Hujun Yin
José A.F. Costa
Guilherme Barreto (Eds.)

Intelligent Data Engineering and Automated Learning – IDEAL 2012

LNCS 7435

13th International Conference
Natal, Brazil, August 2012
Proceedings

Springer

Lecture Notes in Computer Science 7435

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

Hujun Yin José A.F. Costa
Guilherme Barreto (Eds.)

Intelligent Data Engineering and Automated Learning – IDEAL 2012

13th International Conference
Natal, Brazil, August 29-31, 2012
Proceedings

Springer

Volume Editors

Hujun Yin
University of Manchester
School of Electrical and Electronic Engineering
Sackville Street
Manchester, M13 9PL, UK
E-mail: hujun.yin@manchester.ac.uk

José A.F. Costa
Federal University of Rio Grande do Norte
Department of Electrical Engineering
Lagoa Nova
59072-970, Natal, RN, Brazil
E-mail: jafcosta@gmail.com

Guilherme Barreto
Federal University of Ceará
Department of Teleinformatics Engineering
Campus of Pici, CP 6005
60455-760, Fortaleza, CE, Brazil
E-mail: g_barreto@uol.com.br

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-32638-7 e-ISBN 978-3-642-32639-4
DOI 10.1007/978-3-642-32639-4
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012943982

CR Subject Classification (1998): I.2.6-7, H.2.8, I.5, I.2.10-11, J.3, F.2.2, H.3, F.1, H.4

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

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The IDEAL conference is established as a unique interdisciplinary forum in the areas of intelligent data analysis and associated learning paradigms. It attracts experts, researchers, leading academics, practitioners, and industry representatives from machine learning, information processing, data mining, knowledge management, bio-informatics, neuro-informatics, bio-inspired models, agents and distributed systems, and hybrid systems. It has enjoyed a vibrant and successful history in the last 14 years in over ten locations in seven different countries. It continues to evolve to embrace emerging topics and exciting trends. This year was not an exception: IDEAL set foot in one of the most exciting and the fast developing countries in South America: Brazil. The conference received over 200 submissions, which were strictly peer-reviewed by the Program Committee. Only the papers judged to be of sufficient quality and novelty were accepted and included in the proceedings.

This volume contains around 100 papers accepted and presented at the 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2012) held during August 29–31, 2012, in Natal, Brazil. The authors and attendees came from all over the world. The conference papers provide a good sample of current topics from methodologies, frameworks, and techniques to applications and case studies. The techniques include evolutionary algorithms, artificial neural networks, association rules, probabilistic modelling, agent modelling, particle swarm optimization, and kernel methods. The applications cover regression, classification, clustering and generic data mining, biological information processing, text processing, physical systems control, video analysis, and time series analysis. The conference also featured a Special Session on bio-inspired models for industrial and medical applications.

IDEAL 2012 enjoyed outstanding plenary talks by distinguished guest speakers: Jose C. Principe of the University of Florida, Hartmut Schmeck of the Karlsruhe Institute of Technology, Andre Carvalho of the University of São Paulo, Hojjat Adeli of Ohio State University, and Carlos E. Pedreira of The Federal University of Rio de Janeiro.

We would like to thank all the people who devoted so much time and effort to the successful running of the conference and in particular the members of the Program Committee and reviewers, as well as the authors of the papers who

contributed to conference. We are also grateful for the hard work by the local organizing team at UFRN, especially Jackson Gomes, in registration and local arrangements, as well as the help at Manchester, particularly Zareen Mehboob, in checking through all camera-ready files. Continued support and collaboration from Springer, in particular from the LNCS editors, Alfred Hofmann and Anna Kramer, are greatly appreciated.

June 2012

Hujun Yin
José A. Costa
Guilherme Barreto

Organization

General Chair

José A. Costa	Federal University of Rio Grande do Norte, Brazil
Hujun Yin	University of Manchester, UK

Program Chair

Guilherme Barreto	Federal University of Ceará, Brazil

Program Co-chairs

Peter Tino	University of Birmingham, UK
Marcelo Costa	Federal University of Minas Gerais, Brazil
Michal Wozniak	Wroclaw University of Technology, Poland

International Advisory Committee

Lei Xu (Chair)	Chinese University of Hong Kong, Hong Kong
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
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

Hujun Yin (Chair)	University of Manchester, UK
Laiwan Chan (Chair)	Chinese University of Hong Kong, Hong Kong
Nigel Allinson	University of Sheffield, UK
Yiu-ming Cheung	Hong Kong Baptist University, Hong Kong
Emilio Corchado	University of Burgos, Spain
Colin Fyfe	University of the West of Scotland, UK
Marc van Hulle	K.U. Leuven, Belgium
Samuel Kaski	Helsinki University of Technology, Finland

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

Asia Liaison

Yang Gao	Nanjing University, China

Europe Liaison

Alfredo Cuzzocrea	University of Calabria, Italy

Publicity Co-chairs

Emilio Corchado	University of Salamanca, Spain
Fernando Buarque	University of Pernambuco, Brazil

Program Committee

Ajith Abraham	Machine Intelligence Resarch Labs, USA
Jesús Alcalá-Fernández	University of Granada, Spain
Jamil Al Shaqsi	Sultan Qaboos University, Oman
Davide Anguita	University of Genoa, Italy
Bruno Apolloni	University of Milan, Italy
Francisco Assis	Federal University of Campina Grande, Brazil
Romis Attux	University of Campinas, Brazil
Javier Bajo Pérez	Pontifical University of Salamanca, Spain
Bruno Baruque	University of Burgos, Spain
Carmelo Bastos Filho	University of Pernambuco, Brazil
Amit Bhaya	COPPE/Federal University of Rio de Janeiro, Brazil
Silvio Bortoleto	Positivo University, Brazil
Vicente Botti	Polytechnic University of Valencia, Spain
Antonio Braga	Federal University of Minas Gerais, Brazil
Fernando Buarque	University of Pernambuco, Brazil
Andrés Bustillo	University of Burgos, Spain
Luiz Calôba	COPPE/Federal University of Rio de Janeiro, Brazil
Jose. Calvo-Rolle	Universidad de la Coruña, Spain
David Camacho	Universidad Autónoma de Madrid, Spain
Heloisa Camargo	Federal University of São Carlos, Brazil
Anne Canuto	Federal University of Rio Grande do Norte, Brazil

Jaime Cardoso	University of Porto, Portugal
Andre Carvalho	University of São Paulo, Brazil
Richard Chbeir	LE2I-CNRS, France
Songcan Chen	Nanjing University of Aeronautics and Astronautics, China
David Clifton	Oxford University, UK
André Coelho	University of Fortaleza, Brazil
Leandro Coelho	Pontifical Catholic University of Paraná, Brazil
Francesco Corona	Aalto University, Finland
Paulo Cortez	University of Minho, Portugal
Ernesto Cuadros-Vargas	San Pablo Catholic University, Peru
Alfredo Cuzzocrea	University of Calabria, Italy
Ernesto Damiani	University of Milan, Italy
Jose Manoel De Seixas	COPPE/Federal University of Rio de Janeiro, Brazil
José Dorronsoro	Autónoma de Madrid University, Spain
Adrião Duarte	Federal University of Rio Grande do Norte, Brazil
Pablo Estevez	University of Chile, Chile
Igor Farkas	Comenius University in Bratislava, Slovakia
Felipe M.G. França	COPPE/Federal University of Rio de Janeiro, Brazil
Richard Freeman	Michael Page International, UK
Kunihiko Fukushima	Kansai University, Japan
Colin Fyfe	University of the West of Scotland, UK
Marcus Gallagher	University of Queensland, Australia
Fernando Gomide	State University of Campinas, Brazil
Manuel Grana	University of Basque Country, Spain
Alberto Guillen	University of Granada, Spain
Barbara Hammer	University of Bielefeld, Germany
Francisco Herrera	University of Granada, Spain
Álvaro Herrero	University of Burgos, Spain
Michael Herrmann	University of Edinburgh, UK
Estevam R. Hruschka Junior	Federal University of São Carlos, Brazil
Ata Kaban	University of Birmingham, UK
Frank Klawonn	University of Applied Sciences Braunschweig/Wolfenbuettel, Germany
Andreas Koenig	Kaiserslautern University of Technology, Germany
Mario Koeppen	Kyushu Institute of Technology, Japan
Rudolf Kruse	University of Magdeburg, Germany
Wei-Chiang S. Hong	Oriental Institute of Technology, Taiwan
Pei Ling Lai	Southern Taiwan University, Taiwan
Paulo Lisboa	Liverpool John Moores University, UK
José Everardo B. Maia	State University of Ceará, Brazil

Roque Marín Morales	University of Murcia, Spain
Urszula Markowska Kaczmar	Wroclaw University of Technology, Poland
Yoan Miche	Aalto University, Finland
Tim Nattkemper	University of Bielefeld, Germany
Antonio Neme	Universidad Autónoma de la Ciudad de México, México
Erkki Oja	Aalto University, Finland
Vasile Palade	University of Oxford, UK
Carlos Pereira	University of Coimbra, Portugal
Jose Principe	University of Florida, USA
Ajalmar Rêgo Da Rocha Neto	Federal Institute of Ceará, Brazil
Agnaldo Reis	Federal University of Ouro Preto, Brazil
Bernardete Ribeiro	University of Coimbra, Portugal
Javier Sedano	University of Burgos, Spain
Ivan Silva	University of São Paulo, Brazil
Catarina Silva	Polytechnic Institute of Leiria, Portugal
Olli Simula	Aalto University, Finland
Alessandro Sperduti	University of Padua, Italy
Dante I. Tapia	University of Salamanca, Spain
Renato Tinós	University of São Paulo, Brazil
Alicia Troncoso	Pablo de Olavide University, Spain
Eiji Uchino	Yamaguchi University, Japan
Marc van Hulle	Katholieke Universitiet Leuven, Belgium
Marley Vellasco	Pontifical Catholic University of Rio de Janeiro, Brazil
Alfredo Vellido Alcacena	University Politécnica de Cataluña, Spain
Michel Verleysen	Universite Catholique de Louvain, Belgium
Fernando Von Zuben	State University of Campinas, Brazil
Lipo Wang	Nanyang Technological University, Singapore
Dongqing Wei	Shanghai Jiaotong University, China
Takashi Yoneyama	Technological Institute of Aeronautics, Brazil
Du Zhang	California State University, USA
Huiyu Zhou	Queen's University Belfast, UK
Rodolfo Zunino	University of Genova, Italy

Local Organizing Committee

Jose Alfredo Costa	DEE / Federal University, UFRN, Brazil
Marcos C. Madruga A. Pinheiro	Federal University, UFRN, Brazil
Bruno Motta de Carvalho	Federal University, UFRN, Brazil
Samuel Xavier de Souza	Federal University, UFRN, Brazil
Anne Magály de Paula Canuto	Federal University, UFRN, Brazil
Daniel Aloise	Federal University, UFRN, Brazil
Aarão Lyra	UnP, Brazil

Gustavo Fontoura	IFRN, Brazil
Teófilo C. Mattozo	UERN, Brazil
André P. Fernandes	UERN, Brazil
Nivaldo Vasconcelos	Uni-RN, Brazil
Cristhianne Linhares de Vasconcelos	Federal University, UFRN, Brazil
Jackson Gomes	ULBRA/UFRN, Brazil
Adriane Oliveira	SENAI/FIERN, Brazil
Gutembergue Soares da Silva	UFRN, Brazil Vic Rayward-Smith (Chair)

Special Session on Bio-inspired Models for Industrial and Medical Applications

Organizers

Emilio Corchado	University of Salamanca, Spain
Michal Wozniak	Wroclaw University of Technology, Poland
Vaclav Snasel	VSB-Technical University of Ostrava
Vicente Vera	Universidad Computense de Madrid, Spain

Program Committee

Álvaro Herrero	University of Burgos, Spain
Ana Belén Gil González	University of Salamanca, Spain
Angel Arroyo	University of Burgos, Spain
Arkadiusz Grzybowski	University of Applied Science in Legnica, Poland
Bartosz Krawczyk	
Belen Vaquerizo García	University of Burgos, Spain
Boguslaw Cyganek	University of Science and Technology, Kraków, Poland
Bruno Baruque	University of Burgos, Spain
Héctor Quintian	University of Salamanca, Spain
Hernando Silva Varela	University of Salamanca, Spain
Jan Martinovic	VSB-Technical University of Ostrava, Czech Republic
Jan Platos	VSB-Technical University of Ostrava, Czech Republic
Javier Sedano	University of Burgos, Spain
Jesús Ángel Román Gallego	University of Salamanca, Spain
José Luis Calvo	University of la Coruña, Spain
Katarzyna Wegrzyn-Wolska	ESIGETEL, Fontainebleau-Avon, France
Konrad Jackowski	
Laura Garcia	University of Cordoba, Spain
Leticia Curiel	University of Burgos, Spain

Mª Araceli Sánchez Sánchez	University of Salamanca, Spain
Michal Wozniak	Wroclaw University of Technology, Poland
Muhanned Al Farras	Gulf University, Bahrain
Pavel Kromer	VSB-Technical University of Ostrava, Czech Republic
Pedro Antonio Hernández Ramos	University of Salamanca, Spain
Petr Musilek	University of Alberta, Canada
Piotr Cal	
Piotr Sobolewski	
Raquel Redondo Guevara	University of Burgos, Spain
Robert Burduk	
Vicente Vera Gonzalvez	

Table of Contents

Special Session on Bio-inspired Models for Industrial and Medical Applications

Artificial Bee Colony Algorithm with Improved Explorations for Numerical Function Optimization

Mohammad Shafiul Alam[1], Md. Monirul Islam[2], and Kazuyuki Murase[3]

[1] Department of Computer Science and Engineering,
Ahsanullah University of Science and Technology, Dhaka 1208, Bangladesh
shuvo23@gmail.com
[2] Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology, Dhaka 1000, Bangladesh
mdmonirulislam@cse.buet.ac.bd
[3] Department of Human and Artificial Intelligence Systems,
University of Fukui, Fukui 910-8507, Japan
murase@synapse.his.fukui-u.ac.jp

Abstract. A major problem with Artificial Bee Colony (ABC) algorithm is its premature convergence to local optima, which originates from lack of explorative search capability of the algorithm. This paper introduces ABC with Improved Explorations (ABC-IX), a novel algorithm that modifies both the selection and perturbation operations of the basic ABC algorithm in an explorative way. Unlike the basic ABC algorithm, ABC-IX employs a probabilistic, explorative selection scheme based on simulated annealing which can accept both better and worse candidate solutions. ABC-IX also maintains a self-adaptive perturbation rate, separately for each candidate solution, to promote more explorations. ABC-IX is tested on a number of benchmark problems for numerical optimization and compared with several recent variants of ABC. Results show that ABC-IX often outperforms the other ABC-variants on most of the problems.

Keywords: Artificial bee colony algorithm, exploration and exploitation.

1 Introduction

The Artificial Bee Colony (ABC) algorithm [1] is a recently introduced swarm intelligence algorithm inspired by the intelligent food foraging behavior of honey bees. ABC and its variants have frequently showed superior performance in comparison to many other existing evolutionary and swarm intelligence algorithms [2]. Over the last few years, ABC has been successfully applied to wide and diverse range of problems, such as numerical optimization, discrete optimization, multi-objective optimization, machine learning, design of IIR filters, PID controller, software testing and so on [3].

In comparison to other greedy and local search based algorithms, ABC is more resilient against local optima, because the population of candidate solutions provides an advantage of preserving diversity and continuing explorations of the search space. However, from practical experiences, it is often found that the evolving population of

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 1–8, 2012.

candidate solutions loses its diversity and explorative capability too soon and the solutions prematurely get trapped around the locally optimal points of the fitness landscape. Another problem often found with ABC is the fitness stagnation, where the population of solutions stops progressing towards the global optimum for no apparent reason, even without converging to some local optima [4]. The risk of fitness stagnation and premature convergence usually depends on the mix of explorative and exploitative operations during the search. Similar to other population based metaheuristic algorithms, ABC drives its search towards global optimum with two operators – perturbation and selection. The perturbation operation is generally responsible for explorations of different regions of the search space by random alteration of the existing candidate solutions, while the fitness based selection operation of ABC performs the exploitation of the search regions explored so far.

There exist a number of recent works (e.g. [5-8]) that attempt to alter the explorative and/or exploitative properties of the standard ABC algorithm to avoid premature convergence and fitness stagnation. However, most of them focus on altering the perturbation operation only. In the literature, not much has been reported to improve the greedy fitness-based selection procedure of ABC. Our proposed algorithm – ABC with Improved Explorations (ABC-IX) alters both the selection and perturbation operations of the standard ABC algorithm to increase the explorative capacity of both the operations. A detailed description of both the improvements by ABC-IX is presented in section 3. Increased explorations can spread the population across more search regions allowing more possible trial solutions to be produced, which make ABC-IX robust against fitness stagnation and premature convergence.

The rest of this paper is organized as follows. Section 2 describes the standard ABC algorithm with its pseudocode. Section 3 presents the proposed algorithm – ABC-IX and explains the improvements on the selection and perturbation operations along with their pseudocode. Section 4 provides details of the benchmark problems, parameter settings of the algorithms, comparison of their results and experiments on ABC-IX to investigate its improvements. Finally, section 5 concludes with a few new directions for further study with ABC-IX.

2 Artificial Bee Colony (ABC) Algorithm

The ABC algorithm mimics the food foraging behavior of honey bees with three groups of bees that are found in nature, i.e. employed, onlooker and scout bees. A bee working to forage a particular food source (i.e., candidate solution) previously visited by itself and searching only around its vicinity is called an employed bee. Onlooker bees randomly pick and follow any of the employed bees. The probability of picking an employed bee is proportional to the quality of its food source. Scout bees perform random explorations of the search space to find new food sources. Suppose $\boldsymbol{x_i}$ is a food source currently being visited by an employed bee. The bee employs (1) to search around the neighborhood of $\boldsymbol{x_i}$ in order to produce a new, trial food source $\boldsymbol{v_i}$.

$$v_{ij} = x_{ij} + \varphi_{ij}\left(x_{kj} - x_{ij}\right) \tag{1}$$

Here, $j \in \{1, 2, \ldots, D\}$ and $k \in \{1, 2, \ldots, SN\}$ are randomly picked indices, D is the dimensionality of the problem, SN is the number of food positions, φ_{ij} is a uniform random value produced from [-1, 1]. If $\boldsymbol{v_i}$ has better 'fitness' than the old food source position $\boldsymbol{x_i}$, then $\boldsymbol{x_i}$ is replaced by $\boldsymbol{v_i}$. Else, $\boldsymbol{v_i}$ is discarded. With f being the function to be minimized, ABC computes the 'fitness' of a candidate solution, say $\boldsymbol{x_i}$, using (2).

$$fitness(\boldsymbol{x_i}) = \begin{cases} 1/\big(1 + f(\boldsymbol{x_i})\big), & if\ f(\boldsymbol{x_i}) \geq 0 \\ 1 + |f(\boldsymbol{x_i})| & otherwise \end{cases} \tag{2}$$

An onlooker bee randomly picks an employed bee $\boldsymbol{x_i}$ to follow and forages around the vicinity of its food source. The probability p_i that a food source $\boldsymbol{x_i}$ would be picked by an onlooker bee is computed using (3), which makes the probability p_i proportional to $fitness(\boldsymbol{x_i})$. This ensures that a more promising solution $\boldsymbol{x_i}$ with high fitness value is often foraged and exploited more intensively by (1) than relatively less fit solutions.

$$p_i = fitness(\boldsymbol{x_i}) / \sum_{n=1}^{SN} fitness(\boldsymbol{x_n}) \tag{3}$$

If a particular food source position $\boldsymbol{x_i}$ has not been improved over the last *limit* cycles, then it is abandoned and the bee employed to $\boldsymbol{x_i}$ now becomes a scout bee that is placed at random across the search space using (4), where $j = 1, 2, \ldots, D$ and $[min_j, max_j]$ is the search space along the j-th dimension.

$$x_{ij} = min_j + rand(0,1) * (max_j - min_j) \tag{4}$$

Algorithm: Artificial Bee Colony (ABC) Algorithm

1: Initialize a population of *SN* food source positions (i.e., candidate solutions) $\boldsymbol{x_i}$, for i = 1, 2, ..., *SN*. Each $\boldsymbol{x_i}$ is a vector of *D* parameters: $\boldsymbol{x_i} = [x_{i1}, x_{i2}, \ldots, x_{iD}]^T$
2: Evaluate the fitness of each food source positions using (2)
3: **repeat**
4: Perturb each food position $\boldsymbol{x_i}$ to produce a new position $\boldsymbol{v_i}$ using (1)
5: Evaluate each new solution $\boldsymbol{v_i}$ using (2). If $\boldsymbol{v_i}$ is better than $\boldsymbol{x_i}$, then accept $\boldsymbol{v_i}$ to replace $\boldsymbol{x_i}$
6. Calculate the probability p_i associated with each food source position $\boldsymbol{x_i}$ using (3)
7: For each onlooker bee, assign it to a food source $\boldsymbol{x_i}$, proportionally based on the probability p_i
8: Produce new food position $\boldsymbol{v_i}$ by perturbing the food source $\boldsymbol{x_i}$ of each onlooker bee using (1)
9: Evaluate each new solution $\boldsymbol{v_i}$ by (2). If $\boldsymbol{v_i}$ is better than $\boldsymbol{x_i}$, then accept $\boldsymbol{v_i}$ to replace $\boldsymbol{x_i}$
10: If a food source has not improved during the last *limit* cycles, then abandon it and replace it with a new randomly placed scout bee with its food source $\boldsymbol{x_i}$ produced by (4).
11: Memorize the best food source position found so far
12: Set cycle counter $C = C + 1$
13: **until** C = Maximum cycle number (*MCN*)
14: **return** the best food source position (i.e., candidate solution) found so far

Fig. 1. Pseudocode for the standard Artificial Bee Colony (ABC) algorithm

3 ABC Algorithm with Improved Explorations (ABC-IX)

The proposed algorithm – ABC-IX differs from the standard ABC algorithm in two important ways. First, ABC accepts a newly produced solution $\boldsymbol{v_i}$ only if $\boldsymbol{v_i}$ has higher fitness value than the original solution $\boldsymbol{x_i}$ (Fig. 1, steps 5,9). This exploitative selection

Algorithm: Simulated Annealing (SA) based Probabilistic Selection Scheme for ABC-IX
input: Two candidate solutions $\boldsymbol{x_i}$ and $\boldsymbol{v_i}$
output: Either of $\boldsymbol{x_i}$ and $\boldsymbol{v_i}$, selected to be included into the population
begin
  if $fitness(\boldsymbol{v_i}) \geq fitness\ (\boldsymbol{x_i})$ **then**
    return $\boldsymbol{v_i}$
  else
    $\Delta E = fitness(\boldsymbol{x_i}) - fitness\ (\boldsymbol{v_i})$
    if $rand(0,1)\ \leq \exp(-\Delta E/T)$ **then**
      return $\boldsymbol{v_i}$
    endif
  endif
  return $\boldsymbol{x_i}$
end

Algorithm: Perturbation by ABC-IX
input: An existing candidate solution $\boldsymbol{x_i}$
output: Perturbed candidate solution $\boldsymbol{v_i}$
begin
  $\boldsymbol{v_i} = \boldsymbol{x_i}$
  if $rand(0,1) \leq t$ **then**
    $q[\boldsymbol{v_i}] = q_{min} + (1.0 - q_{min})\ *\ rand(0,1)$
  else
    $q[\boldsymbol{v_i}] = q[\boldsymbol{x_i}]$
  endif
  for $j = 1$ to D **do**
    if $rand(0,1) \leq q[\boldsymbol{v_i}]$ **then**
      $k = rand\{1, 2, ..., SN\}$
      $\varphi_{ij} = rand(-1,1)$
      $v_{ij} = x_{ij} + \varphi_{ij}\ *\ (x_{kj} - x_{ij})$
    endif
  enddo
  return $\boldsymbol{v_i}$
end

Fig. 2. Pseudocode for simulated annealing (SA) based probabilistic selection scheme (on the left) and perturbation with self-adaptive perturbation rate (on the right) for ABC-IX

Algorithm: Artificial Bee Colony Algorithm with Improved Explorations (ABC-IX)

1: Initialize the population of SN food source positions (i.e., candidate solutions) $\boldsymbol{x_i}$, for $i = 1, 2, ..., SN$. Each $\boldsymbol{x_i}$ is a vector of D parameters: $\boldsymbol{x_i} = [x_{i1}, x_{i2}, ..., x_{iD}]^T$
2: Evaluate the fitness of each food source positions using (2)
3: **repeat**
4*: Perturb each food source $\boldsymbol{x_i}$ to produce a new food $\boldsymbol{v_i}$ using the pseudocode for perturbation in Fig. 2
5*: Evaluate each new solution $\boldsymbol{v_i}$ by (2). Select either $\boldsymbol{x_i}$ or $\boldsymbol{v_i}$ by SA-based probabilistic selection (Fig. 2)
6: Calculate the probability p_i associated with each food source position $\boldsymbol{x_i}$ using (3)
7: For each onlooker bee, assign it to a food source $\boldsymbol{x_i}$, proportionally based on the probability p_i
8*: Produce $\boldsymbol{v_i}$ by perturbing the food position $\boldsymbol{x_i}$ of each onlooker bee with the perturbation code (Fig. 2)
9*: Evaluate each new solution $\boldsymbol{v_i}$ by (2). Select either $\boldsymbol{x_i}$ or $\boldsymbol{v_i}$ by SA-based probabilistic selection (Fig. 2)
10: If a food source has not improved during the last *limit* cycles, then abandon it and replace it with a new, randomly placed scout bee with its food source $\boldsymbol{x_i}$ produced by (4).
11: Memorize the best food source position found so far
12*: Set cycle counter $C = C + 1$ and system temperature T = α * T
13: **until** C = Maximum cycle number (MCN)
14: **return** the best food source position (i.e., candidate solution) found so far

Fig. 3. Pseudocode for ABC-IX. Steps that differ from the ABC algorithm are marked with '*'

scheme denies any possible downhill movement and allows only uphill steps in the fitness landscape, which may lead to local optima. In contrast, ABC-IX promotes more search space explorations by probabilistically allowing some downhill steps using a simulated annealing-based selection scheme (Fig. 2). Second, ABC perturbs only a single parameter of each existing solution $\boldsymbol{x_i}$ which usually produces the new solution $\boldsymbol{v_i}$ in the neighborhood of $\boldsymbol{x_i}$, which is exploitative. In contrast, ABC-IX can perturb any number of parameters of $\boldsymbol{x_i}$ by introducing a self-adaptive perturbation rate for each individual solution $\boldsymbol{x_i}$, which is addressed as $q[\boldsymbol{x_i}]$ (Fig. 2). The value of $q[\boldsymbol{x_i}]$ is self-adapted gradually, cycle by cycle, separately for each $\boldsymbol{x_i}$. Fig. 3 presents

the pseudocode for ABC-IX, which differs from ABC in how the existing solutions are perturbed (steps 4, 8) and how the new solutions are selected into the population (steps 5, 9). They are further elaborated in the following paragraphs.

3.1 Simulated Annealing (SA) Based Probabilistic Selection

SA accepts both better and worse new solutions, but the probability of accepting a worse new solution is reduced over time, depending on a gradually decreasing control parameter T (i.e. temperature, from the analogy to the real annealing procedure in metallurgy). Given an initial candidate solution $\boldsymbol{x_i}$, SA randomly perturbs $\boldsymbol{x_i}$ to produce $\boldsymbol{y_i}$ in the neighborhood of $\boldsymbol{x_i}$. The change ΔE of the objective function value, $\Delta E = obj(\boldsymbol{x_i}) - obj(\boldsymbol{y_i})$. If $\boldsymbol{y_i}$ is better than $\boldsymbol{x_i}$ (i.e. $\Delta E < 0$), SA accepts $\boldsymbol{y_i}$ as its current state and discards $\boldsymbol{x_i}$. If $\boldsymbol{y_i}$ is worse (i.e. $\Delta E > 0$), SA may still accept $\boldsymbol{y_i}$, but only with probability = $e^{\frac{-\Delta E}{T}}$. SA usually starts with a high initial temperature T_0 to ensure high degree of initial explorations by frequently accepting worse solutions (larger T makes $\frac{\Delta E}{T} \to 0$, thus probability $e^{\frac{-\Delta E}{T}} \to 1$). As T gradually decreases, SA becomes increasingly exploitative, accepting better solutions only. The SA-based probabilistic selection scheme (Fig. 2) improves the explorative capacity of ABC-IX, because it can now accept both better and worse solutions to be more resilient against local optima. T is gradually decreased by the exponential cooling schedule: $T(t+1) = \alpha * T(t)$. We used $\alpha = 0.99$. The initial temperature T_0 is set to 50 times of the difference of fitness values of the best and the worst solutions of the first generation.

3.2 Self-adaptive Perturbation Rate

ABC perturbs only a single, random parameter of the existing solutions using (1). This allows search along a single dimension at a time, which may be suitable for separable problems, but is inappropriate for non-separable problems. In contrast, ABC-IX can perturb any number of parameters allowing search along any possible direction. For every candidate solution $\boldsymbol{x_i}$, ABC-IX separately maintains and adapts a control parameter $q[\boldsymbol{x_i}]$, which denotes the probability of each parameter of $\boldsymbol{x_i}$ to be perturbed by the perturbation operation. Note that, in the pseudocode (Fig. 2), the value of $q[\boldsymbol{x_i}]$ is perturbed first, with probability t, using (5), before perturbing any parameter of $\boldsymbol{x_i}$ to produce $\boldsymbol{v_i}$. This perturbed value $q[\boldsymbol{v_i}]$ is then used as the perturbation rate (PR) when producing $\boldsymbol{v_i}$ from $\boldsymbol{x_i}$. A better value of PR is supposed to lead towards better offspring solution $\boldsymbol{v_i}$, which is more likely to survive than $\boldsymbol{x_i}$ and produce better, new trial solutions and hence, propagate the better values of the PR. Thus the gradual self-adaptation towards better, more effective PR values takes place across the population. We have used $t = 0.10$, $q_{max} = 1.0$ and $q_{min} = 1/D$.

$$q[\boldsymbol{v_i}] = \begin{cases} q_{min} + rand(0,1) * (q_{max} - q_{min}), & \text{if } rand(0,1) < t \\ q[\boldsymbol{x_i}] & \text{otherwise} \end{cases} \quad (5)$$

4 Experimental Studies

ABC-IX is evaluated using a suite of 13 standard benchmark problems [9] for numerical optimization and compared with some recent variants of the ABC algorithm. For page limit, the benchmark suite functions are not presented here, but they can be readily found as f_1–f_{13} in [9]. These functions have been widely used in many recent studies on evolutionary and swarm intelligence algorithms. The suite contains both unimodal (f_1–f_7) and multimodal (f_8–f_{13}) functions. A function is called multimodal if it has multiple local optima. To optimize such a function, the search algorithm must possess both exploitative and explorative properties so that it can explore the locally optimal points without being trapped anywhere and thus keep exploring further for better unvisited search regions. Some of the multimodal functions can have hundreds of local minima, even when the dimensionality is just two or three (e.g., Griewank function f_{11}). Number of local optima usually increases exponentially with the number of dimensions, which often makes their minimization extremely difficult.

