequence representing the itemset AD is 000010 so $sup(AD)=1 < MinFPC=2 < MinSup=4$ then AD is called **non frequent cyclic itemset** denoted **NFC**.

In this respect, the main thrust of this paper is to propose a new strategy dedicated to the incremental update of cyclic association rules aiming to reduce efficiently the runtime required for the generation of cyclic association rules in the case of addition of transactions at the maintenance process of databases. Indeed, this proposal is outlined in the following section.

4 IUPCAR Algorithm

In order to maintain incrementally the cyclic association rules, we introduce a novel approach called INCREMENTAL UPDATE OF CYCLIC ASSOCIATION RULES denoted IUPCAR. Indeed, the IUPCAR algorithm operates in three phases: (i) In the first phase, a scan of the initial database is done to class the founded itemsets on three classes namely the frequent cyclic itemsets, the frequent pseudo-cyclic itemsets and non frequent cyclic itemsets; (ii) In the second phase, according to the second database, we categorize the itemsets into the three mentioned

classes. Then, depending of the ancient class of the itemset with its ancient support and the new class with its new support in the increment database, an affectation of the suitable class is made according to a weighting model; (iii) In the final phase, given the founded frequent cyclic itemsets after the update operation, the corresponding cyclic association rules are generated.

First and foremost, the IUPCAR algorithm takes on input the initial database, the minimum support threshold $MinSup$, the minimum confidence threshold $MinConf$ and the length of cycle. According to those key settings, a generation of frequent cyclic itemsets, frequent pseudo-cyclic itemsets and non frequent cyclic itemsets from the initial transactions is done. Stressing on the dynamicistic feature of the databases, we add the novel transactions building the db database. To accomplish this update operation, a scan of the new database is done and a generation of the itemsets and their classification are straightforwardly realized. After that, an update of the status and the weights of itemsets are done without rescanning the initial database. Finally, we generate the cyclic association rules based on the retained frequent cyclic itemsets.

Intuitively in the updating problem, we assume the following cases: (i) Frequent cyclic itemset FC is already saved as frequent cyclic FC (case **A**), frequent pseudo-cyclic FPC (case **B**) or non frequent cyclic itemset CNF (case **C**);(ii) Frequent pseudo-cyclic itemset FPC is already saved as a frequent cyclic FC (case **D**), frequent pseudo-cyclic FPC (case **E**) or non frequent cyclic itemset NFC (case **F**);(iii) Non frequent cyclic itemset NFC is already saved as frequent cyclic FC (case **G**), frequent pseudo-cyclic FPC (case **H**) or non frequent cyclic itemset NFC (case **J**).

To handle those various cases, we introduce the following weighting model. In the update operation, a dramatic change in the status of the itemsets between the first and the coming database is intuitively plausible. That's why, we refer to the weighting model as a technique dedicated to decide which status is the suitable to the itemset after the new added transactions. Indeed, we sketch the mechanism of weighting model as follows.

For an itemset X , we :

- *compute* the relative support of X in the initial database DB , denoted by $Sup(X_{DB})$, according to to the given formula: $Sup(X_{DB}) = \frac{Sup(X)}{|DB|}$
- *compute* the relative support of X in the increment database db , denoted by $Sup(X_{db})$, according to the given formula: $Sup(X_{db}) = \frac{Sup(X)}{|db|}$
- *compare* the relative support of X in the initial database $Sup(X_{DB})$ *vs.* that of the increment database $Sup(X_{db})$. And we *choose the greatest one*.
- Two alternatives are plausible :
 1. If the itemset has the same state in the initial and the incremental database, we will **enhance** its weight;
 2. If the state of the itemset has changed from the initial to the incremental database, we will **check** which one of its states has the greatest weight and we will **decrease** its value and **affect** this state as its new one.

In this respect, let the new weight of X be denoted by $W(X_{db})$.