ABC-IX has been compared with the standard ABC algorithm and some other ABC variants [5-8] that try to alter the explorative/exploitative properties of ABC, such as ABC with self-adaptive mutation (ABC-SAM) [5], cooperative ABC (CABC) [6], ABC with diversity strategy (DABC) [7] and global best guided ABC [8]. On each function, ABC-IX has made 30 independent runs and the mean of the best results are presented in Tables 1–5. At first, ABC-IX is compared with the basic ABC [1] and ABC-SAM [5] in Table 1 with $SN = 50$, maximum no. of function evaluations FE = 100,000, no. of employed and onlooker bees = $SN/2$, no. of scout bees = 1 and $limit$ = 100. Table 1 shows that ABC-IX outperforms both ABC and ABC-SAM on most of the functions. In comparison to ABC, ABC-IX performs either better (on 11 out of the 13 functions) or at least equally well (f_6 and f_{13}). The other algorithm, ABC-SAM performs better than ABC-IX on one function only (f_8), while ABC-IX performs better on as many as 11 other functions. ABC-SAM performs better than the basic ABC, because it uses two different distributions, one exploitative and the other explorative, ensuring both explorations and exploitations. But it still uses an exploitative, fitness-based selection, which may be the reason that the more explorative ABC-IX often outperforms it. Fig. 4 shows that ABC-IX starts with more explorations and hence, reduced convergence speed than ABC, but gradually ABC-IX becomes more exploitative and achieves higher convergence speed. For f_9, the basic ABC is trapped during the last half of its execution, while ABC-IX shows no sign of stagnation.

In Table 2, ABC-IX is compared with two cooperative ABC variants – CABC_S and CABC_H [6]. Both the variants enforce more explorations by decomposing the search space and using multiple bee colonies to explore the multiple subspaces. For comparison, ABC-IX is implemented with the same settings [6]. Results show that ABC_IX outperforms CABC on four out of the six functions, while the CABC variants perform better on the remaining two functions only.

The next two comparisons of ABC-IX are with DABC [7] and GABC [8]. DABC tries to ensure more search space explorations by preserving more population diversity, while GABC alters the basic perturbation formula (1) in an exploitative way by using the global best solution found so far. ABC-IX is re-implemented with the same settings as suggested in [7] and [8]. Tables 3-4 show that ABC-IX often outperforms DABC (2 out of 4 functions) and GABC (4 out of 5 functions). The reason may be

that GABC is too exploitative, while DABC uses a naïve strategy of repeated switching that may cause frequent oscillations between exploitations and explorations.

For more investigations, we have designed two more variants of ABC – ABC-SimAn and ABC-SAD. ABC-SimAn includes the simulated annealing based selection but excludes the self-adaptive perturbation. In contrast, ABC-SAD includes the self-adaptive perturbation, but does not use the explorative simulated annealing based selection. Both of them are tested on the six multimodal functions f_8–f_{13} with SN = 100 and MCN = 1000. Results (Table 5) show that both ABC-SimAn and ABC-SAD outperform the basic ABC on most (5 out of 6) of the functions. This indicates the necessity of more explorations. Furthermore, ABC-IX, which combines both the explorative techniques, outperforms both ABC-SimAn and ABC-SAD on all the functions, which indicates that the synergy and interaction between the explorative selection and perturbation operations can be further useful to improve the results.

5 Suggestions for Further Study

There may be several future research directions suggested by this study. Firstly, ABC-IX uses a simple exponential cooling schedule for the system temperature T. A more sophisticated cooling strategy (e.g., a strategy parameterized by the population diversity or current explorative/exploitative requirement) may be more effective. Secondly, ABC-IX concentrates only on the explorative capacity of the algorithm. Putting some emphasis to control the exploitations may further improve the results. Thirdly, the quality of the final solution might be improved further by using an efficient and exploitative local searcher. Finally, ABC-IX has been applied mainly on continuous optimization problems. It would be interesting to study how well ABC-IX performs on many other existing problems, especially the discrete and real world ones.

Table 1. Comparison of ABC [1], ABC-SAM [5] and ABC-IX. Boldface font marks the best performance for each function.

No	ABC	ABC-SAM	ABC-IX
f_1	3.58E-11	4.18E-14	**2.86E-38**
f_2	1.04E-14	2.47E-08	**6.52E-18**
f_3	2.75E-10	3.95E-12	**1.86E-36**
f_4	9.37E+00	1.69E+01	**1.17E-02**
f_5	2.75E+00	4.27E-02	**1.95E-01**
f_6	**0**	**0**	**0**
f_7	8.61E-13	3.66E-16	**1.64E-63**
f_8	3.49E+02	**1.53E+02**	1.56E+02
f_9	5.79E-15	1.26E-16	**6.14E-41**
f_{10}	3.08E-06	9.26E-08	**3.82E-15**
f_{11}	4.35E-08	8.36E-10	**9.70E-40**
f_{12}	5.82E-08	2.78E-12	**7.40E-14**
f_{13}	**2.64E-03**	2.96E-01	**2.61E-03**

Table 2. Comparison of CABC variants [6] and ABC-IX. Best results are in boldface font.

No	CABC_S	CABC_H	ABC-IX
f_1	3.30E-19	5.92E-18	**9.41E-48**
f_5	6.33E+00	4.80E-01	**3.21E-07**
f_8	**1.30E-04**	**1.27E-04**	1.86E-01
f_9	**0**	**0**	3.86E-52
f_{10}	1.83E-14	8.35E-15	**1.14E-16**
f_{11}	4.42E-02	7.96E-03	**3.85E-51**

Table 3. Comparison between DABC [7] and ABC-IX. Best results are in boldface font.

No	DABC	ABC-IX
f_1	1.08E-16	**2.82E-63**
f_5	**1.55E-05**	**1.59E-05**
f_9	**0.00E-306**	4.29E-73
f_{11}	1.11E-16	**7.60E-67**

Table 4. Comparison between GABC [8] and ABC-IX. Best results are shown in boldface font.

No	*D*	GABC	ABC-IX
f_1	30	4.17E-16	**2.75E-107**
f_5	2	**1.68E-04**	2.93E-03
f_9	30	9.47E-15	**9.20E-107**
f_{10}	30	3.21E-14	**4.14E-15**
f_{11}	30	2.96E-17	**5.68E-105**

Table 5. Comparison of ABC [1], ABC-SimAn, ABC-SAD and ABC-IX. Best results in bold.

No	ABC	ABC-SimAn	ABC-SAD	ABC-IX
f_8	7.52E+02	5.29E+02	8.90E+01	**2.15E+01**
f_9	5.02E-14	2.53E-17	6.85E-18	**8.71E-20**
f_{10}	3.04E-07	8.50E-09	8.37E-10	**8.89E-11**
f_{11}	9.84E-10	1.91E-19	3.78E-19	**2.07E-23**
f_{12}	7.18E-10	7.55E-14	5.09E-14	**1.91E-14**
f_{13}	2.63E-03	2.64E-03	2.63E-03	**2.61E-03**

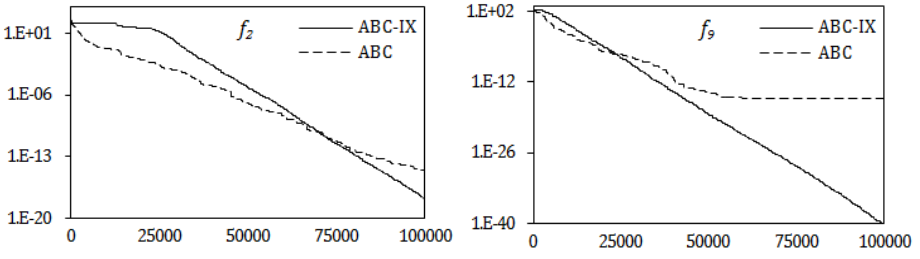


Fig. 4. Convergence of ABC and ABC-IX for the functions f_2 and f_9. The vertical axis shows the function value, while the horizontal axis is the number of function evaluations.

References

1. Karaboga, D., Basturk, B.: On the Performance of Artificial Bee Colony (ABC) Algorithm. Applied Soft Computing 8, 687–697 (2008)
2. Karaboga, D., Akay, B.: A Comparative Study of Artificial Bee Colony Algorithm. Applied Mathematics and Computation 214, 108–132 (2009)
3. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A Comprehensive Survey: Artificial Bee Colony (ABC) Algorithm and Applications. Artificial Intelligence Review, Online First™ (2012), doi:10.1007/s10462-012-9328-0
4. Lampinen, J., Zelinka, I.: On Stagnation of the Differential Evolution Algorithm. In: Proc. 6th International Mendel Conf. on Soft Computing, Czech Republic, pp. 76–83 (2000)
5. Alam, M.S., Islam, M.M.: Artificial Bee Colony Algorithm with Self-Adaptive Mutation: A Novel Approach for Numeric Optimization. In: Proceedings of the 2011 Int. Conf. on Trends and Developments in Converging Technology, Bali, Indonesia, pp. 49–53 (2011)
6. Abd, M.E.: A cooperative Approach to the Artificial Bee Colony algorithm. In: IEEE Congress on Evolutionary Computation (CEC), Barcelona, Spain, pp. 1–5 (2010)
7. Lee, W.P., Cai, W.T.: A Novel Artificial Bee Colony Algorithm with Diversity Strategy. In: Proc. of the 7th Int. Conf. on Natural Computation, China, pp. 1441–1444 (2011)
8. Zhu, G., Kwong, S.: Gbest-guided Artificial Bee Colony Algorithm for Numerical Function Optimization. Applied Mathematics & Computation 217, 3166–3173 (2010)
9. Yao, X., Liu, Y., Lin, G.: Evolutionary Programming Made Faster. IEEE Transactions on Evolutionary Computation 3, 82–102 (1999)

Extensions of Ant-Miner Algorithm to Deal with Class Imbalance Problem

Murilo Zangari, Wesley Romão, and Ademir Aparecido Constantino

Universidade Estadual de Maringá, Brazil
{murilo.zangari,wesley.uem,ademir.uem}@gmail.com

Abstract. A database has class imbalance when there are more cases of one class then the others. Classification algorithms are sensitive of this imbalance and tend to valorize the majority classes and ignore the minority classes, which is a problem when the minority classes are the classes of interest. In this paper we propose two extensions of the Ant-Miner algorithm to find better rules to the minority classes. These extensions modify, mainly, how rules are constructed and evaluated. The results show that the proposed algorithms found better rules to the minority classes, considering predictive accuracy and simplicity of the discovered rule list.

Keywords: Data Mining, classification, class imbalance, Ant-Miner.

1 Introduction

Database with class imbalance problem is one of the relatively new problems that emerged when machine learning matured from an embryonic science to an applied technology, broadly used in the worlds of business, industry and scientific research [4]. This increase in interest gave rise to two workshops held in 2000 at AAAI (Association of the Advancement of Artificial Intelligence) conference [2] and other in 2003 at ICML (International Conference on Machine Learning) conference [3] and a special edition of ACM SIGKDD Explorations in 2004 [4] [5] [6].

In the classification task, a database is deemed imbalanced when there are many more cases of some class then the others. Many classification algorithms (e.g., C4.5) tend to ignore the minority classes. In real-world scenarios, the ratio of the small to the large classes can be drastic such as 1 to 100, 1 to 1.000, 1 to 10.000 and sometimes even greater [4] [5] [7].

Several techniques to solve the class imbalance problem have been proposed both at the data and algorithmic levels. At data levels, these solutions include many forms of re-sampling [4] [5] [6] [8]. At the algorithmic level, solutions include adjusting the cost of the various classes so as to counter the class imbalance or learning from one class [3] [4].

This paper proposes two extensions to the Ant-Miner classification algorithm [1] that address the class imbalance issue, thus finding better rules to minority classes with a better predictive accuracy and comprehensibility.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 9–18, 2012.

2 Related Work

The Ant-Miner [1] is a classification algorithm that aims to solve the classification task of Data Mining. This algorithm is based on the ACO (ant colony optimization) metaheuristic [10] [11], in which each ant incrementally builds/modifies a solution to a certain problem, in this case the problem is the classification rule discovery.

Parpinelli et al. [1] evaluated the Ant-Miner with six databases from UCI (University of California at Irvine) [19]. The algorithm was compared with other classification algorithms and achieves good results. The results showed that the Ant-Miner is a promising technique for classification rule discovery. Several researches and modifications have been proposed to improve efficiency: Ant-Miner2 [12], Ant-Miner3 [13], Ant-Miner+ [14], cAnt-Miner [15], Multiple pheromone types Ant-Miner [17].

3 The Algorithms

Most of Ant-Miner extensions aim to improve the predictive accuracy. In this paper we present two Ant-Miner extensions that aim to find better rules when applied on databases with class imbalances. Such algorithm should find rules to minority classes, since classification algorithms are sensitive of this kind of imbalance and tend valorize the majority classes, which is a problem when the minority classes are the interest classes. Both are developed in Java on Eclipse development environment. The only difference between the two algorithms proposed is how the rule quality is calculated.

3.1 The Ant-MinerCI Algorithm

The Ant-MinerCI (Ant-Miner Class Imbalance) receives databases in the ARFF file format, the same format used in Weka [18]. Figure 1 shows the pseudo-code of the Ant-MinerCI algorithm.

```
Ant-MinerCI:
TrainingSet = {all the training cases};
DiscoveryRuleList = []; //initialized with an empty list
Calculate the heuristic value for each term
<attribute=value> related to interesting_class;
WHILE (TrainingSet > Max_uncovered_cases)
  i = 1; //ant index
  j = 1; //convergence test index
  Initialize all terms with the same amount of pheromone;
  REPEAT
    IF (First iteration of WHILE)
      Ant_i add the interesting class on the consequent;
    END IF
    WHILE (No. of antec. on rule Ant_i ≤ max_num_antec)
      Ant_i starts with an empty rule and incrementally
      constructs a classification rule R_i by adding one
      term at a time to the current rule;
    END WHILE
```

Fig. 1. Pseudo-code *Ant-MinerCI*

```
    Update the pheromone of all terms by increasing pher-
    omone in the terms used by Ant_i proportional to the
    quality of rule R_i and decreasing pheromone in the
    others terms.
    IF (R_i is equal to R_i-1)//update convergence test
      THEN j = j +1;
      ELSE j = 1;
    END IF
    i = i + 1;
  UNTIL (i >= no_of_Ants) OR (j >= no_rules_converg)
Choose the best rule R_best among all the rules R_i;
Add the rule R_best to DiscoveredRuleList;
TrainingSet=TrainingSet−{set of cases correctly covered by R_best};
END WHILE
Add the pattern rule at the end of DiscoveredRuleList;
```

Fig. 1. (*continued*)

Initially the RuleList is empty; the TrainingSet has all the cases of training. Each iteration of the first WHILE loop of Algorithm1, corresponding to a number of executions of the REPEAT-UNTIL loop, discovers one classification rule. This rule is added to the list of discovered rules and the training cases that are covered correctly by this rule (i.e., cases satisfying the rule antecedent and having the class predicted by the rule consequent) are removed from the training set. This process is performed iteratively while the number of uncovered training cases is greater than a user-specified threshold, called max_uncovered_cases.

Each iteration of the REPEAT_UNTIL loop consists of two steps, comprising rule construction and pheromone updating. In the rule construction, an Ant_i starts with an empty rule, i.e., a rule with no terms in its antecedent, and adds one term at time to its current partial rule. The current partial rule constructed by an ant corresponds to the current partial path followed by that ant. The choice of the term to be added to the current partial rule depend on both a problem-dependent heuristic function (η) and on the amount of pheromone (τ) associated with each term.

The amount of pheromone in the trail followed by Ant_i (according to the quality of rule R_i) and decreasing the pheromone in the other trails (simulating the pheromone evaporation). Then another ant starts construct its rule, using the new amount of pheromone to guide its search.

3.1.1 Heuristic Value and Pheromone Update

Ant-Miner2 [12] uses the Equation (1) to calculate the heuristic value for each $term_{ij}$ (attribute,value). The heuristic value is related to how often a term occurs with the majority class, so that the ants tend to find the paths (terms) that lead to the majority class more interesting.

$$\eta_{ij} = \frac{|majority\_class T_{ij}|}{|T_{ij}|} \tag{1}$$

Where:

- T_{ij} is how often the $term_{ij}$ occurs in the training set;
- $majority\_class T_{ij}$ is how often the majority class occurs with the $term_{ij}$ in training set.

The heuristic value with each $term_{ij}$ used by Ant-MinerCI is not calculated with the majority class, it is calculated with the interest class. The interest class can either be the minority class or not, defined by Equation (2).

$$\eta_{ij} = \frac{|CI \,\&\, T_{ij}|}{|T_{ij}|} \tag{2}$$

Where:

- T_{ij} is how often that the $term_{ij}$ occurs in the training set;
- $CI \,\&\, T_{ij}$ is how often the interest class occurs with each $term_{ij}$.

The idea is that the ants have a preference for terms that are most relevant to find rules for the interesting class. But this is not enough, with the ants iterations and the pheromone update, the ants tend to converge to paths that find rules to the majority class. The initial amount of pheromone and update pheromone with each term are the same as the original Ant-Miner [1] [12].

3.1.2 Rules Construction

In the original Ant-Miner, the rule consequent (class) is chosen after the antecedent construction. In the Ant-Miner+ [14], the consequent is probabilistically chosen first, the consequent is selected according to the pheromone value associated, and this pheromone indicates which class contributed the most to the rule construction.

In Ant-MinerCI, during the first iteration to find the best rule, the algorithm selects the consequent first, however not in a probabilistic way, obligatorily selecting the interest class, thus the ants obligatorily select terms to add to the partial rule that maximize the rule quality that has a consequent with the interest class. In other words, the ants tend to converge to shorter paths (best rules) with the interest class. This interest class must be set by the user before the algorithm starts the rules construction.

The process to add terms to the current rule stops when one of the following conditions is met:

- Any term to be added to the rule would make the rule cover a number of cases that is smaller than a user-specified threshold, called `min_cases_per_rule`;
- All attributes have been already used by the current ant;

In Ant-MinerCI there is one more condition to stop the terms addition, which is `max_num_antec`. Such parameter limits the number of antecedents that a rule can have, since rules that have a large number of terms in the antecedent part can difficult the comprehensibility of the rules.

After the first iteration of the While loop (external) at Figure 1, the best rule R_{best} among all the R_i (that have the interest class) is selected and added at the discovered

rules list. The next iterations of the While loop, the algorithm can find rules for every class. The formula to calculate the rule quality is the same that original Ant-Miner, Q = sensitivity × specificity, defined as Equation (3).

$$Q = \left(\frac{TP}{TP + FN}\right) \times \left(\frac{TN}{FP + TN}\right) \tag{3}$$

Where:

- TP (true positives): the number of cases covered by the rule antecedents that have the class predicted by the rule;
- FP (false positives): the number of cases covered by the rule antecedents that have a class different from the class predicted by the rule;
- FN (false negatives): the number of cases that are not covered by the rule antecedents but that have the class predicted by the rule;
- TN (true negatives): the number of cases that are not covered by the rule antecedents and that do not have the class predicted by the rule;

The discovered ordered rule list by the algorithm will have at least one discovered rule for the interesting class, even if the interest class is the minority class, which is the most difficult discovery.

3.1.3 Ant-MinerCI Parameters

The conventional parameters of Ant-Miner are: `no_of_ants` (the maximum number of candidate rules during an iteration of the While loop), `min_cases_per_rule` (each rule must cover a minimum number of cases from the training set and this parameter helps to avoid overfitting), `max_uncovered_cases` (the maximum number of cases uncovered from the training set, so that the algorithm discovers rules until the number of training cases is smaller than this threshold), `no_rules_converg` (the number of consecutives equals rules, then the algorithm concludes that the ants have converged to a single rule).

Beyond the conventional parameters, the Ant-MinerCI has two others: `interesting_class` (the user specifies which is the interest class and that will be used to discover the first rule), `max_no_antec` (the maximum number of terms that a rule can have in the antecedent part).

3.2 The Ant-MinerCIP Algorithm

The discovered rules from the minority classes have few covered cases considering all the training cases. When the function Q = sensitivity × specificity is used, the tendency is that the rules have a reasonable number of covered cases considering all the training cases.

The data set Letter-a [19] has two classes, the class “A” and the class “others”. The distribution of this data set is 96.05% to class “others” and 3.95% to class “A”. Most rule induction algorithms probably will find rules only to “others” class, but the most interesting is at 3.95% of class “A”. Even the rules discovered by Ant-MinerCI for the minority class, this rules can have a low predictive accuracy using Q = sensitivity ×

specificity. For example: this data set, Letter-a, has 20.000 cases. An ant finds a rule R_i (for class "A") that cover 100 cases, and among these cases, TP = 80, FP = 20, FN = 709 and TN = 19191. Using Q = sensitivity × specificity, has the following value:

$$Q = \left(\frac{TP}{TP + FN}\right) \times \left(\frac{TN}{FP + TN}\right) = \left(\frac{80}{80 + 709}\right)\left(\frac{19191}{20 + 19191}\right) = 0{,}097 \quad (4)$$

The value from Equation (4) is the quality value to R_i, the probability for this rule to be chosen as the best rule by algorithm is low, even if this rule has a predictive accuracy of 80%. This happens because the algorithm selects the best rule when the Q value is high, in other words, the algorithm select the rule that has high coverage (TP and FP). In this case, that there is a rare class, the discovered rules for this class can have a low predictive accuracy, which harms all the predictive accuracy and the relevant rule discovery.

The Ant-MinerCIP (Ant-Miner Class Imbalance Precision) algorithm uses the precision formula directly, Equation (5), which evaluates rule quality, since the precision aims to analyze only the covered cases by the rule. The lower the FP number covered by the rule, better the precision.

$$\text{Precision} = \left(\frac{TP}{TP+FP}\right) \quad (5)$$

Thus, the precision metric does not harm the discovery of rules to the minority class, as well as for other classes. A situation that can be happen is that the rule may be too adjusted, which characterizes overffiting. To avoid this problem, the `min_cas_per_rule` parameter must be used, which matches the minimum value of cases that a rule can cover. Thus adjusting this parameter may avoid overffiting.

4 Results and Analysis

4.1 Ant-MinerCI Evaluation

The algorithm was evaluated using public-domain data sets from University of California at Irvine repository [19]. Table 1 describes the data sets, the names, number of cases, the number of attributes and number of classes. The data sets Letter-a and Letter-vowel were constructed based on the original data set Letter [6].

Table 1. Data sets used to evaluate the algorithms

Data sets	No. of cases	No. of attributes	No. of classes
Ljubljana breast cancer	286	9	2
Wisconsin breast cancer	683	9	2
tic-tac-toe	958	9	2
dermatology	366	34	6
Votes	435	16	2
Letter-a	20.000	17	2
Letter-vowel	20.000	17	2

Two algorithms are used to compare with the Ant-MinerCI: C4.5 and GUI-Ant-Miner. The C4.5 [20] is a rule induction algorithm that uses de decision tree representation. The GUI-Ant-Miner [16] is the original version of Ant-Miner with a better interface for user interaction.

Table 2 shows the results with the three algorithms considering the predictive accuracy and the number of rules. The predictive accuracy is related to the test set. The evaluation method used was Cross Validation with ten partitions. In Ant-MinerCI and GUI-Ant-Miner the parameters were set as follows: `no_of_ants` = 1000, `min_cases_per_rule` = 10, `max_uncovered_cases` = 15, `no_of_rules_converg` = 10. The Ant-MinerCI has more two parameters, `max_no_antec` = 4 and `interesting_class` was the minority class for each data set.

Table 2. Predictive accuracy (μ%) e No. of Rules (μ) (*Ant-MinerCI* x GUI-*Ant-Miner* x C4.5) [1]

Data Sets	*Predictive accuracy (%)*			*No. of Rules*		
	Ant-MinerCI (μ%±sd%)	*GUI-Ant-Miner* (μ%±sd%)	C4.5 (μ%)	*Ant-MinerCI* (μ±sd)	*GUI-Ant-Miner* (μ±sd)	C4.5 (μ)
Ljubljana breast cancer	74.73 ± 5.6	74.62 ± 2.54	75.2	3.1 ± 0.54	4.4 ± 0.16	4
Wisconsin breast cancer	93.79 ± 1.13	94.12 ± 1.27	94.99	5.8 ± 0.8	7.2 ± 0.2	19
tic-tac-toe	72.14 ± 4.65	72.22 ± 2.21	84.9	6 ± 0.5	6.5 ± 0.4	95
dermatology	80.67 ± 8.35	86.49 ± 3.8	93.98	5 ± 0.77	6.5 ± 0.31	30
Vote	95.86 ± 9.72	95.45 ± 1.31	96.3	5.1 ± 0.6	5 ± 0.15	6
Letter-a	96.17 ± 4.33	97.82 ± 0.17	98.39	4.2 ± 0.6	7.2 ± 0.13	37
Letter-vowel	81.35 ± 3.12	84.09 ± 1.28	93.02	5 ± 0.5	14 ± 0.1	495

As shown in Table 2, the C4.5 discovered rules with a better predictive accuracy in *tic-tac-toe* and *dermatology*, but the number of discovery rules by the C4.5 in these data sets was higher than the other algorithms. It is important to note that the Ant-MinerCI and GUI-Ant-Miner sacrifice the predictive accuracy to build a model with fewer rules, thus contributing with comprehensibility.

In the remaining data sets *Ljubljana breast cancer, Wisconsin breast cancer, Vote* and *Letter-a, Letter-vowel* the three algorithms achieved a similar predictive accuracy, but the C4.5 has a larger number of rules for *Wisconsin breast cancer, Letter-a* and *Letter-vowel.*

Lastly, analyzing the Ant-MinerCI and GUI-Ant-Miner results, the fact of learning the minority class first did not decrease the predictive accuracy of the algorithm.

The decision tree model of C4.5 has a set of independent rules; consequently, these rules do not have an order. For a new case test, no matter how many rules were tested, there is only one rule that cover the new case.

[1] The numbers rights after the "±" symbol are the standard deviations of the corresponding predictive accuracies rates.

In the ordered rule list discovered of Ant-MinerCI each rule depends of the previous rules. For a rule R_i to be applied in the new case test, the rules of R_0 to R_{i-1} do not cover the new case. This can decrease the comprehensibility of the discovered rules of Ant-MinerCI, but this effect is compensated by the fact that, usually, the size of rule list discovered by Ant-MinerCI is smaller than the decision tree rule discovered by C4.5.

4.2 Ant-MinerCI x Ant-MinerCIP

The goal of Ant-MinerCIP is to discover better rules to rare classes with a better predictive accuracy e comprehensibly. For this study was used the two data sets that are more imbalanced from the previous section, that are Letter-a and Letter-vowel data set, these data sets are characterized as an imbalanced class, which has fewer cases to the interesting class. Table 3 shows the data sets distribution.

Table 3. Data sets distribution

Data set	Classes	Distribition (%)
Letter-a	(a, others)	(4%, 96%)
Letter-vowel	(vowel, others)	(20%, 80%)

The parameters in this experiment were the same as the ones used in Section 4.1, except for the `min_cases_per_rule` which is now 50. Table 4 and Table 5 show the results of predictive accuracy, the coverage (recall), and the number of antecedents of the first rule discovered by the algorithms with the interesting class (minority class).

Table 4. Predictive accuracy(%), Coverage (%) e No. of antecedent of the first rule for class "a" from the *Letter-a* data set (*Ant-MinerCI* x *Ant-Miner*P)

Letter-a	***Class***	***Predictive accuracy (μ%±sd%)***	***Coverage (μ%±sd%)***	***No. of antecedents***
Ant-MinerCI	a	66.07 ± 17.56	**39.58 ± 15.31**	1.5 ± 0.5
Ant-MinerCIP	a	**98.74 ± 0.02**	10.64 ± 2.78	3.2 ± 1

Table 5. Predictive accuracy(%), Coverage (%) e No. of antecedent of the first rule for class "vowel" from the *Letter-vowel* data set (*Ant-MinerCI* x *Ant-Miner*P)

Letter-a	***Class***	***Predictive accuracy (μ%±sd%)***	***Coverage (μ%±sd%)***	***No. of antecedents***
Ant-MinerCI	vowel	28.43 ± 1.61	**31,17 ± 2.2**	1.28 ± 0.45
Ant-MinerCIP	vowel	**100 ± 0**	3.58 ± 0.8	3 ± 1.2

The Ant-MinerCI uses the function Q = sensitivity × specificity to evaluate the rules, thus the algorithm find more general rules, this means that the rule covers a large number of cases, consequently the rule has a lower predictive accuracy because it has a many erroneous cases, which are the FP (false positive) to the rule. However, the Ant-MinerCIP, which uses the precision as a function to evaluate the rules, finds more specific rules, but with a better predictive accuracy. Even if the rules are more specific, they have at least 50 cases covered, because this is the threshold chosen.

Thus, we can say that the Ant-MinerCI is more appropriate to find a model of rules to use as a predictor, and the Ant-MinerCIP is more appropriate to find rules to a particular class, mainly when it is the rare class.