The possible cases that can be summarized on three possible scenarios:

1. No change in the status simply happens. So, the itemset remains frequent cyclic FC (case **A**) or frequent pseudo-cyclic itemset FPC (case **E**) or non cyclic frequent CNF (case **J**). The new weight is computed as follows: $W(X_{db}) = \frac{Sup(X_{DB}) + Sup(X_{db})}{|DB| + |db|}$
 2. A change in the status between the initial transactions and the new ones occurs. So one of the cases (case **B**), (case **C**), (case **D**), (case **F**), (case **G**) or (case **H**) happens. Then, two situations are obviously outlined:
 - (a) If *the previous support of the itemset is greater than the new one in the increment database*, the affected status is remained the same and its novel weight is computed as follows: $W(X_{db}) = \frac{Sup(X_{DB})}{|DB|} - \frac{Sup(X_{db})}{|db|}$
 - (b) If *the new support of the itemset is greater than the previous one*, the affected status is the new one and its novel weight is computed as follows: $W(X_{db}) = \frac{Sup(X_{db})}{|db|} - \frac{Sup(X_{DB})}{|DB|}$
- Considering the update operation of the itemsets' status and weights, we extract cyclic association rules based on frequent cyclic itemsets.

5 Experimental Study

To assess the IUPCAR efficiency, we conducted several experiments on a PC equipped with a 3GHz Pentium IV and 2GB of main memory. During the carried out experimentation, we used benchmarks datasets taken from the UC Irvine Machine Learning Database Repository. Through these experiments, we have a twofold aim: first, we have to stress on the performance of our proposal by the variation of $MinSup$; second we put the focus on the efficiency of our approach *vs.* that proposed by the related approaches of the literature.

5.1 Performance Aspect

In the carried out experimentations, we divided database in two partitions: DB is the initial database and db is the incremental one. Firstly, the DB is constituted of 70% of the size of the benchmark dataset and db is constituted of the remainder namely the 30%. Secondly, we increase the size of the initial database to achieve 80% from the size of the benchmark dataset so the db presents only 20%. To finish with an initial database representing 90% and an incremental one providing only 10%.

Considering the given parameters : the $MinConf = 50\%$, the length of cycle $= 30$, we present the variation of $MinSup$ and the corresponding runtime of IUPCAR in figure 11. Indeed, by varying the support, it is obvious that the more support is increasing the more runtime of IUPCAR decreases. For the T10I4D100K dataset, having $DB=70\%$ and $db=30\%$, the runtime of IUPCAR increases from 926,295 seconds for 1% as a $MinSup$ to 482,726 seconds for 50% as

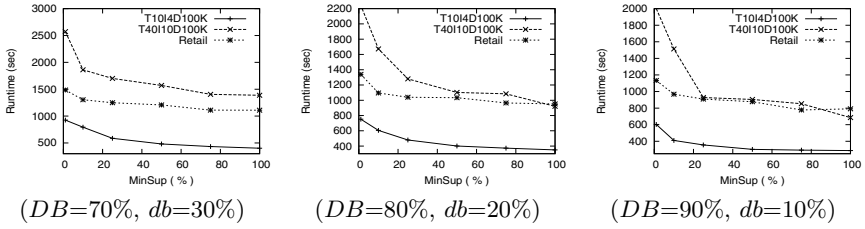


Fig. 1. Experimental results of IUPCAR for an incremental database =10%, 20% or 30%

a *MinSup*, to stabilize at around 400, for *MinSup* exceeding 50%. As expected, this fact is similarly conceivable for T40I10D100K and RETAIL datasets.

As shown figure 1, we assume worthily that on whatever the size of the initial and increment database, the more support increases the more runtime required for IUPCAR goes down.

5.2 Efficiency Aspect

In order to evaluate the efficiency of our algorithm, we conducted comprehensive experiments to compare IUPCAR with the most efficient classical algorithm dedicated to cyclic association rules extraction namely PCAR algorithm. The following values of parameters are set during the several experiences: the minimum of confidence equal to 50%, the length of the cycle equal to 30 and the runtime of the algorithms regarding the T10I4D100K, T40I10D100K and RETAIL datasets.