5 Conclusions

In this paper we propose the Ant-MinerCI and the Ant-MinerCIP algorithms; which are extensions to the Ant-Miner algorithm to discovery better rules for the minority classes when the data set has class imbalances. The results showed that, comparing with an original version of Ant-Miner, the Ant-MinerCI obtained a similar predictive accuracy, however the advantage to discovery better rules to the minority class has and improve the simplicity of the rules, because the algorithm limits the number of antecedents. The way that the Ant-MinerCIP calculates the rule quality improves the predictive accuracy to the minority class rules.

References

1. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: Data Mining With an Ant Colony Optimization Algorithm. IEEE Transactions on Evolutionary Computation 6, 321–332 (2000)
2. Japkowicz, N.: Proceedings of the AAAI 2000 Workshop on Learning from Imbalanced Data Sets. AAAI Tech Report WS-00-05. AAAI (2000)
3. Chawla, N.V., Japkowicz, N., Kotcz, A.: Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data Sets (2003)
4. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: Special Issue on Learning from Imbalanced Data Sets. ACM SIGKDD Explorations 6, 1–6 (2004)
5. Weiss, G.: Mining with Rarity: A Unifying Framework. ACM SIGKDD Explorations 6, 7–19 (2004)
6. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. ACM SIGKDD Explorations 6, 20–29 (2004)
7. Japkoawicz, N., Jo, T.: Class Imbalances versus Small Disjuncts. ACM SIGKDD Explorations 6, 40–49 (2004)
8. Liu, X.-Y., Wu, J., Zhou, Z.: Exploratory Under-Sampling for Class-Imbalance Learning. IEEE Transactions on Systems, Man and Cybernetics, Part B, 1–14 (2008)
9. Carvalho, D.R.: Árvore de Decisão/Algoritmo Genético para tratar o Problema de Pequenos Disjuntos em Classificação de Dados. Tese apresentada à COPPE/UFRJ (2005)
10. Dorigo, M., Colorni, A., Maniezzo, V.: The Ant System: Optimization by a colony of cooperating agentes. IEEE Transactions on Systems, Man and Cybernetics 26, 29–41 (1996)

11. Dorigo, M., Di Caro, G.: The Ant Colony Optimization meta-heuristic. In: New Ideas in Optimization, pp. 11–32 (1999)
12. Bo, L., Abbass, H.A., Mckay, B.: Density based Heuristic for Rule Discovery with Ant-Miner. In: The Sixth Australia-Japan Joint Workshop on Intelligent and Evolutionary System, pp. 180–184 (2002)
13. Bo, L., Abbass, H.A., Mckay, B.: Classification Rule Discovery with Ant Colony Optimization. In: Proc. IEEE/WIC International Conference on Intelligence Agent Technology, pp. 83–88 (2003)
14. Martens, D., Backer, M.D., Haesen, R., Vanthienen, J., Snoeck, M., Baesens, B.: Classification with ant colony optimization. IEEE Transactions on Evolutionary Computation 11, 651–665 (2007)
15. Otero, F.E.B., Freitas, A.A., Johnson, C.G.: *c*Ant-Miner: An Ant Colony Classification Algorithm to Cope with Continuous Attributes. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217, pp. 48–59. Springer, Heidelberg (2008)
16. Meyer, F., Parpinelli, R.S.: Gui Ant-Miner: uma versão atualizada do minerador de dados baseado em colônias de formigas. In: I Congresso Sul Catarinense de Computação: UNESC - Criciúma (2005)
17. Salama, K.M., Abdelbar, A.M.: Extensions to the Ant-Miner Classification Rule Discovery Algorithm. In: Dorigo, M., Birattari, M., Di Caro, G.A., Doursat, R., Engelbrecht, A.P., Floreano, D., Gambardella, L.M., Groß, R., Şahin, E., Sayama, H., Stützle, T. (eds.) ANTS 2010. LNCS, vol. 6234, pp. 167–178. Springer, Heidelberg (2010)
18. Machine Learning Group at University of Waikato, `http://www.cs.waikato.ac.nz/~ml/weka`
19. Machine Learning Repository from University California Irvine, `http://archive.ics.uci.edu/ml/`
20. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)

Multivoxel Pattern Analysis Using Information-Preserving EMD

Zareen Mehboob[1,3], Hujun Yin[1,*], Sophie M. Wuerger[4], and Laura M. Parkes[2]

[1] School of Electrical and Electronic Engineering, The University of Manchester, UK
[2] Imaging Sciences, School of Medicine, The University of Manchester, UK
[3.] Department of Computer Science, COMSATS Institute of Information Technology, PK
[4.] School of Psychology, University of Liverpool, UK

Abstract. This paper presents a quantitative analysis on fMRI data using the information-preserving mode decomposition. Multivoxel patterns in fMRI responses in a cognitive experiment were analyzed for spatial selectivity to color perceptions of neurons in the Lateral Geniculate Nucleus (LGN) and the primary visual cortex (V1). The performance of the new method is tested and evaluated in a case study and the results are compared with the previous findings on the same dataset. While conforming to the previous study, the new results have shown improved classification of patterns for unique hues in V1.

1 Introduction

The aim of this work is to develop quantitative methods for analyzing and decoding spatial responses obtained from fMRI (functional magnetic resonance imaging) experiments. It explores the application and performance of our recently proposed method, termed as the information-preserving empirical mode decomposition (IP-EMD) [1] for the analysis of fMRI responses recorded in a cognitive experiment.

The EMD is an adaptive and data-driven method that decomposes a nonlinear time-series into a number of intrinsic oscillatory modes [2]. The IP-EMD uses information theory to retrieve only the informative (stimulus-coding) modes [1]. Such methods offer advantages over direct application of Fourier and wavelet analysis [1-3], as decomposed mode functions usually have narrow bandwidth. EMD has been successfully employed for decoding local field potentials [1] and synchronization between EEG signals [3]. The current work employs the IP-EMD on spatial data obtained from multiple voxels in an fMRI experiment and utilizes informative modes extracted from the multivoxel subtraction signals for further analysis and decoding of spatial information.

The paper is divided into five sections. Section 2 describes the concept of the IP-EMD and the process of extraction of informative modes from multivoxel data. Section 3 presents the case study and description of dataset used for analysis. The experimental details and discussion of results are presented in Section 4, followed by conclusions in Section 5.

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 19–26, 2012.

2 Information-Preserving Empirical Mode Decomposition (IP-EMD)

2.1 General Framework

The general framework for the IP-EMD is depicted in Fig. 1. EMD and information theory are applied on neural responses obtained as continuous time-series in different experiments and then the modes with the maximal information content with respect to original data are extracted and used for further analysis by using different suitable features obtained from these (informative) modes instead of raw (original) data.

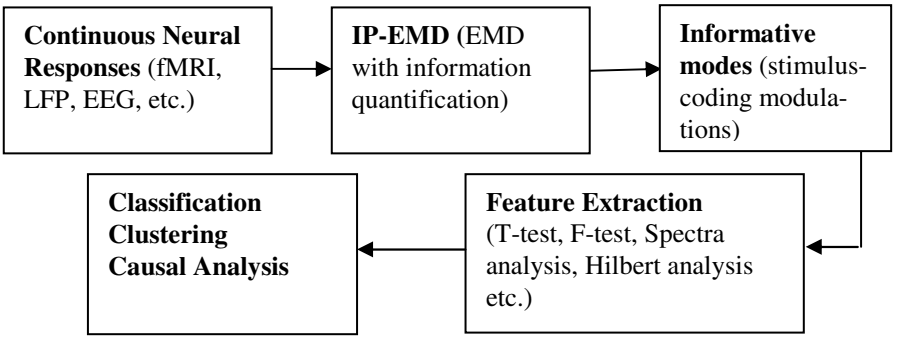


Fig. 1. General framework of the IP-EMD method

2.2 IP-EMD Procedure on fMRI Data

The multivoxel subtraction data is used, which is obtained by subtracting the average minimum image from the average maximum image to give a single subtraction image per color presentation [5,6]. Then the voxel values of each subtraction image are rearranged as a 1D series. The IP-EMD [1] is applied. The details of the procedure are as follows.

1) Apply EMD to each multivoxel subtraction series (MVS). This gives N intrinsic mode functions (IMFs) or simply *modes*. N is the number of modes obtained from a single trial, usually 6-8 for fMRI.
2) Calculate the power spectrum density of MVS and all *modes* using multi-taper method. The power spectrum is the magnitude of Fourier transform of the series on pseudo-frequencies.
3) Calculate Shannon's mutual information (MI) [4] for the series against stimuli, S, using Eq. (1), store it as I_D.

$$I_D(S;MVS) = \sum_{s=1}^{S} P(s) \sum_{MVS_f} P(MVS_f \mid s) \log_2 \frac{P(MVS_f \mid s)}{P(s)} \tag{1}$$

4) Calculate MI for all *modes* using Eq. (2). This gives $I_M(n)$, n=1, 2, ...N.

$$I_M(S;\text{mode}) = \sum_{s=1}^{S} P(s) \sum_{\text{mode}_f} P(\text{mode}_f \mid s) \log_2 \frac{P(\text{mode}_f \mid s)}{P(s)} \quad (2)$$

where in Eqs. (1) and (2), MVS_f and mode_f is magnitude of power spectrum of MVS and mode at pseudo-frequency *f*.

5) For each $\mathcal{I}_M (n = 1, ..., N)$, compare its percentage correlation with $\mathcal{I}_D$.
   Calculate the correlations of all pairs (*N!* in total). Choose the best informative modes (*BIMs*), i.e. a combination of informative modes that have their total information correlation closest to the data and store them as the set of *BIM*.
6) Retain the modes in *BIM* for further analysis.

The next section describes the case study where the proposed scheme is applied for the analysis of an fMRI dataset from a color perception experiment.

3 Case Study

3.1 Experiment and Dataset

Color processing begins in retina where cone cells are tuned to short, medium, or long wavelengths of light. The output of these cone cells is compared by ganglion cells and by neurons in the Lateral Geniculate Nucleus (LGN), which is then projected to neurons in the primary visual cortex (V1) [5-8]. The aim of these experiments was to study the transformation that occurs between the LGN and V1 and the spatial organization of color-tuned neurons in V1 [5].

Five healthy subjects with normal or corrected-to-normal vision participated in this study [5]. Two different sets of color stimuli were used to determine the spatial clustering of V1 and LGN neurons with similar color preferences. fMRI activation patterns were recorded for 4 perceptual hues and 3 cardinal colors [5]. Each subject was scanned twice against the two color modulations (hues and card).

In each experiment, stimuli were presented for 12 seconds followed by 12 seconds of an isoluminant gray screen. The color stimuli had similar spatial and temporal parameters as the stimuli used by [6] and consisted of flickering radial sinusoidal gratings (0.8 cycles/deg; 1.5 Hz; 20 degrees of visual angle diameter), reversing contrast between the endpoints of the cardinal directions in one experiment and the perceptual hues and isoluminant gray in other experiment [5].

The hue map is defined by four perceptual colors: red (PercRed), green (PercGreen), yellow (PercYellow) and blue (PercBlue); this gives a set of four classes (Fig. 2-top row). The cardinal colors are defined by the modulations between red and green (CardRG), lime and violet (CardLV); and black and white (CardBW). This gives three classes for the second set of experiments, shown in Fig. 2 (bottom row).

The mean voxel numbers in V1 across all 5 subjects were 1852 with a standard deviation of 100. In LGN, mean voxel numbers across all 5 subjects were 198 with a standard deviation of 125.

The objective in this study was to analyze the multivoxel patterns (spatial responses) obtained from all voxels in the experiments as in the previous study [5], but

using the IP-EMD. We examined the information content in the spatial patterns and tested whether the data was specific to a particular class. Informative modes were used for classification purpose and classification performances were compared with the results obtained from the original study. This way we quantified the performance of IP-EMD based spatial feature extractions.

IP-EMD analysis and the results determine the degree of spatial clustering of neurons with similar color preferences in human V1 and LGN based on the current knowledge and hypotheses given in following section.

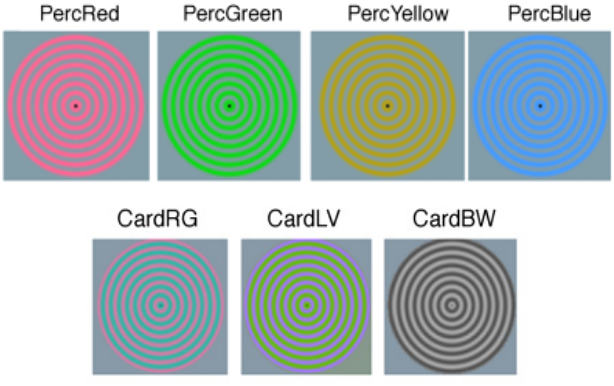


Fig. 2. Top color stimuli show the profiles of four perceptual hue modulations. The color varies over time between gray and the peak contrast of a particular perceptual hue. Color stimuli shown at the bottom are the profiles of three cardinal color modulations. The mean contrast is 0. The figure is adapted from [5].

3.2 Hypothesis

i. There is evidence that in V1, unique activation patterns for perceptual hues are generated which provide evidence for a spatially clustered hue map [5,7,9,10]. It was expected that IP-EMD analysis should reveal the same results and produce the same or similar classification results.

ii. In contrast, previous work has shown that the cardinal color modulations produce highly reproducible patterns of activity in V1 [8], but these were found not to be spatially clustered and unique to each color. This suggests that V1 neurons with tuning characteristics are not spatially clustered [5,9]. This should be supported by the results obtained from IP-EMD analysis and the classification performance in this case should be lower as compared to that for the V1 hues map data.

iii. In contrast, LGN neurons are not expected to produce a unique activation pattern across the voxels in case of hues maps [5]. However the pattern should be unique for cardinal colors due to the known layered structure of the LGN, with each layer responding to a particular cardinal direction. This should also be revealed by the IP-EMD analysis.

The details of the analysis are given in the next section.

4 Experiments and Analysis

4.1 IP-EMD Analysis of fMRI Activation Patterns

Informative modes were extracted from the subtraction signals using the IP-EMD. For step 2 (Section 2.2), the window for the multitaper method was set to 256.

The analysis was conducted on the data from individual subjects in both experiments because it was observed that the mutual information profiles for each subject against same stimuli generated uncorrelated patterns. This is to be expected because of the differences in micro-structural anatomy between people. For example, the information from two subjects namely *Tc* and A*b* against hue mapping is shown in Fig. 3. It reveals that information content varies and peaks in different frequency ranges. For comparison purpose, the information was calculated over the same number of voxels (1731 points in this case). The IP-EMP retains most of the information, except for the range 20-40 (pseudo-frequency Hz).

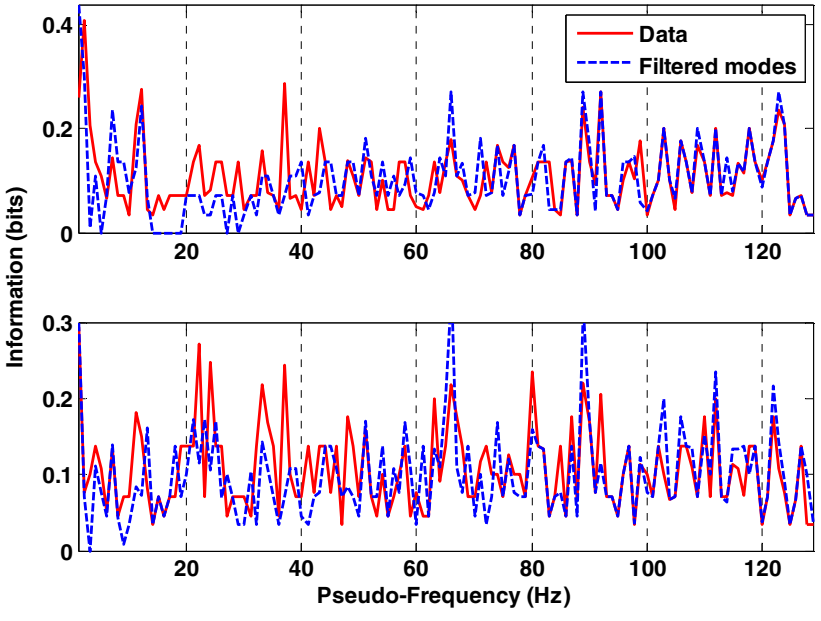


Fig. 3. Information obtained from multivoxel patterns of two subjects against hue mapping. For subject *Tc* (top), Information peaks are visible between 1-40 Hz in the range of 0.2-0.4 bits. For subject *A*b (bottom), information peaks are in lower range (0.2-0.3) bits.

4.2 Multivoxels Pattern Classification

Classification was performed between all color pairs within each set, i.e., three pairings for the cardinal colors and six pairings for the perceptual hues and analysis was performed for all five subjects.

Two-way classification was used to produce the largest number of tests. A SVM (support vector machine) based classifier [11,12] was trained with some samples, and tested with a completely different samples later. Given n data samples (trials) in each experiment, 2/3n of the trials were used for training at one time and the remaining 1/3n were used for testing, so that the final error assessment is objective. In case of n=15, 10 trials were used for training and remaining 5 were used for testing. This means that the SVM was trained 3003 times for a unique combination of 10 (out of 15) trials and each testing prediction was recorded. The SVM parameters for both classification tasks were identical and the final classification performance is the average result from all runs.

4.3 Results and Discussions

For all five subjects, the mean of classification results obtained from six combinations of hue maps and three combinations of cardinal color maps are shown in Table 1.

Table 1. SVM based classification results obtained from five subjects

Classification Accuracy (%)							
Hue Colors Map				Cardinal Colors Map			
Raw data	Informative modes	Raw data	Informative modes	Raw data	Informative modes	Raw data	Informative modes
Tc V1 hues *Icorr*=0.577±0.06		***Tc LGN hues*** *Icorr*=0.46±0.14		***Tc LGN card*** *Icorr*=0.39±0.18		***Tc V1 card*** *Icorr*=0.37±0.11	
64.76 ±14.94	69.32 ±14.71	50.02 ±5.65	52.75 ±3.67	46.98 ±2.98	52.62 ±3.83	48.20 ±8.83	48.78 ±9.78
Ab V1 hues *Icorr*=0.40±0.07		***Ab LGN hues*** *Icorr*=0.51±0.12		***Ab LGN card*** *Icorr*=0.38±0.12		***Ab V1 card*** *Icorr*=0.48± 0.03	
44.24 ±6.24	50.09 ±7.30	50.31 ±3.02	50.18 ±6.71	45.54 ± 3.49	48.78 ±3.24	59.69 ±6.36	60.93 ±10.37
Lp V1 hues *Icorr*=0.44±0.09		***Lp LGN hues*** *Icorr*=0.44±0.04		***Lp LGN card*** *Icorr*=0.49±0.16		***Lp V1 card*** *Icorr*=0.41± 0.02	
57.21 ±8.03	60.51 ±7.40	47.91 ±7.76	50.98 ±7.76	49.42 ±2.10	50.55 ±1.67	57.64 ±5.76	53.47 ±3.99
Lm V1 hues *Icorr*=0.51±0.05		***Lm LGN hues*** *Icorr*=0.45±0.12		***Lm LGN card*** *Icorr*=0.51±0.11		***Lm V1 card*** *Icorr*=0.49± 0.09	
61.45 ±10.85	62.67 ±10.05	51.95 ±4.48	52.26 ±4.70	50.21 ±3.72	51.34 ±4.78	57.47 ±8.18	53.13 ±8.46
Cf V1 hues *Icorr*=0.56±0.14		***Cf LGN hues*** *Icorr*=0.41±0.15		***Cf LGN card*** *Icorr*=0.36±0.05		***Cf V1 card*** *Icorr*=0.34± 0.01	
64.08 ±12.23	69.22 ±14.21	52.45 ±8.56	51.77 ±9.16	43.54 ±1.32	44.78 ±0.25	47.20 ±5.37	35.53 ±0.67

The results obtained using the original data and the extracted informative modes are presented. The subject name and data labels are given, together the total information correlation (*Icorr*) of the informative modes with original (raw) data.

From columns 1 and 2, it can be seen that the *Icorr* values are higher for V1 hues (in all subjects except Ab) as compared to LGN hues data (columns 3 and 4). This shows that better information preservation level was obtained for V1 hues. It is also verified by the improvement in classification accuracy for V1 hues data and a reduction in classification accuracy for LGN hues data (columns 5 and 6). Therefore the hypotheses (i) and (ii) are confirmed.

From columns 5 and 6, it can be seen that *Icorr* values are higher for LGN card data (in all subjects except Ab) as compared to V1 card (columns 7 and 8). This shows better information preservation level for LGN card data. In terms of LGN card classification accuracy, we did not find much evidence in support of hypothesis (iii) as the results were hardly above chance in all subjects. However, V1 card classification results are supporting the hypothesis (iii) with no significant improvement in the classification accuracy.

5 Conclusion

The analysis shows that mutual information from multivoxels spatial patterns is extracted well when neurons are spatially clustered and there is a distinct response pattern across the region for hue colors in V1; this was also shown by a slight improvement in classifier performance (1-6%). In contrast, IP-EMD analysis did not show a higher level of preservation in cases when the neuronal responses were not spatially clustered in a brain region. This was also reflected by the classification performance which remained unchanged or reduced by 1-3% for indistinguishable patterns against hue colors in LGN and cardinal colors in V1 region.

The classifier in this study used a single feature extracted from the informative modes and showed a slight improvement in the classification performance for distinctively generated patterns. It is likely that extending the feature set may further improve the classification performance in case of hue colors in V1 and cardinal colors in LGN.

References

1. Mehboob, Z., Yin, H.: Information quantification of empirical mode decomposition and applications to field potentials. International Journal of Neural Systems 21(1), 49–63 (2011)
2. 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 nonstationary time series analysis. Proc. Roy. Soc. Lond. A 454, 903–1005 (1998)
3. Sweeney-Reed, C.M., Nasuto, S.J.: A novel approach to the detection of synchronisation in EEG based on empirical mode decomposition. Journal of Computational Neuroscience 23, 79–111 (2007)

4. Abbot, L.F., Dayan, P.: Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. MIT Press (2001)
5. Parkes, L.M., Marsman, J.C., Oxley, D.C., Goulermas, J.Y., Wuerger, S.M.: Multivoxel fMRI analysis of color tuning in human primary visual cortex. Journal of Vision 9(1), 1–13 (2009)
6. Haxby, J.V., Gobbini, M.I., Furey, M.L., Ishai, A., Aschouten, J.L., Pietrini, P.: Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science 293(5539), 2425–2430 (2001)
7. De Valois, R.L., De Valois, K.K., Mahon, L.E.: Contribution of S opponent cells to color appearance. Proceedings of the National Academy of Sciences of the United States of America 97, 512–517 (2000)
8. Wuerger, S.M., Atkinson, P., Cropper, S.: The cone inputs to the unique-hue mechanisms. Vision Research 45(25-26), 3210–3223 (2005)
9. Horwitz, G.D., Chichilnisky, E.J., Albright, T.D.: Cone inputs to simple and complex cells in V1 of awake macaque. Journal of Neurophysiology 97, 3070–3081 (2007)
10. Brouwer, G.J., Heeger, D.J.: Decoding and reconstructing color from responses in human visual cortex. Journal of Neuroscience 29(44), 13992–14003 (2009)
11. Cristianini, N., Shawe-Taylor, J.: An Introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, New York (2000)
12. De Martino, F., Gentile, F., Esposito, F., Balsi, M., Di Salle, F., Goebel, R.: Classification of fMRI independent components using IC-fingerprints and support vector machine classifiers. Neuroimage 34(1), 177–194 (2007)

Population Resizing Using Nonlinear Dynamics in an Ecology-Based Approach

Rafael Stubs Parpinelli[1,2,⋆] and Heitor Silvério Lopes[2]

[1] Applied Cognitive Computing Group
Santa Catarina State University
Joinville, Brazil
[2] Bioinformatics Laboratory
Federal Technological University of Paraná
Curitiba, Brazil
parpinelli@joinville.udesc.br,
hslopes@utfpr.edu.br

Abstract. It is well known that, in nature, populations are dynamic in space and time. This means that the size of populations oscillate across their habitats over time. This work uses the concepts of habitats, ecological relationships, ecological successions and population dynamics to build a cooperative search algorithm, named ECO. This work aims to explore the population sizing not as a parameter but as a dynamic process. The Artificial Bee Colony (ABC) was used in the experiments where benchmark mathematical functions were optimized. Results were compared with ABC running alone, with and without the use of population dynamics. The ECO algorithm with population dynamics performed better than the other approaches, possibly thanks to the ecological interactions (intra and inter-habitats) that enabled the co-evolution of populations and to a more natural survival selection mechanism by the use of population dynamics.

Keywords: optimization, cooperative search, co-evolution, habitats, logistic chaos model, ecology, population dynamics.

1 Introduction

Nature has always been an endless source of inspiration for computational models and paradigms, in particular for the computer scientists of the area known as Natural Computing [1]. Two outstanding families of bio-inspired algorithms are evolutionary computation (EC) and swarm intelligence (SI) that currently offer a wide range of strategies for optimization [2] [3]. The concept of optimization can be abstracted from several natural processes such as the evolution of the species, the behavior of social groups, the dynamics of the immune system, the strategies

⋆ Authors would like to thank the Brazilian National Research Council (CNPq) for the research grant to H.S. Lopes; as well as to UDESC (Santa Catarina State University) and FUMDES program for the doctoral scholarship to R.S. Parpinelli.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 27–34, 2012.

of searching for food and in the ecological relationships of different populations. Most of these cases were the source of inspiration to the development of new algorithms for optimization.

It is worth mentioning that most bio-inspired algorithms only focus on and take inspiration from specific aspects of the natural phenomena. However, in nature, biological systems are interlinked to each other, e.g. biological ecosystems [4] [5]. Also, in nature, populations are always dynamic in such a way that the size of populations oscillate across their habitats over time. However, in most Evolutionary Computation applications, the population size is constant and does not change during the search [6]. Current practice of manual setting of population size in evolutionary computation is experience-based, but not robust. Hence, this work aims to explore the population sizing not as a parameter but as a dynamic process that changes the population size deterministically over time.

In [7] the authors first introduce the potentiality of some ecological concepts (e.g., habitats, ecological relationships and ecological successions) presenting a simplified ecological-inspired algorithm. In this work we explore a more biologically plausible survival selection mechanism by the use of population dynamics where the logistic chaos model is applied to control the size of populations [8] [5]. The aim is to compare the results obtained by the implementation of the algorithm with the use of ecological concepts, without the use of ecological concepts (application of stand alone algorithms), and with and without the use of population dynamics.

2 The Proposed Ecological-Inspired Approach

The ecological-inspired algorithm, named ECO, represents a new perspective to develop cooperative evolutionary algorithms. The ECO is composed of populations of individuals (candidate solutions for a problem being solved) and each population evolves according to an optimization strategy. Therefore, individuals of each population are modified according to the mechanisms of intensification and diversification, and the initial parameters, specific to each optimization strategy. The ECO system can be modeled in two ways: homogeneous or heterogeneous. A homogeneous model implies that all populations evolve in accordance to the same optimization strategy, configured with the same parameters. Any change in the strategies or parameters in at least one population characterises a heterogeneous model.

The ecological inspiration stems from the use of some ecological concepts, such as: habitats, ecological relationships and ecological successions [4] [5]. Once dispersed in the search space, populations of individuals established in the same region constitute an ecological habitat. For instance, in a multimodal hyper-surface, each peak can become a promising habitat for some populations. A hyper-surface may have several habitats. As well as in nature, populations can move around through all the environment. However, each population may belong only to one habitat at a given moment of time t. Therefore, by definition, the intersection between all habitats at moment t is the empty set.

With the definition of habitats, two categories of ecological relationships can be defined. Intra-habitats relationships that occur between populations inside each habitat, and inter-habitats relationships that occur between habitats [4] [5].

In ECO, the intra-habitat relationship is the mating between individuals. Populations belonging to the same habitat can establish a reproductive link between their individuals, meshing the populations and favoring the co-evolution of the involved populations. Populations belonging to different habitats are called reproductively isolated.

The inter-habitats relationship are the great migrations. Individuals belonging to a given habitat can migrate to other habitats aiming at identifying promising areas for survival and mating.

In addition to the mechanisms of intensification and diversification specific to each optimization strategy, when considering the ecological context of the proposed algorithm, the intra-habitats relationships are responsible for intensifying the search and the inter-habitats relationships are responsible for diversifying the search.

Inside the ecological metaphor, the ecological successions represent the transformational process of the system. In this process, populational groups are formed (habitats), relations between populations are established and the system stabilizes by means of the self-organization of its components.

Algorithm 1 shows the pseudo-code of the proposed approach. For a detailed description refer to [3]. The metric chosen to define the region of reference is the centroid and represents the point in the space where there is a highest concentration of individuals of population i.

2.1 Population Dynamics

In this paper we applied the one-parameter logistic chaos map to drive the population dynamics between ecological successions. The logistic map is often

Algorithm 1. Pseudo-code for ECO

1: Consider $i = 1, \ldots, NQ$, $j = 1, \ldots, NH$ and $t = 0$;
2: Initialize each population Q_i^t with n_i random candidate solutions;
3: **while** stop criteria not satisfied **do** {Ecological succession cycles}
4: Perform evolutive period for each population Q_i^t;
5: Apply metric C_i to identify the region of reference for each population Q_i^t;
6: Using the C_i values, define the NH habitats;
7: For each habitat H_j^t define the communication topology CT_j^t between populations Q_{ij}^t;
8: For each topology CT_j^t, perform interactions between populations Q_{ij}^t;
9: Define communication topology TH^t between H_j^t habitats;
10: For TH^t topology, perform interactions between H_j^t habitats;
11: Check populational scenario and compute population size;
12: Increase t;
13: **end while**

cited as an example of how chaotic behaviour can arise from very simple non-linear dynamical equations. Also, it can be used as a discrete-time demographic model [8]. Equation 1 presents the logistic chaos map used, where $0 < a < 4$.

$$Pop_i^{t+1} = aPop_i^t(1 - Pop_i^t) \tag{1}$$

Three populational scenarios can occur between ecological successions (line 11 in Algorithm 1). The first is when there are no changes in the size of populations from moment t to $t+1$. In this case the evolution proceeds as usual. The second is when there is increment in the size of populations from t to $t+1$. In this case, new solutions are randomly generated using the current centroid as reference. The third scenario is when there is decrement in the size of populations from t to $t+1$. In this case, the worst solutions are discarded. Hence, the second scenario favors exploration and the third scenario favors exploitation. In addition to the exploration and exploitation routines provided by the evolution of populations and by the ecological interactions (inter and intra-habitats), the use of population dynamics creates a new biologically plausible mechanism to diversify the search.

3 Experiments and Results

Experiments were conducted using four benchmark functions extensively used in the literature for testing optimization methods [9]. Each function to be minimized was tested with 10 and 200 dimensions. The first function ($f_1(\boldsymbol{x})$) is known as generalized F6 Schaffer function. The second function ($f_2(\boldsymbol{x})$) is the Rastrigin function. The third function ($f_3(\boldsymbol{x})$) is the Griewank function. Finally, the fourth function ($f_4(\boldsymbol{x})$) is the Rosenbrock function. Table 1 summarizes the informations about the functions used.

The parameters of the ECO algorithm are: number of populations (*N-POP*) that will be co-evolved, the initial population size (*POP-SIZE*), number of cycles for ecological successions (*ECO-STEP*), the size of the evolutive period (*EVO-STEP*) that represents number of function evaluations in each *ECO-STEP*, the tournament size (*T-SIZE*) used to choose solutions to perform intra and inter-habitat communications and the proximity threshold ρ used to define the habitats. Studies about the adjustment of parameters have not been carried out yet. Hence, all the parameters of the algorithm were defined empirically [3].

Table 1. Benchmark Functions

Function	Definition	Domain	Global Optimun
$f_1(\boldsymbol{x})$	$\sum_{i=1}^{n-1}\left(0.5+\frac{\sin^2\left(\sqrt{x_{i+1}^2+x_i^2}\right)-0.5}{\left(0.001\left(x_{i+1}^2+x_i^2\right)+1\right)^2}\right)$	$-100 \leq x_i \leq 100$	$f_1(\mathbf{0}) = 0$
$f_2(\boldsymbol{x})$	$\sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$	$-5.12 \leq x_i \leq 5.12$	$f_2(\mathbf{0}) = 0$
$f_3(\boldsymbol{x})$	$\frac{1}{4000}\left(\sum_{i=1}^{n} x_i^2\right) - \left(\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)\right) + 1$	$-600 \leq x_i \leq 600$	$f_3(\mathbf{0}) = 0$
$f_4(\boldsymbol{x})$	$\sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$-30 \leq x_i \leq 30$	$f_4(\mathbf{1}) = 0$

In all experiments, the Artificial Bee Colony Optimization (ABC) algorithm [10] was used in a homogeneous model, i.e. all populations use this algorithm with the same control parameters.

For all experiments the initial population size was set to $POP\text{-}SIZE = 10$. The logistic chaos map (Equation 1) was applied to adjust the population size dynamically. The logistic chaos map parameter was set to $a = 3.57$ and it is called 'route to chaos' [8]. This choice was done based on the work of Ma [11] where experiments were performed with different values for the parameter a. Figure 1 shows the resizing projection for 100 ecological successions.

For the number of dimensions (D) equal to 10, the parameters used were $N\text{-}POP = 100$, $ECO\text{-}STEP = 100$, $EVO\text{-}STEP = 100$, $T\text{-}SIZE = 5$ e $\rho = 0,5$. With this configuration, the total number of function evaluations was 10,000 for each population. For $D = 200$, some parameters were redefined: $N\text{-}POP = 200$, $ECO\text{-}STEP = 500$, $EVO\text{-}STEP = 200$. With this adjustment of parameters, for 200 dimensions, the total number of function evaluations was 100,000 evaluations for each population.

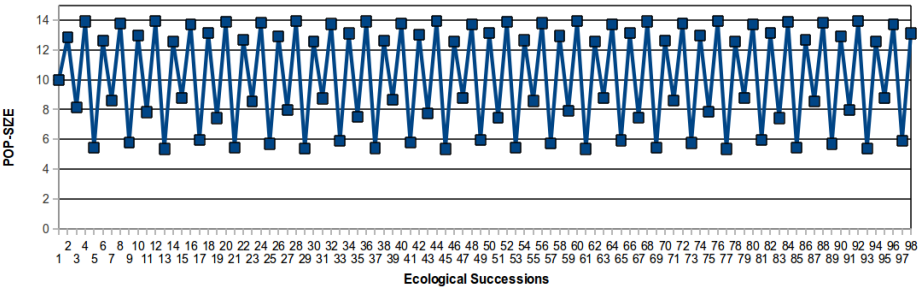


Fig. 1. Population dynamics according to the logistic chaos model ($a = 3.57$)

The ecological-inspired framework (ECO) was tested using four configurations. The first configuration implements the Algorithm 1 as described in Section 2, with the definitions of habitats, topologies and ecological relations. The second configuration complements the first one by adding population dynamics (Section 2.1). The third configuration disables the ability to create habitats and, consequently, topologies and interactions are not defined. This third configuration simulates the evolution completely isolated populations, and they evolve without exchanging information. The fourth configuration complements the third one by adding population dynamics. For each configuration, the algorithm was run 30 times.

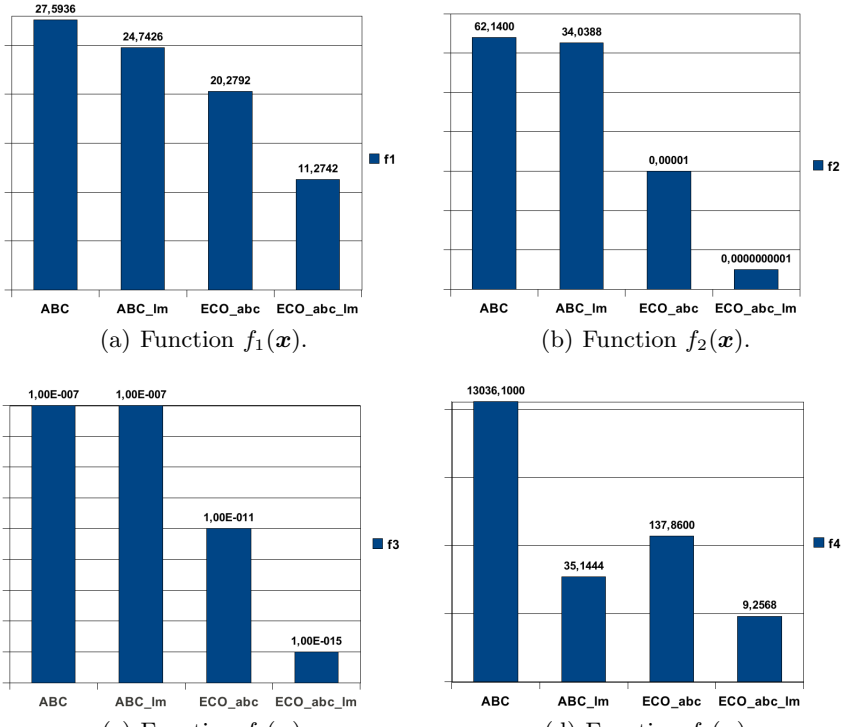
Table 2 shows the averaged results obtained for the benchmark functions. For both dimensions, $D = 10$ and $D = 200$, the results obtained by each configuration of the algorithms are presented (columns 2 to 4). Column 2 shows the results obtained by the ABC algorithm running alone, without co-evolution.

Table 2. Obtained results for the benchmarck functions

$f_1(\boldsymbol{x})$	$D = 10$			
Model	ABC	ABC_{LM}	ECO_{ABC}	ECO_{ABC-LM}
Global Best	4.6569 ± 0.8	1.4559 ± 0.2	1.1344 ± 0.2	0.5193 ± 0.0
	$D = 200$			
Model	ABC	ABC_{LM}	ECO_{ABC}	ECO_{ABC-LM}
Global Best	27.5936 ± 0.73	24.7426 ± 0.5	20.2792 ± 0.40	11.2742 ± 0.4
$f_2(\boldsymbol{x})$	$D = 10$			
Model	ABC	ABC_{LM}	ECO_{ABC}	ECO_{ABC-LM}
Global Best	10^{-11} ± 0.0	10^{-12} ± 0.0	0.0000 ± 0.0	0.0000 ± 0.0
	$D = 200$			
Model	ABC	ABC_{LM}	ECO_{ABC}	ECO_{ABC-LM}
Global Best	62.1453 ± 9.6	34.0388 ± 4.3	10^{-05} ± 0.0	10^{-10} ± 0.0
$f_3(\boldsymbol{x})$	$D = 10$			
Model	ABC	ABC_{LM}	ECO_{ABC}	ECO_{ABC-LM}
Global Best	10^{-06} ± 0.0	10^{-17} ± 0.0	10^{-13} ± 0.0	10^{-18} ± 0.0
	$D = 200$			
Model	ABC	ABC_{LM}	ECO_{ABC}	ECO_{ABC-LM}
Global Best	10^{-7} ± 0.0	10^{-7} ± 0.0	10^{-11} ± 0.0	10^{-15} ± 0.0
$f_4(\boldsymbol{x})$	$D = 10$			
Model	ABC	ABC_{LM}	ECO_{ABC}	ECO_{ABC-LM}
Global Best	0.0098 ± 0.0	0.0098 ± 0.0	0.0086 ± 0.0	0.0064 ± 0.0
	$D = 200$			
Model	ABC	ABC_{LM}	ECO_{ABC}	ECO_{ABC-LM}
Global Best	13036.1 ± 4193.4	35.1444 ± 11.6	137.86 ± 42.0	9.2568 ± 4.1

Column 3 shows the results obtained by the ABC algorithm running with the logistic chaos model for modelling the population dynamics (ABC_{LM}). Column 4 shows the results obtained by the ABC algorithm using the ecological-inspired approach (ECO_{ABC}). Finally, column 5 shows the results obtained by the ABC algorithm using the ecological-inspired approach running with the logistic chaos model for the population dynamics (ECO_{ABC-LM}). For each dimension, the third line (*Global Best*) shows the average and standard deviation of the best result obtained by all populations in all runs.

Analyzing the ABC and ABC_{LM} configurations we can observe that the use of population dynamics improved the results in most cases and remained the same in two. Analyzing the ABC and ECO_{ABC} we can observe that the ecological-inspired approach obtained much better results than the algorithm executed without the concepts of habitats for all functions. Analyzing the results for the ecological-inspired approach with population dynamics, ECO_{ABC-LM}, we can observe that the results were significantly better than the ecological-inspired approach without population dynamics (ECO_{ABC}). This gain is mainly due to the ecological interactions (intra and inter-habitats) that enabled the co-evolution of populations and to the use of a more natural survival selection mechanism afforded by population dynamics. Moreover, the ECO_{ABC-LM} was the best approach for all functions. In Figure 2 we can visually verify the results for $D = 200$, where the x-*axis* shows the different approaches and the y-*axis* represents the *Global Best* values of each approach and are shown at the top of each bar.



(a) Function $f_1(\boldsymbol{x})$. (b) Function $f_2(\boldsymbol{x})$.

(c) Function $f_3(\boldsymbol{x})$. (d) Function $f_4(\boldsymbol{x})$.

Fig. 2. Bar graph of each benchmarck function with $D = 200$

4 Conclusions

This paper presents an ecological-inspired algorithm for optimization with population dynamics. The proposed algorithm uses cooperative search strategies where populations of individuals co-evolve and interact among themselves using some ecological concepts. Each population behaves according to the mechanisms of intensification and diversification, and the control parameters, specific to a given search strategy. The Artificial Bee Colony Optimization algorithm was used in all populations. In this work, a population dynamics model was used to set up population sizes in the computational ecosystem. The population dynamics model applied was the logistic chaos due to its simplicity and its rich dynamic behaviour as discrete-time demographic model.

The main ecological concepts addressed are the definition of habitats, ecological relationships, ecological successions and population dynamics. These features bring a higher biological plausibility to the proposed algorithm, opposed to most bio-inspired algorithms that take inspiration only from one biological phenomenon. Thus, the proposed methodology opens the possibility for the

insertion of several ecological concepts in the optimization process, bringing more biological plausibility to the system.

In addition to the exploration and exploitation routines provided by the evolution of populations and by the ecological interactions (inter and intra-habitats), the use of population dynamics creates a new biologically plausible mechanism to diversify the search.

The results showed that the use of habitats and ecological relationships influence significantly the co-evolution process of populations, leading to better solutions (when compared to the results not using the ecological concepts). Also, the use of a population dynamics model inside the ECO framework improved considerably the results.

This work is still under development and as future work we intend to analyze the influence of the control parameters (number of ecological successions, evolutive period, number of populations, and ρ threshold for creation of habitats) on the quality of solutions, as well as to add other search strategies in the proposed model. Another future research is to use the ECO approach in an asynchronous model. Also, the application of other population dynamics models such as the Lotka-Volterra predator-prey can be an interesting direction. Currently, in order to bring more biological plausibility to the system, other ecological concepts are being modeled.

References

1. de Castro, L.N.: Fundamentals of natural computing: an overview. Physics of Life Reviews 4(1), 1–36 (2007)
2. Engelbrecht, A.P.: Computational Intelligence: An Introduction, 2nd edn. Wiley, Chichester (2007)
3. Parpinelli, R.S., Lopes, H.S.: New inspirations in swarm intelligence: a survey. International Journal of Bio-Inspired Computation 3(1), 1–16 (2011)
4. Begon, M., Townsend, C.R., Harper, J.L.: Ecology: from individuals to ecosystems, 4th edn. Blackwell Publishing, Oxford (2006)
5. May, R.M.C., McLean, A.R.: Theoretical Ecology: Principles and Applications. Oxford University Press, Oxford (2007)
6. Eiben, A., Smith, J.: Introduction to Evolutionary Computing. Springer, London (2003)
7. Parpinelli, R.S., Lopes, H.S.: An eco-inspired evolutionary algorithm applied to numerical optimization. In: Proceedings of the Third World Congress on Nature and Biologically Inspired Computing, Salamanca, Spain, pp. 473–478 (2011)
8. Kaplan, D., Glass, L.: Understanding Nonlinear Dynamics. Springer, New York (1995)
9. Digalakis, J.G., Margaritis, K.G.: An experimental study of benchmarking functions for evolutionary algorithms. International Journal of Computer Mathematics 79(4), 403–416 (2002)
10. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 214, 108–132 (2009)
11. Ma, Z.S.: Ecological theatre for evolutionary computing play: some insights from population ecology and evolutionary ecology. International Journal of Bio-Inspired Computation 3(1), 31–55 (2011)

Multi-class Contour Preserving Classification

Piyabute Fuangkhon and Thitipong Tanprasert

Distributed and Parallel Computing Research Laboratory
Faculty of Science and Technology, Assumption University, Bangkok, Thailand
piyabutefng@au.edu, thitipong@scitech.au.edu
http://www.scitech.au.edu

Abstract. The original contour preserving classification technique was proposed to improve the robustness and weight fault tolerance of a neural network applied with a two-class linearly separable problem. It was recently found to be improving the level of accuracy of two-class classification. This paper presents an augmentation of the original technique to improve the level of accuracy of multi-class classification by better preservation of the shape or distribution model of a multi-class problem. The test results on six real world multi-class datasets from UCI machine learning repository present that the proposed technique supports multi-class data and can improve the level of accuracy of multi-class classification more effectively.

Keywords: contour preserving classification, data preprocessor, neural network, outpost vector, pattern classification.

1 Introduction

A multi-layer perceptron (MLP) [1–3] is a feed-forward neural network (FFNN) model that maps sets of input data onto a set of output. It comprises multiple layers of nodes in a directed graph. Each layer is fully connected to the next one. Each node is a neuron with a nonlinear activation function except for the input nodes. An MLP uses a supervised learning technique called back-propagation for training the network. It is a modification of the standard linear perceptron (single-layer perceptron), which can distinguish data that is not linearly separable.

When an MLP is used to solve multi-class problem, the placement of classifying hyper-planes significantly affects the level of accuracy of classification. Considering a problem in Fig. 1a, a two-dimension three-class problem comprising three classes of input vectors that is separable by a two-neuron non-linear classifier is presented. When a two-layer feed-forward neural network having two hidden neurons is applied, it possibly learns to classify the three classes of input vectors as shown in Fig. 1b. Line 1 and 2 are the classifying hyper-plane that represents the two hidden neurons. When a two-layer feed-forward neural network having four hidden neurons is applied, a possible solution may be as shown in Fig. 1c. Line 3, 4, 5 and 6 are the classifying hyper-plane that represents the four hidden neurons. Applying a typical variation of back-propagation learning

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 35–42, 2012.

algorithm to learn those input vectors will normally converge to the solution similar to what shown in Fig. 1d. The classifier may classify the data inaccurately especially when input vectors are placed at the boundary between consecutive classes of data because it does not recognize the shape or distribution model of the problem correctly.

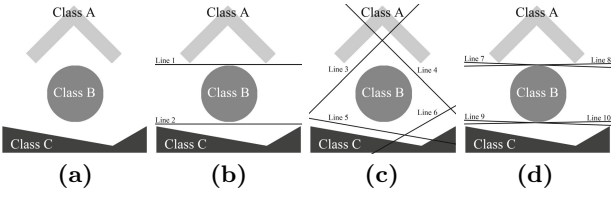


Fig. 1. (a) A two-dimension three-class problem (b) Possible classification with a Two-layer Perceptron with two hidden neurons (c) Possible classification with a Two-layer Perceptron with four hidden neurons (d) Typical result of four-hidden-neuron network trained by a variation of back-propagation algorithm

Improving the level of accuracy of multi-class classification is significant because plenty of real world problems have more than two classes of data. This paper presents a technique that can improve the level of accuracy of multi-class classification. The technique generates additional vectors at the boundary between consecutive classes of data to better preserve the shape or distribution model of the multi-class data. These vectors assist a feed-forward neural network to classify the multi-class data more accurately.

2 Related Works

Researches [4–9] had been conducted to directly and indirectly improve the level of accuracy of classification of two-class data. Tanprasert et al. [4] proposed a contour preserving classification technique, called Outpost Vector Model, to preserve the shape or distribution model of two-class data to improve the robustness and weight fault tolerance of a neural network applied with a linearly separable problem. Its major idea is to force nonlinear classification on a linearly separable problem so as to take advantage of the nonlinear contour of the input vector regions to widen the clearance between the classification hyper-plane and the input vectors. A concept of outpost vectors is introduced for augmenting a set of training vectors with a set of outpost vectors (additional vectors) that is generated from the training vectors and is placed at the boundary between both classes of data. As a result, the learning process is indirectly biased towards distributing classification workload around the set of hidden neurons, thereby forcing the network to perform nonlinear classification. Although outpost vectors have some similarity with support vectors [10], they are synthesized instead of being selected from training vectors. The technique was found in [5, 6] to be improving the level of accuracy of classification of two-class data.

Mongkonsirivatana [7] proposed an improved version of original outpost vector generation algorithm [4] that uses boundary detection technique to reduce the number of original vectors used to generate outpost vectors. The outpost vectors are generated from a set of original vectors that corresponds to the boundary of the shape or distribution model of the problem only. Tanprasert and Kripruksawan [8] and Kaitikunkajorn and Tanprasert [9] proposed techniques that use subtractive clustering [11] to find the center of cluster of original vectors in the same class. The outpost vectors are generated around that cluster center only. These techniques are intended to reduce the number of generated outpost vectors. However, they do not accurately preserve the shape or distribution model of the problem.

3 Multi-class Contour Preserving Classification

This section presents the augmentation of the original contour preserving classification technique [4] to support multi-class data and improve the level of accuracy of multi-class classification.

Multi-class outpost vectors generated from multi-class outpost vector generation algorithm presented in Fig. 2 comprise two types of multi-class outpost vectors.

1. **Fundamental Multi-class Outpost Vector (FMCOV)** is a synthesized vector that is used to declare the boundary of the territory of an input vector of one class, let's say input vector i of class A (denoted by A_i), against an input vector of any other class, let's say input vector j of class X (denoted by X_j), that has smallest Euclidean distance to A_i. An FMCOV of input vector i (denoted by $o(i)$) is placed at the boundary of A_i's territory in the direction of X_j. X_j is designated as a *paired input vector* of A_i (denoted by $\phi(i)$).
2. **Additional Multi-class Outpost Vector (AMCOV)** is a synthesized vector that is used to declare the boundary of a paired input vector of an input vector, let's say a paired input vector of input vector i ($\phi(i)$), against that input vector (A_i). An AMCOV of input vector i (denoted by $o'(i)$) is placed at the boundary of $\phi(i)$'s territory in the direction of A_i. Many paired input vectors may be generated at the same point in the problem space. To reduce the number of duplicated AMCOVs, an AMCOV of an input vector will be generated only when a paired input vector of a paired input vector of that input vector is not that input vector itself ($\phi(\phi(i)) \neq i$).

The multi-class outpost vector generation algorithm in Fig. 2 takes $O(c^2n^2)$ where c is the number of classes of data and n is the number of original vectors. When n is large enough, the number of classes of data (c) will be insignificant.

The concept of Three-class Outpost Vector is illustrated in Fig. 3. There are three classes of data designated as class A, B and C. To find the territory of each input vector, each input vector is modeled to span its territory as a

01 **for** each input vector i
02 find an vector $\phi(i) \in$ **all other classes** and has shortest Euclidean distance to vector i.
03 place a fundamental multi-class outpost vector, $o(i) \in$ the class of vector i, at **almost** half way between vector i and vector $\phi(i)$ on the territory of i in the direction of vector $\phi(i)$.
04 **for** each input vector i
05 **if** $(\phi(\phi(i)) \neq i)$ **then**
06 place an additional multi-class outpost vector, $o'(\phi(i)) \in$ the class of vector $\phi(i)$, at **almost** half way between vector $\phi(i)$ and vector i on the territory of $\phi(i)$ in the direction of vector i.

Fig. 2. Multi-class outpost vector generation algorithm. The first *for-loop* in the algorithm is the FMCOV generator while the second *for-loop* in the algorithm is the AMCOV generator. During the FMCOV generation process, the initial training set, regardless of the number of classes of data, is viewed as a two-class dataset: class A for a being processed input vector and class B for all other input vectors not in class A. During the AMCOV generation process, only two input vectors from two different classes are involved: class A for a being processed input vector and class B for a paired input vector of that input vector. As a result, the multi-class outpost vector generation algorithm can be applied with any number of classes of data.

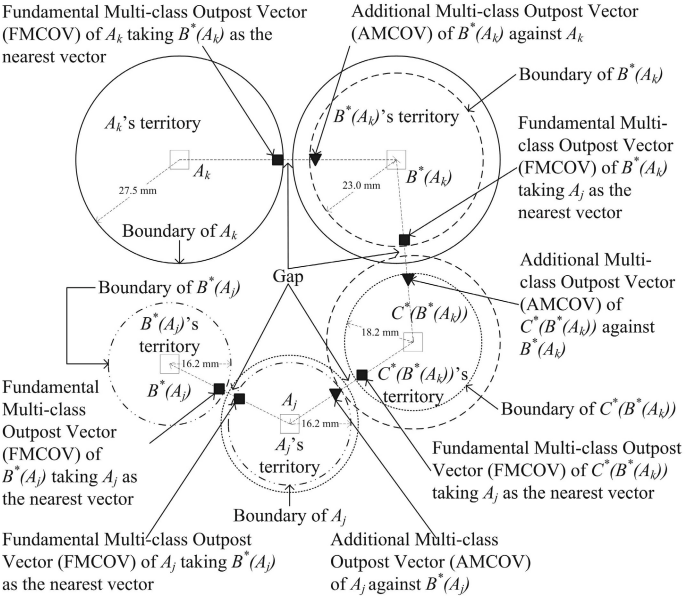


Fig. 3. Outpost vectors and their territory in a two-dimension three-class problem. The top left circle and bottom center circle are the territory of class A. The top right circle and bottom left circle are the territory of class B. The bottom right circle is the territory of class C.

circle (sphere in case of three-dimension space or hyper-sphere in case of more-dimension space) until the territories collide against any other.

The territory of input vector k of class A (denoted by A_k) is found by locating the input vector *in any other class* that is nearest to A_k. In this case, $B^*(A_k)$ of class B is nearest to A_k and referred to as A_k*'s pair* (denoted by $\phi(A_k)$). Then, the territory of A_k is declared at half way between A_k and $B^*(A_k)$. Consequently, the radius of A_k's territory is set at half of the distance between A_k and $B^*(A_k)$. This is to guarantee that if $B^*(A_k)$ sets its territory using the same radius, then the distance from the hyper-plane to either A_k or $B^*(A_k)$ will be at maximum. A_k then places its *fundamental outpost vector* (FMCOV) (denoted by $o(A_k)$) against $B^*(A_k)$ at the boundary of A_k's territory. The territories of $B^*(A_k)$ of class B, $C^*(B^*(A_k))$ of class C, A_j of class A and $B^*(A_j)$ of class B are also found by the same method done with A_k of class A.

After that, *additional multi-class outpost vectors* (AMCOVs) will then be generated from all input vectors as well. The AMCOV of $B^*(A_k)$ of class B (denoted by $o'(B^*(A_k))$) against A_k of class A is placed at the boundary of $B^*(A_k)$'s territory in the direction of A_k. The AMCOV of $C^*(B^*(A_k))$ of class C (denoted by $o'(C^*(B^*(A_k)))$) against $B^*(A_k)$ of class B, is placed at the boundary of $C^*(B^*(A_k))$'s territory in the direction of $B^*(A_k)$. The AMCOV of A_j of class A (denoted by $o'(A_j)$) against $C^*(B^*(A_k))$ of class C is placed at the boundary of A_j's territory in the direction of $C^*(B^*(A_k))$. The AMCOV of $B^*(A_j)$ of class B (denoted by $o'(B^*(A_j))$) against A_j of class A is placed at the boundary of $B^*(A_j)$'s territory in the direction of A_j. There is no AMCOV of A_k and $B^*(A_j)$ because $\phi(\phi(A_k))$ is A_k and $\phi(\phi(B^*(A_j)))$ is $B^*(A_j)$, respectively.

4 Simulation Results

The simulations are conducted on feed-forward neural network having parameters: network = 1 hidden layer, hidden neuron = 5, 10, 20, 30, 40 and 50, training function = Levenberg-Marquardt backpropagation, maximum epochs = 100 and performance goal = 0.01.

The data used in the simulations comprise six real world multi-class datasets from UCI machine learning repository [12] having various numbers of classes and dimensions of the input vectors. The characteristics of these datasets are presented in Table 1. The training sets are combined with FMCOVs, AMCOVs and both of them to construct a total of twenty four training sets used to evaluate the performance of FMCOV and AMCOV on multi-class data. The test results are produced by (1) and presented in Fig. 4.

$$MSE(I) = \frac{\sum_{i=1}^{n}(T_i - O_i)^2}{n}. \tag{1}$$

where I is a set of input vectors, n is the number of input vectors, T_i is the target value of input vector i and O_i is the output value of input vector i.

The test results are summarized as followings.

Table 1. Characteristics of Six Datasets from UCI Machine Learning Repository

UCI DATASET	CLASSES	FEATURES	TRAINING SAMPLES	TEST SAMPLES
Landsat Satellite (Statlog)	6	36	4,435	2,000
Shuttle (Statlog)	7	9	43,500	14,500
Poker Hand	10	10	25,010	1,000,000
Pen-Based Recognition	10	16	7,494	3,498
Optical Recognition	10	64	3,823	1,797
Letter Recognition	26	16	20,000	20,000

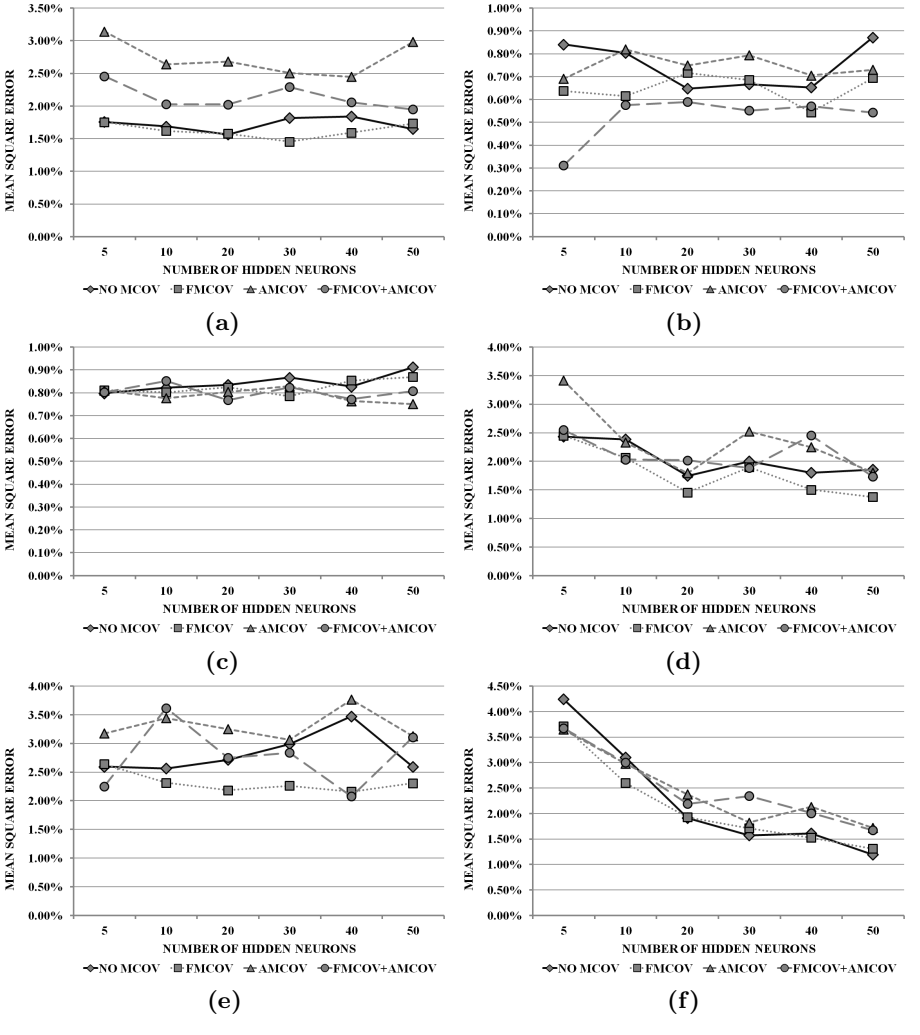


Fig. 4. Simulation Results of (a) Landsat Satellite (Statlog) (b) Shuttle (Statlog) (c) Poker Hand (d) Pen-Based Recognition of Handwritten (e) Optical Recognition of Handwritten (f) Letter Recognition Datasets from UCI Machine Learning Repository

1. The training sets with FMCOVs from five datasets (Landsat Satellite (Statlog), Shuttle (Statlog), Poker Hand, Pen-based Recognition of Handwritten and Optical Recognition of Handwritten) yield higher level of accuracy of classification than the training sets without outpost vector (NO MCOV). The training set with FMCOVs from Letter Recognition dataset yields higher level of accuracy of classification when the number of hidden neurons is small. However, it yields lower level of accuracy of classification when the number of hidden neurons is large.
2. The training set with AMCOVs from only one dataset (Poker Hand) yields higher level of accuracy of classification than the training sets without outpost vector (NO MCOV).
3. The training set with both FMCOVs and AMCOVs from only one dataset (Shuttle (Statlog)) yields higher level of accuracy of classification than the training sets without outpost vector (NO MCOV).