The results of varying the minimum support on running the PCAR algorithm and the IUPCAR one are shown by figure 2.

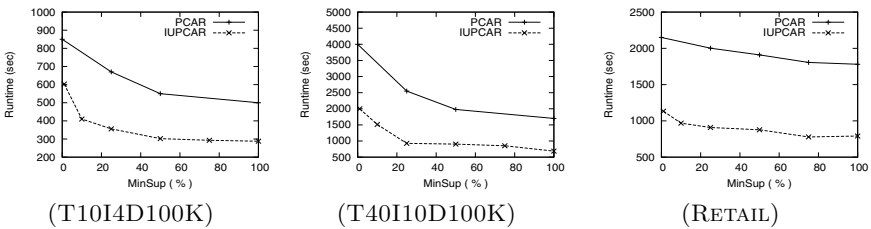


Fig. 2. Comparing the runtime of IUPCAR and PCAR with incremental database 10%

It indicates that the update operation of *DB* by the increment *db*=10% with a minimum support =100%, requires 511,388 seconds by running PCAR and only its half 288,62 seconds by running IUPCAR for T10I4D100K dataset. For T40I10D100K, the original database *DB* =90% with the increment database *db* =10% required with a minimum support=100% for T40I10D100K by running PCAR 1776,12 seconds and interestingly its half 686,884 seconds for IUPCAR

running. Similarly for RETAIL, the updating operation of the initial database by adding 10% of its size with a minimum support=100% requires 1894,416 seconds by running PCAR and efficiently 791,9 seconds for IUPCAR running.

Obviously, IUPCAR amply outperforms the PCAR algorithm in the context of maintenance of cyclic association rules and proves its efficiency in various tests.

6 Conclusion and Perspectives

In this paper, we introduced the problem of incremental maintenance of cyclic association rules. Thus, the flying over the pioneering approaches handling the incremental update of association rules issue conducted us to introduce a new proposal called IUPCAR algorithm dedicated particularly to update the cyclic association rules. To evaluate its efficiency, several experimentations of the proposed method are carried out. So that, encouraging results are obtained. Future work will focus mainly on : (i) the quality of the generated cyclic association rules. In fact, we plan to study deeply the significance of the extracted cyclic association rules for human experts, (ii) tackling the change support threshold in the incremental update operation of cyclic association rules.

References

1. Ozden, B., Ramaswamy, S., Silberschatz, A.: Cyclic Association Rules. In: 14th International Conference on Data Engineering (ICDE'98), p. 412 (1998)
2. Ben Ahmed, E., Gouider, M.S.: PCAR: nouvelle approche de génération de règles d'association cycliques. In: EGC, DBLP. DBLP:conf/f-egc/2010, pp. 673–674 (2010), <http://dblp.uni-trier.de>
3. Cheung, D.W., Wong, C.Y., Han, J., Ng, V.T.: Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique. In: International Conference on Data Engineering, CA, USA, p. 106 (1996)
4. Shiby, T., Sreath, B., Alsabti, K., Sanjay, R.: An efficient algorithm for the incremental updation of association rules in large databases. In: Proceedings of the 3rd International conference on Knowledge Discovery Data Mining (KDD '97), New Port Beach, California (1997)
5. Feldman, R., Aumann, Y., Amir, A., Mannila, H.: Efficient Algorithms for discovering Frequent Sets in Incremental Databases. In: Proceedings of the 1997 SIGMOD Workshop on DMKD, Tucson, Arizon (1997)
6. Toivonen, H.: Sampling large databases for association rules. In: 22nd International Conference on Very Large Databases (VLDB'96), Mumbai, India, pp. 134–145 (1996)