Based on the test results on six real world multi-class datasets, the training sets with FMCOVs generally yield highest level of accuracy of classification while the training sets with AMCOVs and both FMCOVs and AMCOVs rarely yield higher level of accuracy of classification. It is also noticeable that the training sets with any type of multi-class outpost vectors yield lower level of accuracy of classification on Letter Recognition dataset. It is probably because the distance between some input vectors in different classes is smaller than the gap between these input vectors. When the gap is larger, the input vectors placed within the gap can be misclassified.

As a result, it can be concluded that the multi-class contour preserving classification technique that generates only FMCOVs generally helps preserve the shape or distribution model of multi-class data more correctly and can assist the feed-forward neural network to classify the multi-class data more accurately.

5 Conclusions

The contour preserving classification technique was originally proposed to improve the robustness and weight fault tolerance of a neural network applied with a two-class linearly separable problem. Its major idea is to force non-linear classification on the linearly separable problem so as to take advantage of the nonlinear contour of the input vector regions to widen the clearance between the classification hyper-plane and the input vectors. It generates fundamental outpost vectors and additional outpost vectors at the boundary between two classes of data to preserve their shape or distribution model. The technique was recently found to be improving the level of accuracy of classification of feed-forward neural network on two-class data even when fewer numbers of input vectors are available. However, the technique supports only two-class data. This paper presents the augmentation of the original outpost vector generation algorithm to preserve the shape or distribution model of multi-class data to improve the level of accuracy of multi-class classification using feed-forward neural network. The multi-class outpost vector generation algorithm generates two types

of multi-class outpost vector: fundamental multi-class outpost vector (FMCOV) and additional multi-class outpost vector (AMCOV). Both multi-class outpost vectors are inserted into the training sets to preserve the shape or distribution model of the training data to assist the feed-forward neural network to classify the multi-class data more accurately. The simulations are conducted on six real world datasets from UCI machine learning repository having various numbers of classes and dimensions of the input vectors. The training sets are combined with FMCOVs, AMCOVs and both of them to construct a total of twenty four training sets used to evaluate the performance of FMCOV and AMCOV on multi-class data. The test results present that the proposed multi-class contour preserving classification technique supports multi-class data. In addition, the multi-class outpost vector generation algorithm generating FMCOVs generally helps preserve the shape or distribution model of multi-class data more correctly and can improve the level of accuracy of multi-class classification using feed-forward neural network more effectively.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, Upper Saddle River (1999)
2. Russell, S., Norving, P.: Artificial Intelligence A Modern Approach, 2nd edn. Pearson Education, Delhi (2004)
3. Negnevitsky, M.: Artificial Intelligence: A Guide to Intelligent Systems, 2nd edn. Addison Wesley, Essex (2005)
4. Tanprasert, T., Tanprasert, C., Lursinsap, C.: Contour preserving classification for maximal reliability. In: Int. Joint Conf. on Neural Networks, pp. 1125–1130 (1998)
5. Tanprasert, T., Fuangkhon, P., Tanprasert, C.: An improved technique for retraining neural networks in adaptive environment. In: Int. Conf. on Intelligent Technology, pp. 77–80 (2008)
6. Fuangkhon, P.: An incremental learning preprocessor for feed-forward neural network. Artificial Intelligence Review (2012), doi:10.1023/A:1022627411411
7. Mongkonsirivatana, J.: Neural network reliability enhancement with boundary detection contour preserving training. In: Int. Conf. on Intelligent Technology, pp. 234–239 (2005)
8. Tanprasert, T., Kripruksawan, T.: An approach to control aging rate of neural networks under adaptation to gradually changing context. In: Int. Conf. on Neural Information Processing, pp. 174–178 (2002)
9. Kaitikunkajorn, S., Tanprasert, T.: Improving synthesis process of decayed prior sampling technique. In: Int. Conf. on Intelligent Technology, pp. 240–244 (2005)
10. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
11. Chiu, S.: Fuzzy model identification based on cluster estimation. J. of Intelligent and Fuzzy Systems 2(3), 267–278 (1994)
12. Frank, A., Asuncion, A.: UCI Machine Learning Repository. School of Information and Computer Science. University of California (2010), http://archive.ics.uci.edu/ml

Automated Bone Age Assessment Using Feature Extraction

Luke M. Davis, Barry-John Theobald, and Anthony Bagnall

School of Computing Sciences
University of East Anglia
Norwich
UK
{luke.davis,b.theobald,anthony.bagnall}@uea.ac.uk
http://www.uea.ac.uk/cmp

Abstract. Bone age assessment is a task performed daily in hospitals worldwide, this involves a clinician estimating the age of a patient from a radiograph of the non-dominant hand. In this paper, we propose a combination of image processing and feature extraction algorithms to automatically predict the Tanner-Whitehouse bone stage, the assessment standard used in forming bone age estimates.

Keywords: Feature Extraction, Bone Age Assessment, Medical Imaging.

1 Introduction

This research is part of a wider project to build predictive models of bone age using hand radiograph images. Bone age assessment typically involves estimating the expected age of a patient from a radiograph by quantifying the development of the bones of the non-dominant hand. It is used to evaluate whether a child's bones are developing at an acceptable rate, and to monitor whether certain treatments are affecting a patient's skeletal development. Currently, this task is performed manually using a system such as Tanner and Whitehouse (TW) [1]. Manual procedures are time consuming and often inaccurate. We are developing a fully automated system that aims to produce an accurate estimate of bone age directly from a radiograph. Constructing an age estimate automatically involves segmenting the image to find the location of the relevant bones, then regressing information related to the bones on to age. In the segmentation stage we extract the hand outline [2] and then find the location of individual bones.We perform automated checks against false positives at the segmentation stage to stop incorrectly segmented images progressing to the next stage. Once we have the location of the individual bones, we perform the age regression in three stages. Firstly, we extract summary features we think are most powerful in predicting age. Secondly, we construct classifiers to predict the TW bone stages. Finally, we can recreate age using the standard TW approach of summing the bone scores and looking up the age mapping. We adopt this approach (rather than regressing directly on to age) so that our system is more transparent and explicable to

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 43–51, 2012.

clinicians. Evaluating the features discriminatory power for TW offers greater potential for explanatory insights, and will give the clinicians more confidence in the automated system. In this paper we concentrate on predicting TW scores for a single bone, the distal phalange of the third finger. In Section 2, we give a brief review of the TW system and describe other automated bone ageing algorithms. In Section 3, we outline how we extract the location of a bone in an image. In Section 4, we describe the features we derive from the bone, and how we calculate them. In Section 5, we perform an explanatory analysis to evaluate the discriminatory power of these features and in Section 6 we determine how well we can predict TW stages by constructing classifiers. Finally, we describe the future direction of this work in Section 7.

2 Background

The TW approach for bone age assessment involves grading a selection of bones based on a visual and textual description of the idealised stage of development. Table 1 shows the TW stages for the distal phalange of the third finger. The main region of bone is called the phalanx and the small region between phalanges is the epiphysis. TW stages are converted to scores by a fixed mapping. The scores are summed to give the skeletal maturity score, which is converted into a bone age using a look up chart. It has long been recognised that an automated bone age estimator is desirable and, in theory, achievable [1]. There have been numerous published automated bone ageing methodologies. Thodberg *et al.* [3] use an Active Appearance Model [4] for bone ageing. This algorithm involves fitting a model on 3000 manually labelled images. The resulting model is used to directly estimate the bone age, which can then be reverse engineered into a TW stage. Their algorithm is implemented in a commercial system called BoneXpert. Niemeijer *et al.* [5] outline a method to automate skeletal age assessment that uses Active Shape Models [4] to segment the distal phalange of the third finger. They construct a separate model on training data for each TW stage (E–I). For new data they use these models to extract the phalange, then measure the similarity of the fitted bone to the training bones in the model space. The predicted class is that of the nearest neighbour of the test case in the training data. They evaluate their system estimate TW score against objectively rated TW stages. Both of these approaches are hard to reproduce, since they require a large manually labelled training set, the nature of which will effect the resulting model. In addition, for AAM/ASM approaches, it is difficult to relate the discriminatory characteristics to the TW template bone shapes in order to explore how the classification is actually made. Our approach is to separate the image processing from the classification and to adopt methods that are more transparent and flexible. At this stage of the project we wish to mirror the TW approach as much as possible. This will help convince clinicians of the usefulness of the automation, and allow us to take an incremental approach to forming a final estimate that we believe will reduce the variance of the age prediction.

Table 1. The various TW stages of Distal Phalange 3 [1]

Stage	Image	Description
B		The centre is just visible as a single deposit of calcium, or more rarely as multiple deposits. The border is ill-defined.
C		The centre is distinct in appearance and disc-shaped, with a smooth continuous border.
D		The maximum diameter is half or more the width of the metaphysis.
E		The epiphysis is as wide as the metaphysis. The central potion of the proximal border has grown toward the end of the middle phalanx, so that the proximal border no longer consists of a single convex surface; no differentiation into palmar and dorsal surfaces, however, can yet be seen.
F		Palmar and dorsal proximal surfaces are distinct, and each has shaped to the trochlear articulation of the middle phalanx. The palmar surface appears as a projection proximal to the thickened white line representing the dorsal surface.
G		The epiphysis caps the methaphysis.
H		Fusion of epiphysis and metaphysis has now begun.
I		Fusion of epiphysis and metaphysis is completed.

3 Extracting the Bone Outline

100 radiographs were taken from the dataset provided by [6] and TW scores were assigned to the distal phalange of the third finger using Table 1. We perform a three stage automated bone segmentation. First, we find the hand outline using the algorithm described in [2]. The tips and webs are then calculated using peak detection. From the tips and webs, we calculate the axis for the middle finger by calculating the midpoint between the adjacent webs. The axis of the finger is the midpoint of the base to the tip. Using this, we compute a region-of-interest box. An example segmentation is shown in Figure 1(a).

(a) (b)

Fig. 1. Example bone image of the Distal Phalange of the Third Finger (a), and final segmentation (b)

The final step of the image processing involves segmenting the bone from the bounding box found in the previous stage. Simple outlining techniques prove insufficient because the high resolution intensity detail within the bounding box often

leads to an incorrect bone outlining. Down-sampling the image improves the segmentation, but it is crucial not to down-sample too much, since the distinction between the phalanx and epiphysis can be lost. Hence, we constructed a Gaussian pyramid [7] hierarchy of down-sampled images and performed Canny edge detection [8] at each level of the pyramid. The edges found at level n form the basis for the the edge detection at stage $n-1$. Any edges that were connected to the sides of the bounding box, or were not complete loops, were removed. The region inside the largest remaining loop is labelled as the phalanx. If a second loop is present it is labelled the epiphysis. An example final segmentation is shown in Figure 1(b).

4 Feature Extraction from the Outline

Our goal is to be able to automatically extract the features that best capture the variability described in the text for classifying TW stages (see Tables 1 and 2).

Table 2. Features derived from looking at Tanner-Whitehouse Stages

Feature Number	Feature Name
1	Epiphysis
2	Phalanx Ellipse Height
3	Phalanx Ellipse Width
4	Phalanx Height
5	Phalanx Width
6	Phalanx First Quartile Width
7	Phalanx Third Quartile Width
8	Metaphysis (Phalanx Ninety Percentile) Width
9	Phalanx Eccentricity
10	Phalanx Width to Height Ratio
11	Phalanx Roundness
12	Phalanx Area to Perimeter Ratio
13	Phalanx First Quartile to Width Ratio
14	Phalanx Third Quartile to Width Ratio
15	Phalanx Metaphysis to Width Ratio
16	Epiphysis Ellipse Height
17	Epiphysis Ellipse Width
18	Epiphysis Height
19	Epiphysis Width
20	Epiphysis Eccentricity
21	Epiphysis Distance to Phalanx
22	Epiphysis Width to Height Ratio
23	Epiphysis Roundness
24	Epiphysis Area to Perimeter Ratio
25	Epiphysis Width to Metaphysis Ratio

Note that we are currently only calculating shape features. Shape features are clearly the most discriminatory, although the intensity features may well help in the finer distinctions between the stages, and will be included at a later date. The dimensions and shape of the phalanx, the epiphysis, and the region of the phalanx bordering the epiphysis (called the metaphysis) are crucial in forming the classification.

The most important feature is whether the epiphysis is present or not (feature 1). The other features are summary measures of the phalanx and the epiphysis (if present).

Basic size descriptors such as height and width are obviously going to be reasonably indicative of age, although they are of less use than one might first think because the size of the image of the hand does not directly map to the size of the actual hand. This is because the focus of the radiograph machine is

adjusted so that the hand image is approximately the same size independent of actual size. The obvious way to find the height and width of the phalanx and epiphysis is to find the length of the vertical line down the centre of the bone for height and the length of the horizontal line across the middle of the vertical for width. However, this assumes the bones are vertically aligned, which is often not the case, since fingers are often not straight. In order to calculate an estimate for height and width, we fit an ellipse to both the phalanx and epiphysis (if present).

A standard way of fitting an ellipse to an image is to use the Hough transform [9]. However, this transform has time complexity $O(n^5)$, where n is the number of pixels. It is faster therefore to fit the transform on a lower resolution image. The algorithm is also much faster with a good initial approximation of the ellipse. Hence, we once again use a Gaussian pyramid of incrementally downsized images. We fit an ellipse at the lowest resolution with the Hough transform, then use this ellipse as the initial starting point for the transform at the next highest level. We use the axes of the ellipse generated at level 1 to calculate an estimate for the height and width of the phalanx and epiphysis (features 2,3,16 and 17).

We also use the vertical axis of the ellipse to calculate the width of the phalanx at certain points along its length. The objective is to capture the change from a fairly straight sided bone in the early stages, to one that narrows in the middle in the latter stages. In addition to finding the width at the middle of the axis (feature 5), we also find the first quartile and third quartile width (features 6 and 7), and the ratio of the quartile widths to the middle width (features 13 and 14). A further way of capturing the progressive change in the phalanx from fairly round to stretched and irregular is to measure how circular the ellipse of the bone is. A standard measure of roundness of an ellipse is the eccentricity e, (feature 9 and 20) which is calculated using Equation 1. The half length of the semi-major axis and half the length of the semi-minor axis are referred to as h and w respectively.

$$e = \sqrt{1 - \left(\frac{h}{w}\right)^2} \tag{1}$$

We also estimate the roundness directly from the bone itself. The two roundness features 11 and 23 were calculated using Equation 2, where a is the area of the binary mask and p is the length of the the perimeter.

$$r = \frac{4\pi a}{p^2} \tag{2}$$

Another key discriminatory characteristic is the distance from epiphysis to the phalanx. We quantify this as the Euclidean distance between the midpoints of the two (feature 21).

5 Exploratory Analysis of Features

Our first goal is to evaluate how well the features in Table 2 can explain the variation in TW stage. Table 3 shows the features ranked independently by

information gain (in brackets). Information gain is calculated from the labelled data and is used to determine how well a feature can distinguish classes. The first column shows the rank order of the phalanx features on all of the images. The presence of the epiphysis is the most important feature, followed by phalanx height (features 2 and 4). Measures related to the metaphysis (features 8 and 15) are important, as are the shape descriptors (features 10 and 11). The second column of Table 3 shows the feature ranks for when the epiphysis is distinct (i.e. stages C–G). The top positioned phalanx feature, height, is only ranked seventh. The top six features all relate to measurements of the epiphysis.

Table 3. Features ranked by information gain

Phalanx feature ranks	Ranking of features when epiphysis distinct
1(0.8424)	**25** (0.6126)
2 (0.7556)	**17** (0.6108)
4 (0.7556)	**19** (0.5551)
8 (0.7552)	**24** (0.5551)
11 (0.7140)	**16** (0.3985)
10 (0.6583)	**21** (0.3985)
15 (0.6408)	**4** (0.3933)
7 (0.6350)	**12** (0.3933)
9 (0.6312)	**23** (0.3774)
14 (0.5533)	**2** (0.3771)
12 (0.5436)	**5** (0.3580)
5 (0.4291)	**8** (0.3580)
3 (0.4061)	**11** (0.3464)
6 (0.3925)	**3** (0.3405)
13 (0.1775)	**6** (0.2861)

In Table 4, we show the importance of each feature for each TW stage using information gain, with the most important features in bold. Table 4 shows that different features are important in determining different stages. The results from stages where the epiphysis is present show again that the features extracted from this area are the most important.

Table 4. The importance of each feature against each TW stage

Feature Number	Stage B	Stage C	Stage D	Stage E	Stage F	Stage G	Stage H	Stage I
1) Epiphysis	0.0594	0.0067	0.1539	0.1110	0.1120	0.0171	0.2495	0.2495
2) Phalanx Ellipse Height	0.1382	0.0288	0.2254	0.1149	0.0922	0.0468	0.2650	0.2732
3)Phalanx Ellipse Width	**0.1872**	0.0447	0.1648	0.0499	0.0441	0.0356	0.1542	0.0986
4) Phalanx Height	0.1383	0.0299	0.2269	0.1149	0.0922	0.0412	0.2732	0.2650
5) Phalanx Width	0.1531	0.0447	0.1927	0.0529	0.0441	0.0356	0.1533	0.0958
6) Phalanx First Quartile Width	0.1733	0.0259	0.1570	0.0529	0.0384	0.0749	0.1143	0.2183
7) Phalanx Third Quartile Width	0.1321	0.0483	0.1891	0.0886	0.0851	0.0559	0.2376	0.2352
8) Metaphysis Width	0.1531	0.0311	0.2087	0.1110	0.0922	0.0716	**0.3001**	0.2745
9) Phalanx Eccentricity	0.0344	0.0126	0.2084	0.1110	0.0851	0.0356	0.1744	0.2571
10) Phalanx Width to Height Ratio	0.0366	0.0141	0.1769	0.1194	0.0922	0.0275	0.1966	**0.3001**
11) Phalanx Roundness	0.0578	0.0157	0.2221	0.1543	0.0652	0.0412	0.2025	0.2495
12) Phalanx Area to Perimeter Ratio	0.1452	0.0418	0.2269	0.1033	0.0652	0.0652	0.2311	0.2350
13) Phalanx First Quartileto Width Ratio	0.0333	0.0532	0.0647	0.0527	0.0301	0.0499	0.0241	0.0966
14) Phalanx Third Quartile to Width Ratio	0.0377	0.0141	0.1082	0.1359	0.1033	0.0583	0.2281	0.1698
15) Phalanx Metaphysis to Width Ratio	0.0462	0.0214	0.1177	0.1641	0.0792	0.0440	0.2421	0.2376
16) Epiphysis Ellipse Height	*N/A*	0.0133	0.1595	0.1908	0.1906	0.0548	*N/A*	*N/A*
17) Epiphysis Ellipse Width	*N/A*	0.0141	0.1595	**0.2025**	**0.2281**	0.1711	*N/A*	*N/A*
18) Epiphysis Height	*N/A*	0.0126	0.1539	0.1797	0.1852	0.0698	*N/A*	*N/A*
19) Epiphysis Width	*N/A*	0.0141	0.1595	0.1966	**0.2281**	0.1450	*N/A*	*N/A*
20) Epiphysis Eccentricity	*N/A*	0.0126	0.1539	0.1190	0.1495	0.0933	*N/A*	*N/A*
21) Epiphysis Distance to Phalanx	*N/A*	0.0110	0.1595	0.1908	0.1641	0.0794	*N/A*	*N/A*
22) Epiphysis Width to Height Ratio	*N/A*	0.0130	0.1539	0.1273	0.1641	0.0787	*N/A*	*N/A*
23) Epiphysis Roundness	*N/A*	**0.0808**	**0.2718**	0.1190	0.1149	0.0620	*N/A*	*N/A*
24) Epiphysis Area to Perimeter Ratio	*N/A*	0.0141	0.1652	0.1966	0.2086	0.1202	*N/A*	*N/A*
25) Epiphysis Width to Metaphysis Ratio	*N/A*	0.0141	0.1595	0.1744	0.2214	**0.1822**	*N/A*	*N/A*

The final stage of our exploratory analysis is to examine how the features can interact to classify all the stages. Figure 2 shows the C4.5 decision tree constructed on all 100 instances. This tree illustrates some simple decision rules that encapsulate the description of how to determine the stage given in Table 1. So, for example, the rule ***if*** *no epiphysis* ***then*** *B, G, H or I* describes the main left hand branch of the tree and Stage B is classified by the rule ***if*** *no epiphysis* ***and*** *phalanx small* ***then*** *B.*

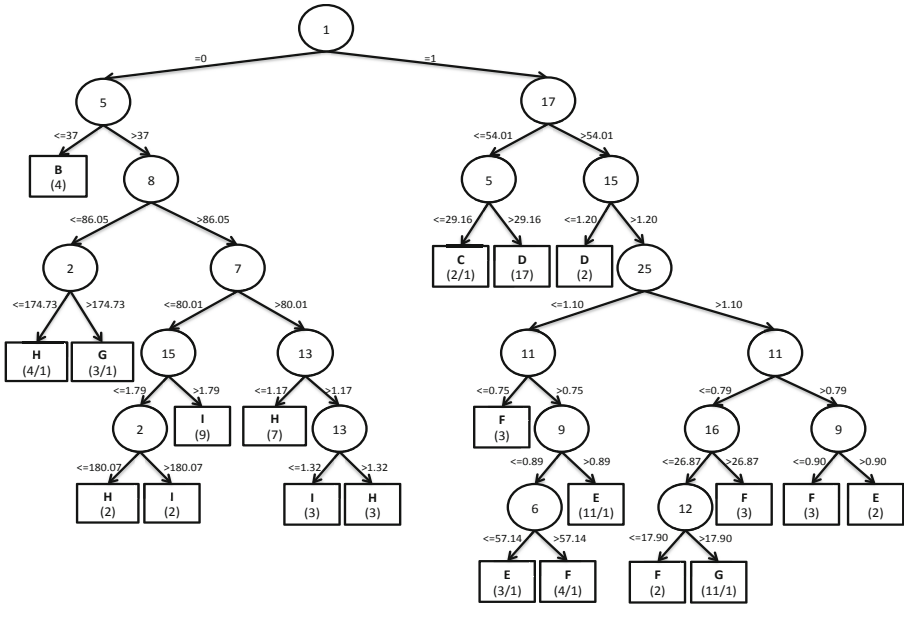


Fig. 2. C4.5 Decision Tree trained on whole dataset

6 Classification of Tanner-Whitehouse Stages

Whilst the exploratory analysis allows us to examine the importance of the features and how they interact, we are also concerned with the predictive power of models constructed in this feature space. Table 5 shows the test data confusion

Table 5. Confusion Matrix for C4.5 tree from ten run, ten fold cross validation

	B	C	D	E	F	G	H	I
B	40	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0
D	0	10	157	25	0	10	0	0
E	0	0	33	48	52	15	0	0
F	0	0	9	56	58	48	0	0
G	0	0	0	21	40	49	26	10
H	0	0	0	0	0	17	53	70
I	0	0	0	0	0	11	71	70

matrix for a C4.5 classifier found using a repeated cross validation. Whilst the classification accuracy is only 47.5%, the vast majority of classifications (92.3%) are within one category of the the true class.

In addition to a C4.5 decision tree, we have applied a variety of alternative classifiers on the extracted features. The results from ten run, ten fold cross validation are summarised in Table 6. All the classifiers are in the range 46%-56% accuracy, but all are getting close, with a best result of 95.5% within one stage of the correct stage. Given stage classification is subjective, we believe this is an encouraging result and that with more data we will form more accurate classifiers.

Table 6. Results from ten run, ten fold cross validation on on TW Stages B-I. The first column refers to the percentage of images with correct stage classified and the second column the percentage within one TW stage of the correct stage.

Classifier	Correct Stage %	Within One Stage %
3-NN	55.3	86.9
Naive Bayes	46.9	93.0
C4.5	47.5	92.3
SVM	53.3	95.5
Bayesian Network	48.2	95.0
Random Forest	52.1	92.8
Rotation Forest	52.5	94.8
Logistic Regression	53.9	94.2

7 Conclusions and Future Work

In this paper we have described a system to predict TW stages using automated feature extraction. We believe separating the image processing from the classification/regression will ultimately be a more robust approach. This transparent modular approach allows us to reduce the risk of incorrect image processing corrupting the age prediction and provides a means for clinicians to investigate exactly how the system operates. We believe this white box approach is crucial in order to gain widespread acceptance.

In the future we plan to investigate incorporating more features, such as those based on image intensity. We will extend the system to include more bones, and hence produce age estimates from combined TW scores. We will assess the system by comparing the automated age estimates to independently made clinical estimates and to the chronological age. Finally, we will investigate regressing features directly onto age, to determine the benefits or otherwise of the TW scoring system.

References

1. Tanner, J., Whitehouse, R., Marshall, W., Healy, M., Goldstein, H.: Assessment of skeletal maturity and prediction of adult height (TW2 method), vol. 16. Academic Press, London (1975)

2. Davis, L., Theobald, B.J., Toms, A., Bagnall, A.: On the Extraction and Classification of Hand Outlines. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 92–99. Springer, Heidelberg (2011)
3. Thodberg, H., Kreiborg, S., Juul, A., Pedersen, K.: The BoneXpert method for automated determination of skeletal maturity. IEEE Transactions on Medical Imaging 28(1), 52–66 (2009)
4. Cootes, T., Edwards, G., Taylor, C.: Comparing active shape models with active appearance models. In: British Machine Vision Conference, vol. 1, pp. 173–183 (1999)
5. Niemeijer, M., van Ginneken, B., Maas, C., Beek, F., Viergever, M.: Assessing the skeletal age from a hand radiograph: automating the Tanner-Whitehouse method. In: SPIE Medical Imaging, vol. 5032, pp. 1197–1205 (2003)
6. Cao, F., Huang, H., Pietka, E., Gilsanz, V.: Digital hand atlas and web-based bone age assessment: system design and implementation. Computerized Medical Imaging and Graphics 24(5), 297–307 (2000)
7. Adelson, E., Anderson, C., Bergen, J., Burt, P., Ogden, J.: Pyramid methods in image processing. RCA Engineer 29(6), 33–41 (1984)
8. Canny, J.: A computational approach to edge detection. Readings in Computer Vision: Issues, Problems, Principles, and Paradigms 184 (1987)
9. Ballard, D.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition 13(2), 111–122 (1981)

A Hybrid Approach Based on DCT-Genetic-Fuzzy Inference System for Speech Recognition

Washington Silva and Ginalber Serra

Federal Institute of Education, Science and Technology
Department of Electroelectronics,
Laboratory of Computational Intelligence Applied to Technology
Av. Getúlio Vargas, 04, Monte Castelo, CEP: 65030-005, São Luis, Maranhão, Brazil
{washington.wlss,ginalber}@ifma.edu.br
http://www.ifma.edu.br

Abstract. The concept of fuzzy sets and fuzzy logic is widely used to propose of several methods applied to systems modeling, classification and pattern recognition problem. This paper proposes a genetic-fuzzy recognition system for speech recognition. In addition to pre-processing, with mel-cepstral coefficients, the Discrete Cosine Transform (DCT) is used to generate a two-dimensional time matrix for each pattern to be recognized. A genetic algorithms is used to optimize a Mamdani fuzzy inference system in order to obtain the best model for final recognition. The speech recognition system used in this paper was named Hybrid DCT-Genetic-Fuzzy Inference System for Speech Recognition **(HGFIS)**.

Keywords: Recognition Speech, Fuzzy Sistems, Optimization, Genetic Algorithm, Discrete Cosine Transform.

1 Introduction

Parameterization of an analog speech signal is the first step in speech recognition process. Several popular signal analysis techniques have emerged as standards in the literature. These algorithms are intended to produce a 'perceptually meaningful' parametric representation of the speech signal [1]. The goal of selecting the best way to encode the signal is to compress the speech data information, eliminating non-phonetic analysis of the signal and improving the aspects of the signal which contributes significantly to detect phonetic differences of speech sounds [2],[3]. One of the most widespread techniques for pattern speech recognition is the "Hidden Markov Model" (HMM) [4]. A well known deficiency of the classical HMMs is the poor modeling of the acoustic events related to each state. Since the probability of recursion to the same state is constant, the probability of the acoustic event related to the state is exponentially decreasing. This probability distribution does not model the speech temporal structure. A second weakness of the HMMs is that the observation vectors within each state are assumed uncorrelated, and these vectors are correlated [5],[6]. In this paper, a speech signal is encoded and parameterized in a two-dimensional time

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 52–59, 2012.

matrix. After coding, the mean and variance of each pattern are used to generate the rule base of Mamdani fuzzy inference system. The mean and variance are optimized using genetic algorithm in order to have the best performance of the recognition system. Consider as patterns the brazilian locutions (digits): $'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'$. The Discrete Cosine Transform (DCT) [7],[8] is used to encoding the speech patterns. This paper demonstrates the potential of DCT and fuzzy inference system in speech recognition [9],[10].

2 Speech Recognition System

The proposed recognition system HGFIS block diagram is depicted in Fig.1.

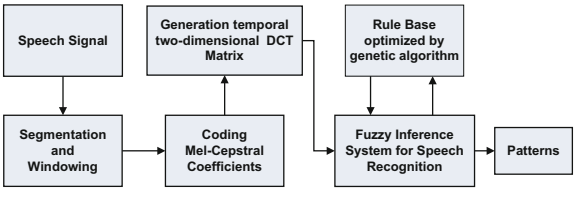


Fig. 1. Block diagram of the proposed recongnition system HGFIS

Initially, the speech signal is digitizing, so it is divided in frames which are windowed and encoded in a set of parameters defined by the order of mel-cepstral coefficients (MFCC). The DCT coefficients are computed and the two-dimensional time DCT matrix is generated, based on each speech signal to be recognized [11]. Let $mfcc$ the mel-cepstral coefficients. The two-dimensional time matrix is the result of DCT in a sequence of T mel-cepstral coefficients observation vectors on the time axis, given by:

$$C_k(n,T) = \frac{1}{N}\sum_{t=1}^{T} mfcc_k(t) cos\frac{(2t-1)n\pi}{2T} \quad (1)$$

where $k, 1 \leq k \leq K$, is the k-th (line) component of t-th frame of the matrix and $n, 1 \leq n \leq N$ (column) is the order of DCT. Thus, there is a two-dimensional time matrix $C_k(n,T)$ for each input speech signal. The elements of the matrix are obtained as follows: For a given spoken word $\boldsymbol{P}$(digit), ten examples of utterances of $\boldsymbol{P}$are gotten. Each examples is properly encoded in T frames distributed along the time axis. Each frame of the word $\boldsymbol{P}$ generates a total of K mel-cepstral coefficients and the significant features are taken for each frame. The *N-th* order DCT is computed for each mel-cepstral coefficient of same order within the frames distributed along the time. Thus, a two-dimensional time matrix DCT is generated for each example of the word $\boldsymbol{P}$, represented by: C_{kn}, where $k = 1, 2, ...K$ and $n = 1, 2, ..., N$.

$$\mathbf{C_0^0}=\begin{pmatrix} c_{11}^{00} & c_{12}^{00} \\ c_{21}^{00} & c_{22}^{00} \end{pmatrix}\mathbf{C_1^0}=\begin{pmatrix} c_{11}^{01} & c_{12}^{01} \\ c_{21}^{01} & c_{22}^{01} \end{pmatrix}\ldots\mathbf{C_9^0}=\begin{pmatrix} c_{11}^{09} & c_{12}^{09} \\ c_{21}^{09} & c_{22}^{09} \end{pmatrix}$$

$$\mathbf{C_0^1}=\begin{pmatrix} c_{11}^{10} & c_{12}^{10} \\ c_{21}^{10} & c_{22}^{10} \end{pmatrix}\mathbf{C_1^1}=\begin{pmatrix} c_{11}^{11} & c_{12}^{11} \\ c_{21}^{11} & c_{22}^{11} \end{pmatrix}\ldots\mathbf{C_9^1}=\begin{pmatrix} c_{11}^{19} & c_{12}^{19} \\ c_{21}^{19} & c_{22}^{19} \end{pmatrix}$$

$$\vdots$$

$$\mathbf{C_0^9}=\begin{pmatrix} c_{11}^{90} & c_{12}^{90} \\ c_{21}^{90} & c_{22}^{90} \end{pmatrix}\mathbf{C_1^9}=\begin{pmatrix} c_{11}^{91} & c_{12}^{91} \\ c_{21}^{91} & c_{22}^{91} \end{pmatrix}\ldots\mathbf{C_9^9}=\begin{pmatrix} c_{11}^{99} & c_{12}^{99} \\ c_{21}^{99} & c_{22}^{99} \end{pmatrix}$$

2.1 Fuzzy Rule Base for Speech Recognition

A fuzzy rule base Ru consists of a set of fuzzy $IF-THEN$ rules. From the coefficients of the matrices $\mathbf{C}_{kn}^{j}$ with $j = 0, 1, 2, ..., 9$, $k = 1, 2$ and $n = 1, 2$ generated during the training process, representing the mean and variance of each pattern j a rule base with $M = 40$ individual rules is obtained and given by:

$$Ru^j : IF \;\; C_{kn}^{j} \;\; THEN \;\; y^j \tag{2}$$

In this paper, the training process is based on the fuzzy relation Ru^j using the Mamdani implication. The rule base Ru^j should be considered a relation $R(X \times Y) \rightarrow [0, 1]$, computed by:

$$\mu_{Ru}(x, y) = I(\mu_A(x), \mu_B(y)) \tag{3}$$

where the operator I should be any t-norm [12]. Given the fuzzy set A' input, the fuzzy set B' output might be obtained by ***max-min*** composition, [13]. For a minimum t-norm and max-min composition it yields:

$$\mu_{(B')} = max_x min_{x,y}(\mu_{A'}(x), \mu_{(Ru)}(x, y)) \tag{4}$$

2.2 Generation of Fuzzy Patterns

The elements of the matrix C_{kn}^{j} were used to generate gaussians membership functions in the process of fuzzification. For each trained model j the gaussians memberships functions $\mu_{c_{kn}^{j}}$ are generated, corresponding to the elements c_{kn}^{j} of the two-dimensional time matrix $\mathbf{C}_{kn}^{j}$ with $j = 0, 1, ..., 9$, where j is the model used in training. The training system for generation of fuzzy patterns is based on the encoding of the speech signal ***s(t)***, generating the parameters of the matrix C_{kn}^{j}. Then, these parameters are fuzzified, and they are related to properly fuzzified output y^j by the relational implications, generating a relational surface $\mu_{(Ru)}$, given by:

$$\mu_{Ru} = \mu_{c_{kn}^{j}} \circ \mu_{y^j} \tag{5}$$

This relational surface is the fuzzy system rule base for recognition optimized by genetic algorithm to maximize the speech recognition. The decision phase is performed by a fuzzy inference system based on the set of rules obtained from the mean and variance matrices of two dimensions time of each spoken digit. The elements of the matrices $\mathbf{C}_{kn}^{j}$ are used by the fuzzy inference system to generate four gaussian membership functions corresponding to each element $c_{kn}^{j}\Big|^{k=1,2;n=1,2}$ of the matrix. The set of rules of the fuzzy relation is given by:

$$\textbf{\textit{IF}} \quad c_{kn}^{j}\Big|^{k=1,2;n=1,2} \quad \textbf{\textit{THEN}} \quad y^{j} \tag{6}$$

After the training process, the relational surfaces is generated based on the rule base and implication method. The speech signal is encoded to be recognized and their parameters are evaluated in relation to the functions of each patterns on the surfaces and the degree of membership is obtained. The final decision for the pattern is taken according to the $max - min$ composition between the input parameters and the data contained in the relational surfaces. The process of defuzzification for the pattern recognition is based on the ***mean of maxima (mom)*** method given by:

$$y^{'} = mom(\mu_{y'^{j}}) = mean\{y|\mu_{y'^{j}} = max_{y\in Y}(\mu_{y'^{j}})\} \tag{7}$$

2.3 Optimization of Relational Surface with Genetic Algorithm

The continuous genetic algorithm (GA) is configured with a population size of 100, generations of 300, with mutations probability of 15% and two individuals chromosomes with 40 genes each, to optimize a cost function with 80 variables, which are the mean and variances of the patterns to be recognized by the proposed fuzzy recognition system [14]. For each element of the matrix C_{kn}^{j} coefficients are determined with variations minimum and maximum , and the coefficient $c_{11} \in [c_{11(minimum)}\ c_{11(maximum)}]$, $c_{12} \in [c_{12(minimum)}\ c_{12(maximum)}]$, $c_{21} \in [c_{21(minimum)}\ c_{21(maximum)}]$, $c_{22} \in [c_{22(minimum)}\ c_{22(maximum)}]$[15].

3 Experimental Results

3.1 System Training

After pre-processing of the speech signal and fuzzification of the matrix C_{kn}^{j}, its fuzzifieds components $\mu_{c_{kn}^{j}}$ had been optimized by the GA that maximize the total of successful recognition. The optimization process was performed with 16 realizations of the genetic algorithm, whose results are shown in Fig.2. The best result of the recognition processing by HGFIS is shown in Fig.3. The total number of hits using GA was 92 digits correctly identified in the training process. The relational surface generated for this result was used for validation process. The best individual in the first generation is shown in Fig.4. In this case the total number of correct answers was 46 digits correctly identified. The ralational surface of the best individual in the first generation is shows in Fig.5. The optimum individual, HGFIS, presents the features in Fig.6 and Fig.7.

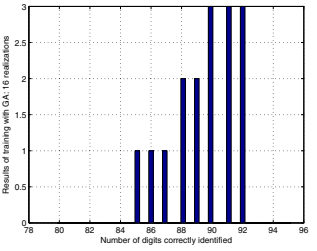


Fig. 2. Histogram for 16 realizations of the training process with the HGFIS

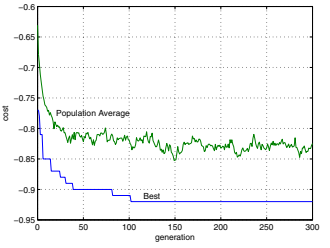


Fig. 3. Plot of the best results obtained in the training process

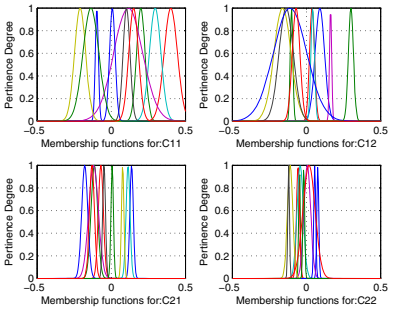


Fig. 4. Membership functions for c_{kn}^{j} in the 1st generation

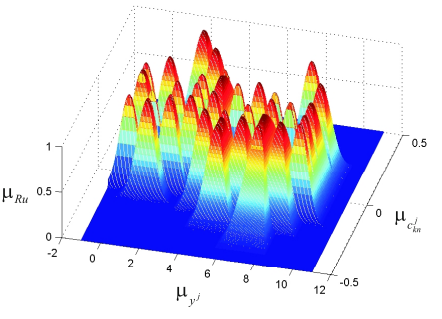


Fig. 5. Relational surface (μ_{Ru}) in the 1st generation

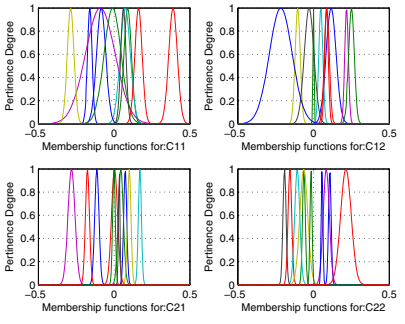


Fig. 6. Membership functions for c_{kn}^{j} optmized by GA

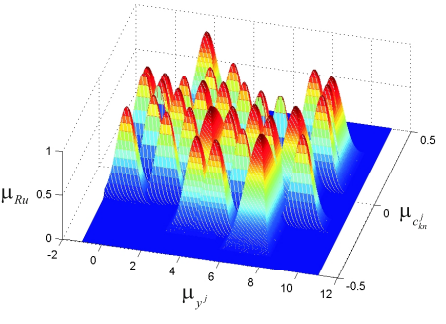


Fig. 7. Relational surface (μ_{Ru}) optmized by GA

3.2 System Test - Validation

In this step, 100 locutions uttered in a room with controlled noise level and 500 locutions uttered in an environment without any kind of noise control were used.

For every ten examples of each spoken digit, was generated two-dimensional time matrix cepstral coefficients $\mathbf{C}_{kn}^{j}$ and they were used in the test procedure. Where performed the training process and five types of tests. To test the ability of characteristics extraction of the method, male and female speakers were used.

TRAINING: Recognition Optimized by HGFIS (5 Female and 5 Male Speakers);

TEST 1: Validation - Strictly speaker dependent recognition, where the words used for training and testing were spoken by a same group of 10 speakers(5 Female and 5 Male Speakers);

TEST 2: Validation test- Recognition based on the partial dependence of the speaker with two examples for each ten examples of each digit(Female Speaker);

TEST 3: Validation test- Recognition based on the partial dependence of the speaker with two examples for each ten examples of each digit(Male Speaker);

TEST 4: Validation test- Recognition independent of the Speaker, where the speaker does not have influence in the training process(Female Speaker);

TEST 5: Validation test- Recognition independent of the Speaker, where the speaker does not have influence in the training process(Male Speaker).

The tables from 1 to 6 presents the comparative analysis of the HMM with three state, three gaussian mixture by state and order analysis, i.e., the number of mel-cepstral parameter equal 8 and 12. The number of hits, using the HMM and HGFIS for speech recognition. In the table for HMM, $sn = state\ number$, and $pn = parameters\ number(\ Mel\text{-}cepstrais\ coefficients)$.

Table 1. Results in the training

Brazilian Digits	English Digits	HMM		HMM		HGFIS
		sn=3	*pn=8*	*sn=3*	*pn=12*	*pn=4*
ZERO	(zero)	9		10		10
UM	(one)	9		10		8
DOIS	(two)	7		8		10
TRES	(three)	8		9		9
QUATRO	(four)	7		8		8
CINCO	(five)	10		8		10
SEIS	(six)	7		10		10
SETE	(seven)	9		10		9
OITO	(eight)	10		10		10
NOVE	(nine)	9		8		8
Total(%)		85%		91%		92%

Table 2. Validation Test 1

Brazilian Digits	English Digits	HMM		HMM		HGFIS
		sn=3	*pn=8*	*sn=2*	*pn=12*	*pn=4*
ZERO	(zero)	9		10		9
UM	(one)	9		10		8
DOIS	(two)	7		7		9
TRES	(three)	8		8		8
QUATRO	(four)	7		8		9
CINCO	(five)	10		10		10
SEIS	(six)	7		8		9
SETE	(seven)	9		9		9
OITO	(eight)	9		9		10
NOVE	(nine)	9		9		9
Total(%)		84%		88%		90%

Table 3. Validation Test 2

Brazilian Digits	English Digits	HMM		HMM		HGFIS
		sn=3	*pn=8*	*sn=3*	*pn=12*	*pn=4*
ZERO	(zero)	9		9		10
UM	(one)	9		9		7
DOIS	(two)	6		6		7
TRES	(three)	10		9		8
QUATRO	(four)	9		9		8
CINCO	(five)	6		7		10
SEIS	(six)	6		7		6
SETE	(seven)	6		7		9
OITO	(eight)	7		8		7
NOVE	(nine)	9		9		9
Total(%)		77%		80%		81%

Table 4. Validation Test 3

Brazilian Digits	English Digits	HMM		HMM		HGFIS
		sn=3	*pn=8*	*sn=3*	*pn=12*	*pn=4*
ZERO	(zero)	7		9		8
UM	(one)	8		9		8
DOIS	(two)	7		8		10
TRES	(three)	6		8		7
QUATRO	(four)	7		8		9
CINCO	(five)	8		8		8
SEIS	(six)	8		7		9
SETE	(seven)	7		8		9
OITO	(eight)	9		9		8
NOVE	(nine)	8		9		8
Total(%)		75%		83%		84%

Table 5. Validation Test 4

Brazilian Digits	English Digits	HMM		HMM		HGFIS
		sn=3	*pn=8*	*sn=3*	*pn=12*	*pn=4*
ZERO	(zero)	6		6		10
UM	(one)	2		3		2
DOIS	(two)	4		5		5
TRES	(three)	5		5		8
QUATRO	(four)	5		7		4
CINCO	(five)	7		8		10
SEIS	(six)	4		8		5
SETE	(seven)	5		6		9
OITO	(eight)	4		6		4
NOVE	(nine)	5		5		9
Total(%)		49%		59%		66%

Table 6. Validation Test 5

Brazilian Digits	English Digits	HMM		HMM		HGFIS
		sn=3	*pn=8*	*sn=3*	*pn=12*	*pn=4*
ZERO	(zero)	4		5		8
UM	(one)	5		9		4
DOIS	(two)	9		9		4
TRES	(three)	3		4		3
QUATRO	(four)	4		5		5
CINCO	(five)	9		7		10
SEIS	(six)	5		6		5
SETE	(seven)	8		6		8
OITO	(eight)	9		8		10
NOVE	(nine)	6		6		10
Total(%)		62%		65%		67%

4 Conclusion

Evaluating the results, it is observed that the proposed speech recognizer HGFIS, even with a minimal parameters number in the generated patterns was able to extract more reliably the temporal characteristics of the speech signal and produce good recognition results compared with the traditional HMM. To obtain equivalent results with HMM is necessary to increase the state number and/or mixture number. Any particular technique of noise reduction, such as those commonly used in HMM-based recognizers, was not used during the development of this paper. It is believed that with proper treatment of the signal to noise ratio in the process of training and testing, the HGFIS Recognizer may improve its performance: Increase the speech bank with different accents; Improve the performance of genetic algorithm to 100% recognition in the training process and increase the parameters number used in HGFIS.

Acknowledgment. The authors would like to thank FAPEMA for financial support, research group of computational intelligence applied to technology at the Federal Institute of Education, Science and Technology of the Maranhão by its infrastructure for this research and experimental results, and the Master and PhD program in Eletrical Engineering at the Federal University of Maranhão (UFMA).

References

1. Picone, J.W.: Signal modeling techiniques in speech recognition. IEEE Transactions on Computer 79(4), 1214–1247 (1991)
2. Rabiner, L., Biing-Hwang, J.: Fundamentals of Speech Recognition. Prentice Hall, New Jersey (1993)
3. Andrews, H.C.: Multidimensional Rotations in Feature Selection. IEEE Transaction on Computers (September 1971)
4. Abushariah, A.A.M., Gunawan, T.S., Khalifa, O.O.: English Digits Speech Recognition System Based on Hidden Markov Models. In: International Conference on Computer and Communication Engineer (ICCCE 2010), Kuala Lumpur, Malaysia (May 2010)
5. De Wachter, M., Matton, M., Demuynck, K., Wambacq, P., Cools, R., Compernolle, D.V.: Template-Based Continuous Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing 15(4) (May 2007)
6. Revathi, A., Venkataramani, Y.: Speaker Independent Continuous Speech and Isolated Digit Recognition using VQ and HMM. In: International Conference on Communications and Signal Processing, ICCSP (February 2011)
7. Ahmed, T.N.N., Rao, K.: Discrete Cosine Trasnform. IEEE Transaction on Computers c-24 (January 1974)
8. Jianqin Zhou, P.C.: Generalized Discrete Cosine Transform. In: 2009- Pacific-Asia Conference on Circuits,Communications and System, Chegdu, China (May 2009)
9. Zeng, J., Liu, Z.-Q.: Type-2 Fuzzy Hidden Markov Models and Their Application to Speech Recognition. IEEE Transactions on Fuzzy Systems 14(3) (June 2006)
10. Silva, W.L.S., Serra, G.L.O.: Proposta de Metodologia TCD-Fuzzy para reconhecimentos de Voz. X SBAI – Simposio Brasileiro de Automacao Inteligente. Sao Joao del-Rei - MG - Brasil, pp. 1054–1059 (September 2011)
11. Milner, B.P., Vaseghi, S.V.: Speech Modeling using Cepstral-Time Feature and Hidden Markov Models. In: Proceedings of In. Conference on Acustic Speech and Signal Processing, Adelaide, vol. I, pp. 601–604 (1994)
12. Chen, G.: Discussion of Approximation Properties of Minimum Inference Fuzzy System. In: Proceedings of the 29th Chinese Control Conference, Beijing, China, July 29-31 (2010)
13. Seki, H., Ishii, H., Mizumoto, M.: On the Monotonicity of Fuzzy-Inference Methods Related to T–S Inference Method. IEEE Transactions on Fuzzy Systems 18(3) (June 2010)
14. Zhou, E., Khotand, A.: Fuzzy Classifier Design Genetic Algorithms (2007)
15. Zhang, X., Wang, X., Zhang, S., Yu, F.: Approximating the True Domain of Fuzzy Inference Sentence with Genetic Algorithm. In: Seventh International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010 (2010)

An Approach to Reshaping Clusters for Nearest Neighbor Search

Yong Shi and Brian Graham

Department of Computer Science and Information Systems
Kennesaw State University
1000 Chastain Road
Kennesaw, GA 30144
yshi5@kennesaw.edu

Abstract. In this paper, we present our research on similarity search and clustering problems. Similarity search problems define the distances between data points and a given query point Q, efficiently and effectively selecting data points which are closest to Q. Clustering algorithms separate data points into different groups, in a way that data points in the same group have high similarity and data points from different groups are different from each other. In this paper, we explore the meaning of clusters from a new perspective, and propose an approach to reshape the clusters based on K nearest neighbor search results. The reconstructed clusters can help improve the performance of the following K nearest search process.

Keywords: KNN, Similarity Search, Clustering.

1 Related Work

The similarity search problem has been studied in the last decade, and many algorithms have been proposed for the K nearest neighbor search [6,5,10]. In traditional nearest neighbor problems, the similarity between two data points is based on a similarity function such as Euclidean distance which aggregates the difference between each dimension of the two data points. However, such approaches only focus on full similarities, i.e., the similarity in full data space of the data set. Some approaches [7,1] are proposed targeting partial similarities. However, they have limitations such as the requirement of the fixed subset of dimensions, or fixed number of dimensions as the input parameter(s) for the algorithms.

Clustering algorithms [12,8,3,9] specialize in techniques for grouping similar objects into a cluster in which objects inside a cluster exhibit certain degree of similarities, and separates dissimilar objects into different clusters. It is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. There are various clustering algorithms including partitioning clustering, hierarchical clustering, grid-based clustering, density-based clustering, etc.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 60–67, 2012.

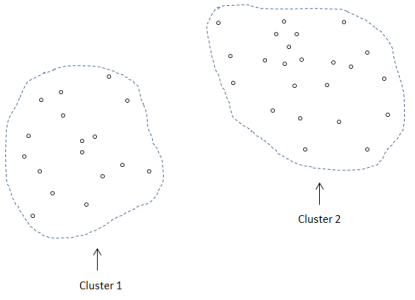


Fig. 1. Two clusters in a two dimensional space

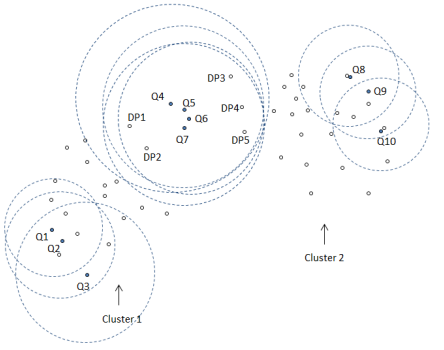


Fig. 2. 10 K Nearest Search results with k as 5

2 Reshape the Clusters Based on the K Nearest Neighbor Search Results

Clustering algorithms can be used as a stand-alone tool to get insight into data distribution, as well as a preprocessing step for other algorithms such as pattern recognition, image analysis, bioinformatics and similarity search. Clustering algorithms can be customized for different applications to improve other algorithms. For example, clustering algorithms can be tailored for K nearest neighboring problems. The new clusters can be called query-based clusters, which group data points not only based on their locations, but also based on various situations in which the data points are the nearest neighbors of certain query points. This kind of clusters is more meaningful in terms of helping improve the performance of the following K nearest neighbor search process.

Figure 1 shows two original clusters in a two-dimensional data space. Cluster 1 contains the data points on the left side, and cluster 2 contains the data points on the right side. Various query points can come in to search for their nearest neighbors. Figure 2 shows an example of a set of 10 query points (Q_1, Q_2, ..., Q_{10}). Suppose the value of K is set as 5. Each query point searches for 5 nearest neighbors in the original data set. From figure 2 we can see the 5 nearest neighbors of Q_1, Q_2 and Q_3 are the data points from cluster 1. The 5 nearest neighbors of Q_8, Q_9 and Q_{10} are the data points from cluster 2. However, the 5 nearest neighbors of Q_4, Q_5, Q_6 and Q_7 are the data points from both cluster 1 (DP_1 and DP_2) and cluster 2 (DP_3, DP_4 and DP_5).

When more and more new query points are in the middle area of this two-dimensional data space, between cluster 1 and cluster 2, more and more K nearest neighbors contain data points from two different clusters. If we readjust the cluster constructions based on this observation, and generate a new cluster based on the frequent appearance of query points in the middle of the data space, the following K nearest search process will be more efficient and effective.

For example, figure 3 shows the reshaping of the clusters. DP_1, DP_2, DP_3, DP_4 and DP_5 are moved out of their original clusters and form a new cluster 3. When a new query point Q comes in, if Q is near the center of cluster 3, we can simply conclude that DP_1, DP_2, DP_3, DP_4 and DP_5 are the 5 nearest neighbors of Q without performing any K nearest search algorithms.

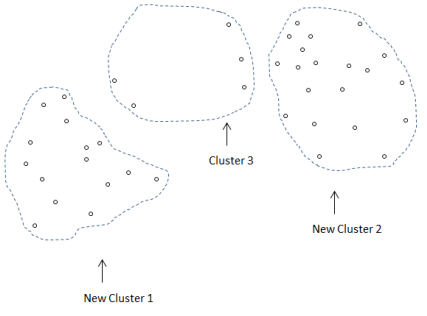


Fig. 3. Three new clusters after adjustment

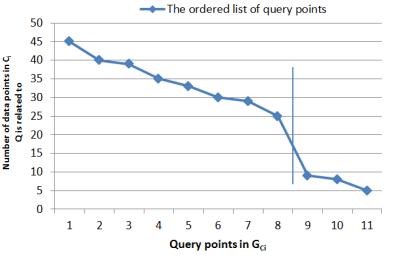


Fig. 4. An example of the ordered list of query points in C_i based on F_{Q,C_i}

3 Problem Definition

Based on the observation of the problems described in the previous section, we propose a novel approach to nearest neighbor search and clustering problems. Before the discussion of our approach we will first introduce a few notations and definitions. Let *n* denote the total number of data points and *d* be the dimensionality of the data space. Let D_l be the *l*th dimension, where l = 1, 2, ..., d. Let the input *d*-dimensional data set DS contain **X**

$$\mathbf{X} = \{X_1, X_2, ..., X_n\}, \tag{1}$$

which is normalized to be within the hypercube $[0,1]^d \subset R^d$. Each data point X_i (i=1,2,...,n) is a *d*-dimensional vector:

$$X_i = [x_{i1}, x_{i2}, ..., x_{id}]. \tag{2}$$

Data point X_i has the id number i. Let m denote the number of query points. Let the *d*-dimensional query set be **Q**

$$\mathbf{Q} = \{Q_1, Q_2, ..., Q_m\} \tag{3}$$

which is also normalized to be within the hypercube $[0,1]^d \subset R^d$. Each query point Q_j (j=1,2,...,m) is a *d*-dimensional vector:

$$Q_j = [q_{j1}, q_{j2}, ..., q_{jd}]. \tag{4}$$

Given a data set DS, we will first generate original clusters. The clustering algorithm is not the focus of our approach, and we can apply our previous work [9] to achieve the original set of clusters. Let the original number of clusters be k_c. Let the set of clusters be $\mathcal{C} = \{C_1, C_2, ..., C_{k_c}\}$.

The K nearest neighboring search algorithm is not the focus of our approach as well, and we can also apply our previous work [10] to find the K nearest neighbors for a query point. For a given query point Q_j (j=1,2,...,m), let KNN_{Q_j} be the set of K data points in $\mathbf{X}$ that are closest to Q_j (K nearest neighbors). Given two data points DP_1 and DP_2, if there exists a query point $Q \in \mathbf{Q}$, so that $DP_1 \in KNN_Q$ AND $DP_2 \in KNN_Q$, we call DP_1 and DP_2 are "CO-KNN'd".

For each data point X_i, we define $S_{X_i} = \{Q | Q \in \mathbf{Q}$, and $X_i \in KNN_Q\}$. S_{X_i} contains all the query points whose K nearest neighbors include X_i. We call the query points in S_{X_i} are related to X_i, and X_i is related to the query points in S_{X_i}.

After m KNN requests, we readjust the clusters according to these KNN results. For each original $C_i \in \mathcal{C}$, we check the data points it contains. For each data point $DP_j \in C_i$, the set S_{DP_j} contains the query points DP_j is related to, as defined in the last paragraph. Let $G_{C_i} = \{Q | Q \in \mathbf{Q}$, and $\exists DP_j \in C_i$ so that $Q \in S_{DP_j}\}$. G_{C_i} contains all the query points that are related to the data points in C_i.

For those query points that are not related to any data points in C_i, they are not in G_{C_i}. For those that are in G_{C_i}, some query points might be related to a lot of data points in C_i, and other query points might be related to just a few data points in C_i. For each query point $Q \in G_{C_i}$, we check how many data points in C_i it is related to, denote it as F_{Q,C_i}. F_{Q,C_i} describes how much Q is related to the data points in C_i. We sort the query points in G_{C_i} in descending order on F_{Q,C_i}, and generate an ordered list L_{C_i} of query points in G_{C_i}.

We choose the query points that have the largest values of F_{Q,C_i}. In order to make the clusters as stable as possible, without dramatic changes, the second half of the ordered query point list L_{C_i} is checked, and the cut spot is set on the first sharp descent query point. Figure 4 shows an example of this case. In Figure 4, the query points are ordered based on F_{Q,C_i}, and the cut spot is set between query point number 8 and query point number 9, where the value of F_{Q,C_i} (on the Y axis) decreases dramatically.

Let $CutSpot_{C_i}$ be the set that contains all the query points before the cut spot in the ordered list L_{C_i}. For each data point $DP_j \in C_i$, there are three cases: 1) DP_j is related to query point(s) in $CutSpot_{C_i}$; 2) DP_j is not related to query point(s) in $CutSpot_{C_i}$, but related to query point(s) in G_{C_i} - $CutSpot_{C_i}$; 3) DP_j is not related to any query points in G_{C_i}.

The data points in case 2) are related to query points in G_{C_i} - $CutSpot_{C_i}$. Those query points are not only related to data points in C_i, otherwise, they would be in G_{C_i}. Thus, the data points in case 2) are CO-KNN'd with data points from other clusters.