Author Index

- Ahmad, Khurshid 218
Ávila, Bráulio C. 308
- Bagnall, Anthony 234
Bajer, Lukáš 251
Bańkowski, Sławomir 292
Bao, Yongguang 94
Belo, Orlando 194
Ben Ahmed, Eya 387
Biehl, Michael 21
Binaghi, Elisabetta 110
- Camacho, David 259
Carrijo, Gilberto Arantes 29
Chang, Chih-Cheng 145
Chen, Kuan-Ta 316
Chien, Li-Jen 145
Ciecholewski, Marcin 63
Cohagan, Clinton 118
Costa, José Alfredo F. 334
- D.V.L.N, Somayajulu 202
Davies, Anthony 218
de Almeida, Ana 210
de Borja Rodríguez, Francisco 259
de Haro-García, Aida 1
del Castillo, Juan Antonio Romero 1
Dicken, Luke 186
Dillon, Tharam S. 324
- Elfers, Carsten 13
Enembreck, Fabrício 308
Esseghir, M.A. 226
- Fadlil, Junaidillah 316
Ferreira, Júlio César 29
Figueiredo, Marisa B. 210
Forestier, Germain 45
Fujimoto, Yu 153
- Gançarski, Pierre 45
García-Pedrajas, Nicolás 1
Goncalves, Gilles 226
Gorawski, Marcin 292
Gorawski, Michał 292
- Granados, Ana 259
Grzenda, Maciej 379
Grzymala-Busse, Jerzy W. 118
Guglielmin, Mauro 110
Guidali, Andrea 110
Gutierrez, Eladio 162
- Hadzic, Fedja 324
Hammer, Barbara 21
Herzog, Otthein 13
Hippe, Zdzislaw S. 118
Holeña, Martin 251
Horstmann, Mirko 13
Hughes, Arthur 218
- Ishii, Naohiro 94
- Jumutc, Vilen 70
- Kadampur, Mohammad Ali 202
Kao, Zhi-Peng 145
Kelleher, Dermot 218
Kimura, Hiroaki 94
Koga, Hisashi 86
Krause, Andreas 78
Kryszkiewicz, Marzena 284
- Lasek, Piotr 284
Lebby, Gary L. 368
Lee, Yuh-Jye 145
Levine, John 186
Li, Xinyang 78
Lin, Hong-Yi 316
Lipinski, Piotr 344, 352
Lopes, Noel 275
Loureiro, Jorge 194
- Magdon-Ismail, Malik 300
Martínez, Rafael 259
Martins, António 210
Masseglia, Florent 45
Mehboob, Zareen 360
Miller, Shonique L. 368
Miura, Takao 53
Miyazaki, Kazuteru 178

- Moczurad, Włodzimierz 242
Morioka, Yuichi 94
Murata, Noboru 153
Nalepa, Gislaine M. 308
O'Connor, Maria F. 218
Osareh, Ali R. 368
Ouyang, Yicun 170
Pais, Mônica Sakuray 29
Pan, Qi H. 324
Pao, Hsing-Kuo 316
Parviainen, Elina 37
Pascale, Marco 110
Petitjean, François 45
Pittard, Jonathan J. 126
Plata, Oscar 162
Posadas-Yagüe, Juan-Luis 137
Poza-Luján, Jose-Luis 137
Purnell, Jonathan T. 300
Quisilant, Ricardo 162
Ribeiro, Bernardete 210, 275
Rodan, Ali 267
Scalabrin, Edson E. 308
Schleif, Frank-Michael 21
Schneider, Petra 21
Shioya, Isamu 53
Simó-Ten, Jose-Enrique 137
Slimani, Yahya 226
Sohr, Karsten 13
Szymański, Julian 102
Teixeira, Marconi Batista 29
Tharp, Alan L. 126
Thukral, Deepak 379
Tiño, Peter 267
Tomokazu, Tsuji 86
Villmann, Thomas 21
Watanabe, Toshinori 86
Yamanaka, Keiji 29
Yanagisawa, Takashi 53
Yin, Hujun 360
Yokoyama, Takanori 86
Younsi, Reda 234
Zapata, Emilio L. 162
Zayakin, Pawel 70
Zhang, Feng 170
Zheng, Chaoxin 218