Our goal is to find data points that are CO-KNN'd to other data points from different clusters and generate new clusters by merging them. To achieve this goal, we define a new set H, and for every $C_i \in \mathcal{C}$, we include the data points in case 2) in H.

After checking every cluster in $\mathcal{C}$, we have the data points of case 2) from all the clusters, and put them into H. For data points DP_1 and DP_2 in H, if they are CO-KNN'd a lot for different query points, they should be in the same new cluster.

To generate new clusters from the data points in H, we create a graph G. Each data point in H is a vertex in G. For two vertices V_1 and V_2 in G, there is an edge between V_1 and V_2 if and only if the two corresponding data points are CO-KNN'd for more than P times. We find all the complete graphes in G. Each complete graph is a new cluster. For those data points in H that do not belong to any new clusters, we move them back to their original clusters.

4 Algorithm

In this section we present the RESHAPE algorithm to adjust the clusters tailored for K nearest neighbor search process.

Figure 5 presents the RESHAPE algorithm. Given a data set DS, we first generate the original cluster set. For each query point, we search for its K nearest neighbors. After we perform the K nearest neighbor search for m times, we generate S_{X_i} for each $X_i \in \mathbf{X}$, and set H as $\emptyset$. Next, we collect the query points related to each cluster in $\mathcal{C}$, calculate F_{Q,C_i} for each query point Q involved, and sort them accordingly. The query points with high F_{Q,C_i} are included in $CutSpot_{C_i}$. Then we check each data point to see if it is CO-KNN'd with data points from other different clusters. We select those data points from different clusters which are CO-KNN'd with each other for many query points, and generate new clusters for them. The last step is to move those data points that do not belong to any new clusters back to their original clusters. Thus a new set of clusters are ready for the following K nearest neighbor search process, including the reconstructed original clusters and the new generated clusters.

Based on the algorithm description we can see that, in the previous example shown in figure 2, DP_1, DP_2, DP_3, DP_4 and DP_5 are data points that are CO-KNN'd with each other for Q_4, Q_5, Q_6 and Q_7, and a new cluster 3 is generated for them in figure 3.

Suppose the size of the data set is n. Throughout the process, we need to keep track of the information of all data points and all query points, which collectively occupies $O(n + m)$ space. For our algorithm, for each cluster $C_i \in \mathcal{C}$, we need to sort query points that are related to the data points in C_i based on F_{Q,C_i}. The time required is $O(k_c m log m)$. The value of P and m varies in different applications. Based on our testing results on these parameter values, we set P as 3 and m as 10.

5 Experiments

In this section we present the experimental results on both synthetic and real data sets to demonstrate the effectiveness and efficiency of our algorithm. Our experiments were run on Intel(R) Pentium(R) 4 with CPU of 3.39GHz and Ram of 0.99 GB.

5.1 Experiments on High-Dimensional Data Sets

We first design a synthetic data generator to produce data sets with normalized distributions in order to test the scalability of our algorithm over data size n and dimensionality

Algorithm RESHAPE (***DS: data set, Q: query point set, D: dimensions, K: number of data points required as nearest neighbors***)

Begin

1) Generate the original cluster set: $\mathcal{C} = \{C_1, C_2, ..., C_{k_c}\}$;

2) For each query point $Q_j \in \mathbf{Q}$, find KNN_{Q_j};

3) For each $X_i \in \mathbf{X}$, generate $S_{X_i} = \{Q | Q \in \mathbf{Q}$, and $X_i \in KNN_Q\}$;

4) Set H = $\emptyset$;

5) For each $C_i \in \mathcal{C}$:

a) Generate $G_{C_i} = \{Q | Q \in \mathbf{Q}$, and $\exists DP_j \in C_i$ so that $Q \in S_{DP_j}\}$;

b) For each $Q \in G_{C_i}$, calculate F_{Q,C_i};

c) Sort the query points in G_{C_i} in descending order based on F_{Q,C_i}, and generate an ordered list L_{C_i};

d) Generate $CutSpot_{C_i}$ to contain the query points that are most related to the data points in C_i;

e) For each data point $DP_j \in C_i$, determine which of the three cases it belongs to;

f) H = H $\bigcup$ $\{DP_j$—$DP_j \in C_i$, and $DP_j \in case2\}$;

6) Set up a graph G based on H, in which each vertex V represents a data point DP ∈ H. For V_1 and V_2 in G, generate an edge $E(V_1, V_2)$ to connect V_1 and V_2 if, in H, the corresponding data points of V_1 and V_2 are CO-KNN'd for more than P time;

7) Find complete graphs in G, and each complete graph is a new cluster. Let $\mathcal{C}_{new}$ be the set of new clusters

8) For each DP ∈ H, if its corresponding vertex in G does not belong to any complete graphs in G, return it to its original cluster;

9) Return $\mathcal{C} \bigcup \mathcal{C}_{new}$.

End.

Fig. 5. Proc: Algorithm RESHAPE

d. The sizes of the data sets vary from 10,000, 15,000... to 25,000, with the gap of 5,000 between each two adjacent data set sizes. The dimensions of the data sets vary from 10, 15, ... to 35, with the gap of 5 between each two adjacent numbers of dimensions. The value of K is set as 5.

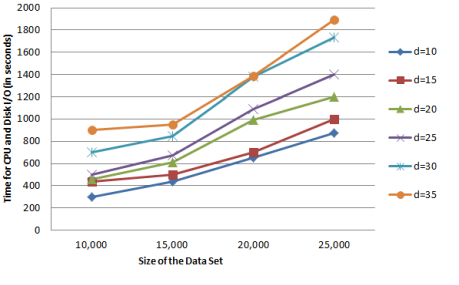


Fig. 6. Running time on a query point set with increasing data set sizes (K = 5)

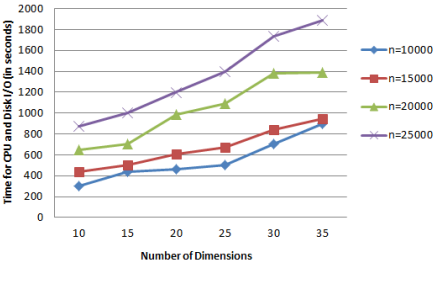


Fig. 7. Running time on a query point set with increasing dimensions (K = 5)

Figure 6 shows the running time of various data sets on a query set of size 10 with data size increasing from 10,000 to 25,000. Each group has fixed dimensionality (from 10, 15, ... to 35).

Figure 7 shows the running time of various data sets on a query set of size 10 with dimensionality increasing from 10 to 35. Each group has fixed data size (from 10,000, 15,000, ... to 25,000).

Figure 6 and figure 7 indicate that our algorithm is scalable over data size and dimensionality.

5.2 Experiments on Real Data Set

We next evaluate the effectiveness of our proposed approach, RESHAPE, for finding nearest neighbors.

We obtained the real data sets from UCI Machine Learning Repository [4]. The first data set is ionosphere data set which contains 351 data points with 34 dimensions. There are two classes in the ionosphere data: *g* as *good*, and *b* as *bad*. The second data set is glass data set. It contains 214 data points with 9 dimensions. There are 7 classes in the glass data, class 1 to class 7. The third data set is iris data set. It contains 150 data points with 4 dimensions. There are 3 classes in the iris data: Iris-setosa, Iris-versicolor, and Iris-virginica.

We compare the testing result of these data sets with other algorithms such as Frequent K-n-match algorithm [11] and IGrid[2]. Data points are selected randomly from each real data set as the initial query points. In our experiment, we set the value of K as 5, and the size of the query point set as 10. For each real data set, we perform our algorithm to find the K nearest neighbors for each query point. If a retrieved data point (a neighbor of a given query point) has the same class with the query point it is related to, we call it a successful retrieval. Otherwise, we call the data point a unsuccessful retrieval. We calculate how many successful retrievals we have among the results from performing RESHAPE on these query points, and evaluate the accuracy rate. The average accuracy rate of RESHAPE algorithm is 92.3%, which is higher than the accuracy rate of IGrid (89.9%), and that of Freq. K-n-match algorithm, which is 91.8%.

6 Conclusion and Discussion

In this paper, we present a novel approach to reshape the clusters tailored for the K nearest neighbor search process. We generate new clusters by merging data points from different clusters if they belong to the same K nearest neighbors of many query points. The reconstructed clusters can improve the effectiveness and efficiency for processing the new coming K nearest neighbor search requests.

References

1. Aggarwal, C.C.: Towards meaningful high-dimensional nearest neighbor search by human-computer interaction. In: ICDE (2002)
2. Aggarwal, C.C., Yu, P.S.: The IGrid index: reversing the dimensionality curse for similarity indexing in high dimensional space. In: Knowledge Discovery and Data Mining, pp. 119–129 (2000)

3. Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J.: OPTICS: Ordering Points To Identify the Clustering Structure. In: Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD 1999), Philadelphia, PA, pp. 49–60 (1999)
4. Bay, S.D.: The UCI KDD Archive. Department of Information and Computer Science. University of California, Irvine, http://kdd.ics.uci.edu
5. Fagin, R., Kumar, R., Sivakumar, D.: Efficient similarity search and classification via rank aggregation (2003)
6. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. The VLDB Journal, 518–529 (1999)
7. Hinneburg, A., Aggarwal, C.C., Keim, D.A.: What is the nearest neighbor in high dimensional spaces? The VLDB Journal, 506–515 (2000)
8. Sheikholeslami, G., Chatterjee, S., Zhang, A.: Wavecluster: A multi-resolution clustering approach for very large spatial databases. In: Proceedings of the 24th International Conference on Very Large Data Bases (1998)
9. Shi, Y., Song, Y., Zhang, A.: A shrinking-based clustering approach for multidimensional data. IEEE Transactions on Knowledge and Data Engineering 17, 1389–1403 (2005)
10. Shi, Y., Zhang, L.: Panknn: A dimension-wise approach to similarity search problems. In: DMIN, pp. 555–561 (2008)
11. Tung, A.K.H., Zhang, R., Koudas, N., Ooi, B.C.: Similarity search: a matching based approach. In: VLDB 2006, pp. 631–642. VLDB Endowment (2006)
12. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Canada, pp. 103–114 (1996)

Interestingness Measures for Fixed Consequent Rules

Jon Hills*, Luke M. Davis, and Anthony Bagnall

School of Computing Sciences, University of East Anglia,
Norwich, NR4 7TJ, United Kingdom
{j.hills,luke.davis,anthony.bagnall}@uea.ac.uk
http://www.uea.ac.uk

Abstract. Many different rule interestingness measures have been proposed in the literature; we show that, under two assumptions, at least twelve of these measures are proportional to Confidence. We consider rules with a fixed consequent, generated from a fixed data set. From these assumptions, we prove that Satisfaction, Ohsaki's Conviction, Added Value, Brin's Interest/Lift/Strength, Brin's Conviction, Certainty Factor/Loevinger, Mutual Information, Interestingness, Sebag-Schonauer, Ganascia Index, Odd Multiplier, and Example/counter-example Rate are all monotonic with respect to Confidence. Hence, for ordering sets of partial classification rules with a fixed consequent, the Confidence measure is equivalent to any of the twelve other measures.

Keywords: Interestingness, Rules, Confidence, Partial Classification.

1 Introduction

Rule induction involves generating rules from data sets; rules can be used for both predictive and descriptive data mining tasks, and are comprehensible to non-experts (see, for example, [1, 2]). We focus on the task of *partial classification* [3]. Partial classification differs from classification because it need not cover the entire attribute and class space.

Rule induction algorithms (e.g. [1]) can be used to generate a set of partial classification rules. Partial classification rules can provide a comprehensible set of predictors for certain outcomes. We consider partial classification rules with a single, fixed consequent, i.e. rules for a single class.

Sets of partial classification rules can be very large; hence, some method for evaluating and selecting rules is required. Many measures have been proposed in the literature. Collectively, they are referred to as *rule interestingness measures*. We show that, under the assumptions of a fixed consequent and fixed data set, twelve different measures are all proportional to the standard measure of rule quality, *Confidence* [1]. We conclude that, when assessing partial classification rule sets with a fixed consequent, the optimal rules in terms of Confidence are

* This work was supported in part by the UEA Annual Alumni Fund.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 68–75, 2012.

the optimal rules by any of the measures analysed. Hence, we do not need the other twelve measures if we know the optimal rules in terms of Confidence.

The rest of the paper is structured as follows. Section two contains background, definitions, and notation. Section three contains proofs that the twelve measures are all proportional to confidence. In section four, we empirically demonstrate our findings, and in section five draw conclusions and consider future work.

2 Background

2.1 Rules

Rules take the form:

$$Antecedent \Rightarrow Consequent$$

where both the antecedent (A) and the consequent (C) of the rule are a set of conditions. For notation, we use $N(A)$ to indicate the number of records in the data set that satisfy the antecedent of the rule, and $N(C)$ for the number of records that satisfy the consequent of the rule. $N(U)$ is used to represent the size of the data set. The propositional connectives $\wedge$, $\vee$, and $\neg$ are used to indicate conjunction, disjunction, and negation. For example, $N(A \vee \neg C)$ represents the number of records that satisfy the antecedent or the negation of the consequent.

Various measures of interestingness can be used to assess rules, so that better rules can be retained and worse rules discarded. Large numbers of interestingness measures have been proposed in the literature, see, for example, [4–6].

2.2 Partial Classification with Fixed Consequent Rules

We focus on partial classification rules with a fixed consequent. Fixed consequent partial classification rules are appropriate for many data mining tasks, as we are often only concerned with predicting or describing a single class of interest. Applications include failure detection and diagnosis (e.g. [3]), prediction of medical conditions (e.g. [7]), and product marketing (e.g. [1]). In contrast to partial classification rules, classification rules do not have a fixed consequent, as they cover all classes.

In general, a single, unchanging set of data is used in the induction of a rule set; the interestingness of the rules makes sense only in relation to this data set. It is reasonable to assume, therefore, that the data set is fixed.

For the proofs in section three, we make the following assumptions:

1. The data set is fixed. That is, $N(U)$ is a constant. We do not compare rules across data sets.
2. The consequent of the rule is fixed. That is, $N(C)$ is a constant, as is $N(\neg C)$.

These assumptions are plausible for many common data mining tasks. In the next section, we show how twelve different interestingness measures are proportional to Confidence given our assumptions.

3 Interestingness Measures

In this section, we prove theoretically that twelve interestingness measures proposed in the literature are monotonic with respect to Confidence, granting the assumptions that the consequent and the data set are fixed. The Confidence of a rule is defined as:

$$Confidence = \frac{N(A \wedge C)}{N(A)}$$

Confidence is a standard measure of the quality of a rule, used throughout the literature (see, for example, [1, 8, 9]).

Several of the proofs rely on the following equivalence:

$$\frac{N(A \wedge \neg C)}{N(A)} = 1 - \frac{N(A \wedge C)}{N(A)} = 1 - Confidence$$

3.1 Satisfaction

The formula for *Satisfaction* [10] is:

$$\frac{N(\neg C) \times N(A) - N(A \wedge \neg C) \times N(U)}{N(C) \times N(A)}$$

The formula can be rearranged to give:

$$\frac{N(\neg C)}{N(C)} - \frac{N(A \wedge \neg C)}{N(A)} \times \frac{N(U)}{N(C)}$$

(cancelling $N(A)$ in the first term). The first and third terms are constants, and the second term is equal to $1 - Confidence$, so satisfaction is proportional to Confidence.

3.2 Ohsaki's Conviction

Ohsaki's Conviction [11] is calculated as:

$$\frac{N(A) \times N(\neg C)^2}{N(A \wedge \neg C) \times N(U)^2}$$

The formula can be rearranged as:

$$\frac{N(A)}{N(A \wedge \neg C)} \times \frac{N(\neg C)^2}{N(U)^2}$$

By dividing both the numerator and denominator of the first term by $N(A)$, we have:

$$1/\frac{N(A \wedge \neg C)}{N(A)}$$

which is equal to $1/(1 - Confidence)$; the second term is a constant. Hence, Ohsaki's conviction is proportional to Confidence.

3.3 Added Value

The formula for the *Added Value* measure [12] is:

$$\frac{N(A \wedge C)}{N(A)} - \frac{N(C)}{N(U)}$$

Since the second term is a constant, this measure is proportional to Confidence.

3.4 Brin's Interest/Lift/Strength

Brin's Interest/Lift/Strength [13–15] is calculated as:

$$\frac{N(A \wedge C)}{N(A)} \times \frac{N(U)}{N(C)}$$

The second term is fixed, so this measure is proportional to Confidence.

3.5 Brin's Conviction

Brin's Conviction [13] is:

$$\frac{N(A) \times N(\neg C)}{N(U) \times N(A \wedge \neg C)}$$

As shown in [8], the formula can be rearranged by dividing both the numerator and the denominator by $N(A)$, giving

$$\frac{N(\neg C)}{N(U) \times N(A \wedge \neg C)/N(A)}$$

$\frac{N(A \wedge \neg C)}{N(A)}$ is equal to $1 - Confidence$. $N(U)$ and $N(\neg C)$ are fixed, so Brin's Conviction is monotonic with respect to Confidence.

3.6 Certainty Factor/Loevinger

The formula for *Certainty Factor/Loevinger* [4, 16] is:

$$\left(\frac{N(A \wedge C)}{N(A)} \times \frac{N(U)}{N(\neg C)}\right) - \frac{N(C)}{N(\neg C)}$$

Both $\frac{N(U)}{N(\neg C)}$ and $\frac{N(C)}{N(\neg C)}$ are constants, so Certainty Factor/Loevinger is proportional to Confidence.

3.7 Mutual Information

The *Mutual Information* measure [4] is:

$$log_2\left(\frac{N(A \wedge C)}{N(A)} \times \frac{N(U)}{N(C)}\right)$$

The log_2 function is monotonic, and $\frac{N(U)}{N(C)}$ is a constant, so mutual information is proportional to Confidence.

3.8 Interestingness

Interestingness [17] is calculated as follows:

$$\frac{N(A \wedge C)}{N(A)} \times log_2 \left(\frac{N(A \wedge C)}{N(A)} \times \frac{N(U)}{N(C)} \right)$$

Interestingness is Confidence multiplied by the Mutual Information measure. Hence, it is monotonic with respect to Confidence.

3.9 Sebag-Schonauer

The *Sebag-Schonauer* measure [18] is:

$$\frac{N(A \wedge C)}{N(A \wedge \neg C)}$$

By dividing both the numerator and the denominator by $N(A)$, we see that the formula is equivalent to $Confidence/(1 - Confidence)$, which is proportional to Confidence. This measure is proportional to Confidence even if we relax the assumption that the consequent is fixed.

3.10 Ganascia Index

The *Ganascia index* [19] for a rule is:

$$\frac{N(A \wedge C) - N(A \wedge \neg C)}{N(A)}$$

$N(A \wedge \neg C) = N(A) - N(A \wedge C)$, so the numerator is equal to:

$$N(A \wedge C) - N(A) + N(A \wedge C) = 2N(A \wedge C) - N(A)$$

Hence, the formula can be rearranged as $2\frac{N(A \wedge C)}{N(A)} - \frac{N(A)}{N(A)}$, which is proportional to Confidence. Like Sebag-Schonauer, Ganascia Index is proportional to Confidence even if we relax our assumption of fixed consequent.

3.11 Odd Multiplier

Odd Multiplier [20] is calculated as:

$$\frac{N(A \wedge C) \times N(\neg C)}{N(C) \times N(A \wedge \neg C)}$$

We rearrange the formula as:

$$\frac{N(A \wedge C)}{N(A \wedge \neg C)} \times \frac{N(\neg C)}{N(C)}$$

The second term is a constant, and the first term is the Sebag-Shonauer measure, which is proportional to Confidence. Hence, odd multiplier is proportional to Confidence.

3.12 Example/Counter-Example Rate

We calculate the *Example/counter-example Rate* [21] as:

$$\frac{N(A \wedge C) - N(A \wedge \neg C)}{N(A \wedge C)}$$

The formula can be rearranged as:

$$\frac{N(A \wedge C)}{N(A \wedge C)} - \frac{N(A \wedge \neg C)}{N(A \wedge C)}$$

By dividing both the numerator and the denominator of the second term by $N(A)$, and cancelling $N(A \wedge C)$ in the first term, we have $1 - \frac{1-Confidence}{Confidence}$, which is proportional to Confidence. This measure is proportional to Confidence without the assumption that the consequent is fixed.

4 Empirical Demonstration

To demonstrate our findings, we generate a rule set from the *Adult* data set [22] using *All Rules Algorithm* [23]. We remove all records with missing data, and the entire field *fnlwgt*. We fix the consequent as $\{Salary =$ '>50K'$\}$. For this rule set, $N(U) = 30,162$ and $N(C) = 7,508$. Table 1 lists seven rules from the rule set. We take it that this sample illustrates the general point that the ordering of the rules does not vary over the thirteen measures. Table 2 shows the value of each interestingness measure for each rule.

Table 1. Seven rules with varying levels of confidence, drawn from a rule set generated by All Rules Algorithm operating on the Adult data set

Rule	$N(A \wedge C)$	$N(A)$
(1) Cap_gain $\geq$ 4687 & Age $\geq$ 21 & Edu_yrs $\geq$ 2	1451	1612
(2) Edu_yrs $\geq$ 13 & Hours_per_week $\geq$ 42 & Relationship = 'Husband'	1,580	1,995
(3) Edu_yrs $\geq$ 13 & Hours_per_week $\geq$ 41 & Relationship = 'Husband'	1,581	1,999
(4) Edu_yrs $\geq$ 11 & Mar_status = 'Married-civ-spouse' & Age $\geq$ 31	3,295	4,673
(5) Edu_yrs $\geq$ 11 & Mar_status = 'Married-civ-spouse' & Age $\geq$ 30	3,376	4,812
(6) Mar_status = 'Married-civ-spouse' & Edu_yrs $\geq$ 10 & Age $\geq$ 23	4,752	7,915
(7) Mar_status = 'Married-civ-spouse' & Age $\geq$ 24 & Edu_yrs $\geq$ 8	6,229	12,434

Table 2. Values for each interestingness measure of each of the seven rules. The ordering is the same for all thirteen measures.

Measure	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Confidence	0.900	0.792	0.791	0.705	0.702	0.600	0.500
Satisfaction	2.616	2.182	2.177	1.833	1.819	1.412	1.013
Ohsaki's Conviction	5.648	2.712	2.698	1.913	1.890	1.412	1.130
Added Value	0.651	0.543	0.542	0.456	0.453	0.351	0.252
Interest/Lift/Strength	3.616	3.182	3.177	2.833	2.818	2.412	2.013
Brin's Conviction	7.520	3.611	3.592	2.547	2.517	1.879	1.505
Certainty Factor/Loevinger	0.867	0.723	0.722	0.607	0.603	0.468	0.336
Mutual Information	1.854	1.670	1.668	1.502	1.495	1.270	1.009
Interestingness	1.669	1.322	1.319	1.059	1.049	0.763	0.505
Sebag-Schonauer	9.012	3.807	3.782	2.391	2.351	1.502	1.004
Ganascia Index	0.800	0.584	0.582	0.410	0.403	0.201	0.002
Odd Multiplier	27.193	11.488	11.412	7.215	7.094	4.533	3.029
Example/counter-example Rate	0.889	0.737	0.736	0.582	0.575	0.334	0.004

5 Conclusions

Numerous measures exist to evaluate the interestingness of rules; choosing the appropriate measure can be a difficult part of the data mining process. We have shown that twelve interestingness measures are, under the assumption of fixed consequent and fixed dataset, proportional to Confidence. Hence, we conclude that, at least for fixed consequent rules, the optimal rules in terms of Confidence are the optimal rules in terms of any of the twelve measures analysed. This result should simplify the data mining process by making it easier to select an appropriate interestingness measure for rule sets that satisfy our assumptions.

There are many interestingness measures that are not proportional to Confidence, e.g. *Negative Reliability* [10] and *Gini Index* [4]. In future work, we will prove that, under our assumptions, any measure composed of only the values $N(U)$, $N(C)$, $N(A \wedge C)$, and $N(A)$ belongs to one of a small number of classes. The measures in each class are proportional to each other; as such, we need only one measure from a class to find the optimal rules by each measure in that class.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record, vol. 22, pp. 207–216. ACM (1993)
2. Clark, P., Boswell, R.: Rule Induction with CN2: Some Recent Improvements. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 151–163. Springer, Heidelberg (1991)
3. Ali, K., Manganaris, S., Srikant, R.: Partial classification using association rules. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, pp. 115–118 (1997)
4. Tan, P., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 32–41. ACM (2002)

5. Carvalho, D.R., Freitas, A.A., Ebecken, N.F.F.: Evaluating the Correlation Between Objective Rule Interestingness Measures and Real Human Interest. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 453–461. Springer, Heidelberg (2005)
6. Ohsaki, M., Abe, H., Tsumoto, S., Yokoi, H., Yamaguchi, T.: Evaluation of rule interestingness measures in medical knowledge discovery in databases. Artificial Intelligence in Medicine 41(3), 177–196 (2007)
7. Breault, J., Goodall, C., Fos, P.: Data mining a diabetic data warehouse. Artificial Intelligence in Medicine 26(1-2), 37–54 (2002)
8. Bayardo Jr., R., Agrawal, R.: Mining the most interesting rules. In: Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 145–154. ACM (1999)
9. Balcázar, J.: Confidence width: An objective measure for association rule novelty. In: Workshop on Quality Issues, Measures of Interestingness and Evaluation of Data Mining Models QIMIE, vol. 9
10. Lavrač, N., Flach, P.A., Zupan, B.: Rule Evaluation Measures: A Unifying View. In: Džeroski, S., Flach, P.A. (eds.) ILP 1999. LNCS (LNAI), vol. 1634, pp. 174–185. Springer, Heidelberg (1999)
11. Ohsaki, M., Kitaguchi, S., Okamoto, K., Yokoi, H., Yamaguchi, T.: Evaluation of Rule Interestingness Measures with a Clinical Dataset on Hepatitis. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 362–373. Springer, Heidelberg (2004)
12. Yao, Y.Y., Zhong, N.: An Analysis of Quantitative Measures Associated with Rules. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS (LNAI), vol. 1574, pp. 479–488. Springer, Heidelberg (1999)
13. Brin, S., Motwani, R., Ullman, J., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. In: ACM SIGMOD Record, vol. 26, pp. 255–264. ACM (1997)
14. Bayardo, R., Agrawal, R., Gunopulos, D.: Constraint-based rule mining in large, dense databases. Data Mining and Knowledge Discovery 4(2), 217–240 (2000)
15. Dhar, V., Tuzhilin, A.: Abstract-driven pattern discovery in databases. IEEE Transactions on Knowledge and Data Engineering, 926–938 (1993)
16. Lenca, P., Meyer, P., Vaillant, B., Lallich, S.: On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. European Journal of Operational Research 184(2), 610–626 (2008)
17. Yao, J., Liu, H.: Searching multiple databases for interesting complexes. In: KDD: Techniques and Applications, vol. 482, pp. 484–485. World Scientific, Singapore (1997)
18. Sebag, M., Schoenauer, M.: Generation of rules with certainty and confidence factors from incomplete and incoherent learning bases. In: Proc. of EKAW, vol. 88 (1988)
19. Ganascia, J.: Deriving the learning bias from rule properties. In: Machine Intelligence, vol. 12, pp. 151–167. Clarendon Press (1991)
20. Geng, L., Hamilton, H.: Interestingness measures for data mining: A survey. ACM Computing Surveys (CSUR) 38(3), 9 (2006)
21. Huynh, X., Guillet, F., Briand, H.: Evaluating interestingness measures with linear correlation graph. Advances in Applied Artificial Intelligence, 312–321 (2006)
22. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets.html
23. Richards, G., Rayward-Smith, V.: The discovery of association rules from tabular databases comprising nominal and ordinal attributes. Intelligent Data Analysis 9(3), 289–307 (2005)

Abnormal Event Detection via Multi-Instance Dictionary Learning

Jing Huo[1], Yang Gao[1], Wanqi Yang[1], and Hujun Yin[2]

[1] State Key Laboratory for Novel Software Technology,
Nanjing University, China
[2] School of Electrical and Electronic Engineering,
The University of Manchester, UK
gaoy@nju.edu.cn

Abstract. In this paper, we present a method for detecting abnormal events in videos. In the proposed method, we define an event containing several sub-events. Sub-events can be viewed as instances and an event as a bag of instances in the multi-instance learning formulation. Given labeled events but with the labels of sub-events unknown, the proposed method is able to learn a dictionary together with a classification function. The dictionary is capable of generating discriminant sparse codes of sub-events while the classification function is able to classify an event. This method is suited for scenarios where the label of a sub-event is ambiguous, while the label of a set of sub-events is definite and is easy to obtain. Once the sparse codes of sub-events are generated, the classification of an event is carried out according to the result given by the classification function. An efficient optimization procedure of the proposed method is presented. Experiments show that the method is able to detect abnormal events with comparable or improved accuracy compared with other methods.

Keywords: Multi-instance learning, Sparse coding, Abnormal event detection.

1 Introduction

Detection of abnormal events in videos has been of great interest to the computer vision community in recent years, due to the wide adoption of video surveillance systems in public places. However, abnormal events detection in videos is still a challenging task in the literature. Many methods have been proposed, varying from supervised learning [1], unsupervised learning [2], semisupervised learning [3], to active learning [4]. In supervised learning, labeled events are given or obtained in the training stage. As it is known that labeled training samples are usually hard to obtain, and in many situations, ambiguous events are hard to label. This is mainly because that normal and abnormal events are sometimes visually similar or share some common features. Contrary to the supervised methods, no prior information is needed in unsupervised learning. However, such methods suffer from concept drift. Although many incremental methods have been

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 76–83, 2012.

proposed to deal with the problem, it is often impossible to solve the problem completely. Therefore most unsupervised methods use only normal events to train a model, and in the later stage, those events that are hard to be represented by the model are identified as abnormal. This kind of methods is also referred to as the one-class unsupervised methods in the literature [5].

Multi-instance learning [6] can be viewed as a variation of supervised learning. In multi-instance learning, a training set contains several labeled bags, each consists of some instances of which the labels are unknown. The task is to predict the label of a bag. Particularly, if a bag contains at least one positive instance, then the bag is labeled as positive. In our framework, a bag is defined as to contain several sub-events. Abnormal events are regarded as positive instances. If there is at least one abnormal event in a bag, then the bag is positive or is an abnormal bag. In video processing, the label of a bag is usually easier to obtain than an event. Therefore, unlike previous work that defines abnormal events as those that are rare and dissimilar from regular event patterns, in this framework, abnormal event is defined as an abnormal bag that contains at least one abnormal instance, and the instances in a bag are defined as sub-events.

Sparse coding has been shown effective in signal reconstruction and classification. Previous work [7, 8] has used reconstructive sparse codes to distinguish abnormal events from normal events. But for the task of classification, it would be better to use discriminative sparse codes. We herein propose a method called Multi-Instance Dictionary Learning (MIDL), which is able to generate discriminative sparse codes of instances using labeled bags.

The rest of this paper is organized as follows. Section 2 gives a brief overview of related work. Section 3 provides a detailed explanation of our method, followed by experimental results and an analysis in Section 4. Section 5 concludes.

2 Related Work

There has been a lot of work in abnormal event detection. The related methods can be categorized into two categories, trajectory based and motion feature based. The trajectory based methods rely mainly on the tracking of an object [9–11]. However, reliable tracking is still a challenging task. Besides, in many crowed scenes, tracking of an object can be unrealistic. Therefore, there is also much effort devoted to motion feature based methods. In these methods, features such as optical flow, motion history, gradients are extracted at pixel level. Then different models are built to learn the spatial-temporal relations between different feature patterns. These models include the Markov Random Field [12], Gaussian Mixture Model [13, 14] and Social Force Model [15]. Such methods avoid explicitly tracking of moving objects and therefore are suited for detecting abnormal events in crowded scenes.

3 Multi-Instance Dictionary Learning for Abnormal Event Detection

3.1 Event Representation

The proposed framework adopts the feature representation method based on Multi-scale Histogram of Optical Flow (MHOF), which is also used in [8]. But unlike their scheme, we only extract MHOF in cells that contain moving pixels. The feature extraction method can be summarized as following steps. (1) Given a video, moving pixels are first detected using frame differencing method. Partition the video view into small over-lapping cells, and the moving pixels will fall into different cells. (2) For those cells that contain moving pixels, MHOF is extracted. (3) For several fames in a succession, the MHOF features are considered instances (i.e. sub-events). All the features jointly form the concept of a bag (i.e. event) in the multi-instance learning framework. The flowchart of the feature extraction process is shown in Fig.1.

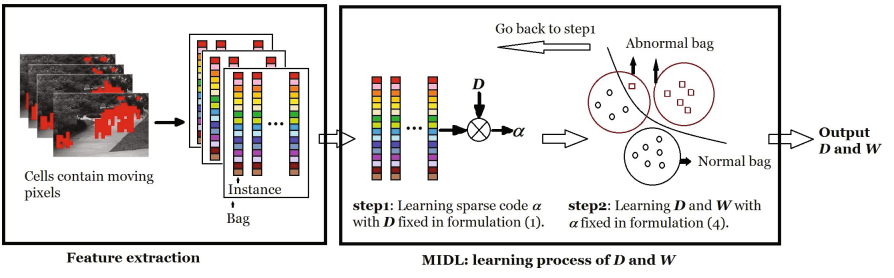


Fig. 1. Flowchart of the feature extraction process. Given a video for training with the label of bags known, feature extraction is firstly performed, as shown in the left side of the figure. The second stage is the learning process of MIDL, $\boldsymbol{D}$ and $\boldsymbol{W}$ are initialized using unsupervised dictionary learning method, and then solved in an iterative manner. In step2, small circles are the sparse code of normal sub-events and rectangles are the sparse code of abnormal sub-events.

3.2 Multi-Instance Dictionary Learning (MIDL)

General Sparse Coding Formulation. The basic formulation of sparse coding is to learn a dictionary given a set of training samples. Then an input sample can be represented as the linear combination of the basis in the dictionary with the coefficient being sparse. This is so-called sparse representation. Here, assume that we are given a learned dictionary $\boldsymbol{D}$, $\boldsymbol{D} \in \mathbb{R}^{m \times k}$ (The learned dictionaries can be over complete with $k > m$) and a sub-event represented by $\boldsymbol{x}$, $\boldsymbol{x} \in \mathbb{R}^m$, the sparse representation of $\boldsymbol{x}$ could be represented as $\boldsymbol{\alpha}^*$, $\boldsymbol{\alpha}^* \in \mathbb{R}^k$, where

$$\boldsymbol{\alpha}^*(\boldsymbol{x}, \boldsymbol{D}) \triangleq \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^p} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}\|_1 \tag{1}$$

where the first term is the sparse reconstruction error; the second term is the sparsity regularization, which guarantees that the coefficient is sparse.

This is the general sparse coding formulation and is able to generate sparse reconstructive codes. Next we present a multi-instance dictionary learning method that is able to generate discriminative sparse codes.

MIDL: The Cost Function. Assume that there is a training set of several labeled bags $\{(B_1, Y_1), ..., (B_n, Y_n)\}$, where $B_i = \{\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, ..., \boldsymbol{x}_{in_i}\}$ is the ith bag containing n_i instances, $\boldsymbol{x}_{ij}$ is the jth instance in bag B_i, and also the jth sub-event in bag B_i in our formulation. Y_i is the label of the ith bag. Inspired by [16], we consider that there exists a function l, which is able to classify the sparse code of an instance as either positive or negative. The function can be formulated as below:

$$l(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{W}) = \boldsymbol{x}^T \boldsymbol{W} \boldsymbol{\alpha} + b \tag{2}$$

where $\boldsymbol{W} \in \mathbb{R}^{m \times k}$ is the classification parameter and $b \in \mathbb{R}$. In the multi-instance learning sense, the label of a bag can be interpreted as follows. If $Y_i = -1$, then $y_{ij} = -1$, for all $j = 1, ...n_i$. On the other hand, if $Y_i = 1$, then there is at least one instance in the bag that is positive. Therefore, the classification function for bag B_i could be written as $F(B_i) = \max_{j=1,...n_i} l(\boldsymbol{x}_{ij}, \boldsymbol{\alpha}, \boldsymbol{W})$. If we only care about the sign in classification, we can represent the label of a bag B_i as:

$$Y_i = sign(F(B_i)) \tag{3}$$

Given a labeled bag, the cost function can be represented as $C(Y_i, B_i)$; and in our case, C is chosen as a logistic cost function, $C(Y_i, B_i) = log(1 + e^{-Y_i F(B_i)})$.

In summary, our goal is to minimize the formulation below:

$$\min_{\boldsymbol{D}, \boldsymbol{W}} f(\boldsymbol{D}, \boldsymbol{W}) + \frac{\nu}{2} \|\boldsymbol{W}\|_F^2 \tag{4}$$

where f has the form $f(\boldsymbol{D}, \boldsymbol{W}) = \frac{1}{n} \sum_{i=1}^{n} C(Y_i, B_i)$.

Optimization. The optimization procedure of formulations (1) and (4) is given below. It is solved in an iterative manner as shown in Fig.1.

Learning sparse code $\boldsymbol{\alpha}$ with $\boldsymbol{D}$ fixed in formulation (1). With $\boldsymbol{D}$ fixed, the optimization problem of $\boldsymbol{\alpha}$ in formulation (1) is a L1-regularized least squares problem. There are many methods to solve the problem: interior point [17], a modification of least angle regression (LARS) [18] and feature sign search method [19]. Here we adopt the modified LARS method.

Learning $\boldsymbol{D}$ and $\boldsymbol{W}$ in formulation (4). As it has been proposed in [20], the following formulation (5) is differentiable with respect to $\boldsymbol{D}$ and $\boldsymbol{W}$ given three assumptions and $\lambda_2 > 0$.

$$\begin{cases} \min_{\boldsymbol{D},\boldsymbol{W}} f(\boldsymbol{D},\boldsymbol{W}) + \frac{\nu}{2}\|\boldsymbol{W}\|_F^2 \\ \boldsymbol{\alpha}^*(\boldsymbol{x},\boldsymbol{D}) \triangleq \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \lambda_1\|\boldsymbol{\alpha}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\alpha}\|_2^2 \end{cases} \tag{5}$$

As has been shown in [20] by experiment, λ_2 could be set as 0 or as a small positive value. This agrees with the formulations of (1) and (4). With the differentiability of f in hand. We can use the gradient method to obtain $\boldsymbol{D}$ and $\boldsymbol{W}$.

$$\nabla_{\boldsymbol{W}} f(\boldsymbol{D},\boldsymbol{W}) = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\boldsymbol{W}} C(Y_i, B_i) + \nu\boldsymbol{W} \tag{6}$$

In the above formulation, note that $C(Y_i, B_i) = log(1+e^{-Y_i \max_{j=1,\ldots n_i} l(\boldsymbol{x}_{ij},\boldsymbol{\alpha},\boldsymbol{W})})$ is convex, but not a smooth function of $\boldsymbol{W}$. We therefore use sub-gradients instead of gradients.

As for the optimization of $\boldsymbol{D}$,

$$\nabla_{\boldsymbol{D}} f(\boldsymbol{D},\boldsymbol{W}) = -\boldsymbol{D}\boldsymbol{\beta}^*\boldsymbol{\alpha}^* + (\boldsymbol{x}_t - \boldsymbol{D}\boldsymbol{\alpha}^*)\boldsymbol{\beta}^{*T} \tag{7}$$

where $\boldsymbol{\beta}^*$ is

$$\boldsymbol{\beta}^*_{\wedge c} = 0, \boldsymbol{\beta}^*_{\wedge} = (\boldsymbol{D}_{\wedge}^T\boldsymbol{D}_{\wedge} + \lambda_2\boldsymbol{I})^{-1}\nabla_{\boldsymbol{\alpha}_{\wedge}} C(Y_i, B_i) \tag{8}$$

In the above formulation, sub-gradient is also used. $\wedge$ denotes the indices of the nonzero coefficients of $\boldsymbol{\alpha}$, and $\wedge c$ the indices of the zero coefficients.

3.3 Abnormal Event Detection

Given a learned dictionary $\boldsymbol{D}$ and the classification parameter $\boldsymbol{W}$, classification of an unlabeled bag B_i follows the following procedure.

First, obtain the sparse codes of the sub-events in B_i using the method of modified LARS with the learned dictionary $\boldsymbol{D}$. Second, use formulation (3) to classify B_i with the learned classification parameter $\boldsymbol{W}$. If the sign of bag B_i is negative, bag B_i is a normal bag or a normal event. While the sign of bag B_i is positive, bag B_i is an abnormal bag or an abnormal event.

If we are not only interested in the label of the bag, but also the label of the instances, i.e. the label of sub-events, we can use formulation (2) to find such results. Interestingly, given a properly trained dictionary and a classification function, the classification of sub-events is fairly accurate. This is shown in Section 4.2.

4 Experiments and Results

To verify the effectiveness of the proposed method, it was tested on two datasets, the UMN dataset [21] and the UCSD Ped1 dataset [22]. On the UMN dataset, we present the detection result of events. On the UCSD Ped1 dataset, we show detection results of sub-events.

4.1 UMN Dataset

The UMN dataset consists of 3 different scenes of crowded escape events, and the total number of frames of the video is 7740 (1450, 4415 and 2145 for scenes 1, 2, and 3, respectively). The original resolution of the UMN dataset is 320×240. Since the events happened in three scenes are the same, we used the first 625 frames of scene 1 to train the model and left the rest for testing. Several frames in a succession were viewed as a bag; and in our experiments, we took every two frames as a bag. The MHOF features were extracted in moving cells. In total, 400 normal bags and 100 abnormal bags were used for training. During initialization, the initial dictionary and sparse codes of instances used in the MIDL were first generated using the unsupervised dictionary learning method. With the initial sparse codes at hand, the classification parameter was optimized using the gradient descent method.

The detection results of the proposed method are shown in Fig. 2. The normal events are noted in green (or light) and abnormal in red (or dark) in the indication bar. Note that the first 625 frames were used for training. Because abnormal event cannot occur in a single frame, temporal smoothing was applied. Table 1 provides a comparison of the method with some state-of-the-art methods.

Table 1. Comparison of the proposed MIDL method and some state-of-the-art methods for abnormal event detection on the UMN dataset

Method	Area under ROC
MIDL(scene1)	0.8927
MIDL(scene2)	0.7541
MIDL(scene3)	0.9482
Social Force[15]	0.96
Optical Flow[15]	0.84

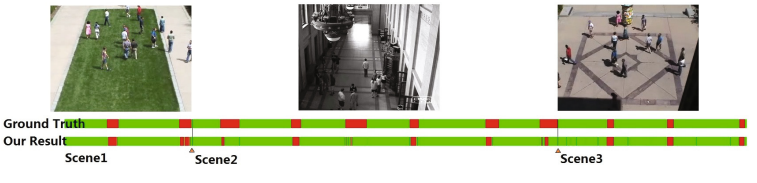


Fig. 2. Ground truth and our detection results: the normal events are marked in green (or light) and abnormal in red (or dark) in the indication bar

4.2 UCSD Ped1 Dataset

The UCSD Ped1 dataset contains 34 short clips for training of normal event patterns and 36 clips for testing with each clip containing 200 frames. The first

two testing clips were used for training because in the proposed method both normal and abnormal events were needed in the training stage. Some detection results of our method are shown in Fig. 3, in which cells containing sub-events that are classified as abnormal using formulation (2) are marked. As can be seen, though no information about the label of sub-events was used, the method was able to learn the concept of normal and abnormal automatically with the help of the label of training bags.

Fig. 3. Detection results on UCSD Ped1 dataset: regions marked in red (or shaded) are cells identified by the proposed method as containing abnormal sub-events

5 Conclusions

A method termed Multi-instance Dictionary Learning is proposed for automatic detecting abnormal events in videos. In this framework, events are defined as bags, and sub-events are defined as instances. The method is capable of learning a dictionary that is able to generate discriminant sparse codes. Using a classification function we are able to classify an event as normal or abnormal considering the label of the sub-events that constitute the event. Promising results of the proposed method are shown on two benchmark datasets.

Acknowledgements. We would like to acknowledge the support for this work from the National Science Foundation of China (Grant Nos.61035003, 61175042, 61021062), the National 973 Program of China (Grant No. 2009CB320702), the 973 Program of Jiangsu, China (Grant No. BK2011005) and Program for New Century Excellent Talents in University (Grant No. NCET-10-0476).

References

1. Gong, S., Xiang, T.: Recognition of group activities using dynamic probabilistic networks. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, pp. 742–749. IEEE (2003)
2. Breitenstein, M., Grabner, H., Van Gool, L.: Hunting nessie-real-time abnormality detection from webcams. In: 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1243–1250. IEEE (2009)
3. Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I.: Semi-supervised adapted hmms for unusual event detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 611–618. IEEE (2005)

4. Loy, C.C., Xiang, T., Gong, S.: Stream-Based Active Unusual Event Detection. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 161–175. Springer, Heidelberg (2011)
5. Loy, C.: Activity understanding and unusual event detection in surveillance videos (2010)
6. Zhou, Z.: Multi-instance learning: A survey. AI Lab. Department of Computer Science and Technology. Nanjing University, Tech. Rep. (2004)
7. Zhao, B., Fei-Fei, L., Xing, E.: Online detection of unusual events in videos via dynamic sparse coding. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3313–3320. IEEE (2011)
8. Cong, Y., Yuan, J., Liu, J.: Sparse reconstruction cost for abnormal event detection. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3449–3456. IEEE (2011)
9. Fu, Z., Hu, W., Tan, T.: Similarity based vehicle trajectory clustering and anomaly detection. In: IEEE International Conference on Image Processing, ICIP 2005, vol. 2, pp. II–602. IEEE (2005)
10. Saleemi, I., Shafique, K., Shah, M.: Probabilistic modeling of scene dynamics for applications in visual surveillance. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(8), 1472–1485 (2009)
11. Piciarelli, C., Foresti, G.: On-line trajectory clustering for anomalous events detection. Pattern Recognition Letters 27(15), 1835–1842 (2006)
12. Kim, J., Grauman, K.: Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 2921–2928. IEEE (2009)
13. Shi, Y., Gao, Y., Wang, R.: Real-time abnormal event detection in complicated scenes. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 3653–3656. IEEE (2010)
14. Kratz, L., Nishino, K.: Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 1446–1453. IEEE (2009)
15. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 935–942. IEEE (2009)
16. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning. Arxiv Preprint arXiv:0809.3083 (2008)
17. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing 20(1), 33–61 (1999)
18. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. The Annals of Statistics 32(2), 407–499 (2004)
19. Lee, H., Battle, A., Raina, R., Ng, A.: Efficient sparse coding algorithms. In: Advances in Neural Information Processing Systems, vol. 19, p. 801 (2007)
20. Mairal, J., Bach, F., Ponce, J.: Task-driven dictionary learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (99), 1–1 (2010)
21. UMN: Unusual crowd activity dataset of university of minnesota, http://mha.cs.umn.edu/movies/crowdactivity-all.avi
22. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1975–1981. IEEE (2010)

Static Packet Routing in NoC Platform Using ACO-Based Algorithms

Luneque Silva Junior[1,⋆], Nadia Nedjah[2], and Luiza de Macedo Mourelle[3]

[1] Systems Engineering and Computer Science Program, COPPE
Federal University of Rio de Janeiro, Brazil
luneque@cos.ufrj.br
[2] Department of Electronics Engineering and Telecommunications
State University of Rio de Janeiro, Brazil
nadia@eng.uerj.br
[3] Department of Systems Engineering and Computation
State University of Rio de Janeiro, Brazil
ldmm@eng.uerj.br

Abstract. Networks-on-Chip (NoC) have been used as an interesting option in design of communication infrastructures for embedded systems, providing a scalable structure and balancing the communication between cores. Because several data packets can be transmitted simultaneously through the network, an efficient routing strategy must be used in order to avoid congestion delays. In this paper, ant colony algorithms were used to find and optimize routes in a mesh-based NoC, where several randomly generated applications have been mapped. The routing optimization is driven by the minimization of total latency in packets transmission between tasks. The simulation results show the effectiveness of the ant colony inspired routing by comparing it with general purpose algorithms for deadlock free routing.

Keywords: Network-on-chip, packet routing, ant colony optimization.

1 Introduction

In the design of embedded systems, the communication plays an important role. To assist the designer, computational tools for project, or EDAs (Electronic Design Automation), are used. The purpose of EDAs is to optimize intermediate stages of network-on-chip (NoC) [1] projects, in order to obtain quickly a more efficient design implementation [2]. In general, NoCs are developed to perform a specific application. The EDA tools must be able to use information about the desired application (at a high level of abstraction) and, through successive stages of optimization, implement a solution that meets the design specification so that the execution is more efficient.

⋆ The work of this author is supported by CAPES, the Coordination of Improvement of Higher Education Personnel of the Brazilian Federal Government.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 84–91, 2012.

Many of the techniques used for routing in NoCs, such as the *XY algorithm*, were originally developed for computer networks and multiprocessor systems. The XY algorithm is a routing technique widely used in 2D mesh networks with wormhole switching. It works by sending packets over the network first horizontally (X dimension), then vertically (Y dimension) [3]. Glass and Ni have proposed the so-called *Turn Model* for adaptive, *livelock* and *deadlock* free algorithms [4]. A turn is a change of 90° in the direction of packet transmission. The main idea of this model is to restrict the amount of turns that a packet route can go through in order to avoid the formation of cycles that cause deadlocks. A related approach is the *Odd-Even turn model* [5] for designing partially adaptive deadlock-free routing algorithms. Unlike the turn model, which relies on prohibiting certain turns in order to avoid deadlock, this model restricts the locations where some types of turns can be taken. As a result, the degree of routing adaptiveness provided is more even for different source-destination pairs.

In a NoC architecture, delays in communication may occur in congestion situations, when multiple packets could be transmitted using the same switch at the same time. In order to overcome this congestion problem, and thus accelerating the packet delivery, which, in turn, would allow for an improvement of the whole execution time of the system, this paper proposes a route optimization step in the design of NoCs, or more precisely, an adaptive and static routing. In this paper, the algorithm used in the search for routes is the Ant Colony Optimization (ACO) [6].

The reminder of this paper is organized as follows. In Section 2, we do an overview on ACO meta-heuristics. The proposed routing is presented in Section 3. Simulation results are presented in Section 4. The paper closes with a conclusion and a description of future work in Section 5.

2 Ant Colony Optimization

Ant algorithms, also known as Ant Colony Optimization (ACO), are a class of heuristic search algorithms that have been successfully applied to solving NP hard problems [7]. Ant algorithms are biologically inspired in the behavior of colonies of ants, and in particular how they forage for food. One of the main ideas behind this approach is that the ants can communicate with one another through indirect means by making modifications to the concentration of highly volatile chemicals called *pheromones* in their immediate environment.

In the artificial ant colony approach, following an iterative process, each ant builds a solution by using two types of information locally accessible: problem-specific information, and information added by ants during previous iterations of the algorithm. In fact, while building a solution, each ant collects information on the problem characteristics and on its own performance, and uses this information to modify the representation of the problem, as seen locally by the other ants. The representation of the problem is modified in such a way that

information contained in past good solutions can be exploited to build new better solutions. This form of indirect communication mediated by the environment is called *stigmergy*, and is typical of social insects. Several ant algorithms make use of the structure shown in the Algorithm 1 [8].

Algorithm 1. ACO meta-heuristics

```
1: initialize parameters and pheromone trails;
2: while termination condition not met do
3:    construct ant solutions;
4:    local search (optional);
5:    update pheromone trails;
6: end while;
```

3 ACO-Based Routing

The Ant Colony Optimization, with the ability to search for paths, emerged as a powerful solution for routing problems. Thus, this paper presents the use of the ACO meta-heuristics in the construction of routing algorithms. Two models of static routing for NoCs are proposed.

3.1 Network Specification

We assume that the network consists of switches with five communication ports: four to communication with neighboring switches and one is for local communication with the resource. The network topology is a two dimension mesh. The switching technique adopted was the wormhole [3]. In this method, packets are divided into smaller units called *flits* (flow-units). It is assumed that each communication channel has a width of one flit. The switches are considered bufferless, i.e., can store only a single flit,, using no virtual channels.

3.2 Proposed Routing Algorithms

In this paper, we propose two algorithms to perform the static routing in NoC design. We rely on two well-known ant algorithms. Thus, these methods were called *REAS* (Routing inspired on Elitist Ant System) and *RACS* (Routing inspired on Ant Colony System). Both algorithms search for paths in an *architecture characterization graph* that represents the network 2D mesh topology. These algorithms make use of *multiple ant colonies*, where each colony is responsible for searching the route of a packet. In this approach, each colony has its own pheromone and ants. Nevertheless, the colonies must exchange information in order to minimize the latency of their respective packets. Therefore, the route found by an ant in a given colony is visible to the ants in other colonies, because these packets are being transmitted simultaneously and within the same network. A simplified pseudo-code of REAS is shown in Algorithm 2.

Algorithm 2. REAS algorithm

Require: network and ACO parameters;
while total of cycles **do**
  for $k = 1 \rightarrow number\ of\ ants$ **do**
    for $g = 1 \rightarrow number\ of\ packets$ **do**
      $Ant_{k,g}$ constructs a solution;
      compute $Ant_{k,g}$ pheromone;
    end for
    compute the elitist pheromone;
    accumulate the pheromone of actual ants;
  end for
  update the global pheromone;
end while
return best solution;

In the proposed algorithms, ants in a network node are only aware of two things: the first is the pheromone concentration in the surrounding nodes; the second is the load on a node, which dictates the waiting time in each of the four possible transmission directions.

The *Elitist Ant System* (EAS) is directly inspired by the *Ant System*, the first of ant algorithms [6]. The EAS is characterized mainly by the use of the concept of *elitism*, in order to differentiate ants that carry out better solutions. In the REAS algorithm, ants build paths through the network selecting the next node based on (1), where p_{ij}^k is the probability of ant k going from node i to node j.

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_j(t)^\alpha \cdot {\eta_{ij}}^\beta}{\sum_{k \in a_k} \tau_k(t)^\alpha \cdot {\eta_{ik}}^\beta} & \text{if } j \in a_k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The probability of selecting a particular direction is a function of pheromone concentration and network load in that direction. These two parameters are weighted by an importance constant α and β, respectively. The network load is used indirectly by η_{ij}, defined in (2), where C_{ij} represents the load in transmission from i to j.

$$\eta_{ij} = 1/C_{ij} \tag{2}$$

At the end of each iterative cycle, the pheromone of all colonies is updated according to (3). Part of the pheromone of the previous iteration is reduced by evaporation rate ρ, and then reinforced by the contribution of all m ants in the current cycle. The pheromone also receives the reinforcement of elitist ants: those that achieve the best solutions deposit their pheromone in every cycle, so directing the search in subsequent cycles.

$$\tau_{t+1} = (1 - \rho) \cdot \tau_t + \sum_{k=1}^{m} \Delta\tau^k + \tau_{elite} \tag{3}$$

The pheromone in the path find by a single ant k is defined in (4), where Q is a constant and L_k represents the total latency imposed by the solution. It is easy to note that the ants with the worst results provide a smaller amount of pheromone.

$$\Delta\tau^k = Q/L_k \tag{4}$$

The second ant algorithm, proposed in this work, is described as follows. The *Ant Colony System* [9] differs from others by the method of selection and the way which the pheromone is updated. Thus, the RACS uses the so-called *pseudo-random proportional* rule.

$$j = \begin{cases} argmax_{j\in[1,4]}\left\{\tau_j \cdot \eta_{ij}^{\beta}\right\} & \text{if } q \leq q_0, \\ S & \text{otherwise} \end{cases} \tag{5}$$

As shown in (5), the probability for an ant to move from node i to node j depends on a random variable q, uniformly distributed over $[0, 1]$, and a parameter q_0. If $q \leq q_0$, then the next node is directly selected by $argmax_{j\in[1,4]}\{\tau_j \cdot \eta_{ij}^{\beta}\}$, i.e., the direction with the largest value of $\tau_j \cdot \eta_{ij}^{\beta}$. Otherwise, the next node is defined by S, that uses a selection method similar to that employed by EAS (1).

Furthermore, the RACS algorithm uses a double pheromone update. The first update, called *local update*, is performed by all ants in each step of construction of a solution. This local update is defined in (6).

$$\tau_{t+1} = (1 - \rho) \cdot \tau_t + \rho \cdot \tau_0 \tag{6}$$

The parameter ρ is the evaporation constant, and τ_0 is the initial pheromone at each node. The second pheromone update, the *offline update*, is applied at the end of each iteration only by the *best-so-far* ant (the ant that found the best solution). The update is given by (7), where $\Delta\tau_j$ is the reinforcement of the best ant pheromone.

$$\tau_{t+1}^j = \begin{cases} (1 - \rho) \cdot \tau_t^j + \rho \cdot \Delta\tau_j & \text{if } j \in best \\ \tau_t^j & \text{otherwise} \end{cases} \tag{7}$$

In both REAS and RACS algorithms the same stop condition was used, which is a maximum number of iterative cycles.

4 Evaluation Experiments and Results

A cycle-accurate network simulator was implemented in Matlab. It supports 2D mesh networks with wormhole switching. To evaluate the performance of the proposed methods, networks were simulated with four different routing algorithms: REAS, RACS, XY and Odd-Even (OE). The time unit adopted is the *simulator cycle*, where one cycle is the transmission time of one flit.

All algorithms were executed with Matlab Version 7.7.0.471 (R008b). The simulations were performed on PCs with Intel Core i7 950 3GHz, 8Gb RAM and Microsoft Windows 7 Home Premium operating system.

Applications can be described by a *task graph*, a data structure divided into blocks responsible for specific tasks. These blocks, in turn, exchange information in order to complete the application execution. The task graph is denoted by $GT = G(T, D)$, an acyclic and weighted directed graph. Each node of T is a task, or an application processing module. In general, an operation is a well defined task, as a mathematical calculation or a data encoding. Each arc of the set D characterizes the data dependencies between two tasks.

4.1 Simulation with Applications

In the simulations, we used five sets of graphs of synthetic applications. These graphs were randomly generated with the aid of the *Task Graph For Free* (TGFF) [10]. The TGFF is a general purpose, user controllable pseudo-random graph generator, widely used in embedded real-time systems research. The task graph is composed by a set of nodes and arcs. In order to explore the behavior of applications with parallel characteristics, tasks were generated following a *fork-join* structure, with the start task sending packets to several destinations, and the end task receiving packets from several origins. Between start and end tasks exist intermediate tasks, arranged in various levels of parallelism. Tasks at the same level can run concurrently and independently.

Therefore, 40 graphs are generated, being arranged in 4 sets of 10 graphs. Each set (called $Ex1$, $Ex2$, $Ex3$ and $Ex4$) has a different characteristic on the maximum number of tasks in each level. Within a set, each of the ten tasks are differentiated by the number of intermediate levels.

From the information of a graph, the routing can be accomplished by identifying which packets are generated by tasks at the same level, i.e., which packets may be transmitted simultaneously. Also, a simple random mapping is employed, were each task is assigned randomly to network nodes. The Algorithm 3 was used to perform this process.

Algorithm 3. Mapping and routing of application

Require: Task Graph;
1: **define** size of NoC;
2: perform the mapping;
3: **for** $l = 1 \rightarrow \#levels$ **do**
4: get all arcs in level l;
5: read t_{start} of source tasks;
6: perform the routing;
7: write t_{start} of destination tasks;
8: **end for**
9: $t_{execution} \leftarrow t_{start}(last\ task) + t_{comp}(last\ task)$
10: **return** routing paths, $t_{execution}$;

4.2 Results

The simulations were performed by submitting applications to four different routing algorithms and measuring its total execution time. This consists of execution of all individual tasks on a critical path plus the communication time of these tasks. The so-called *packet delay* is the difference between the value obtained using a specific routing algorithm and the optimal value of the network without congestion. To calculate this ideal value, we used a modified XY algorithm, called *dummy XY*. In this routing, the XY algorithm is used to define the communication time using shortest paths. But unlike the real XY (and any other routing algorithm), the potential congestion delays are not counted.

The Fig. 1 shown results of the performed simulations. The values of packet delay is presented for the four routing algorithms in each of 40 applications. These values are a mean of packet delay in 10 different mappings.

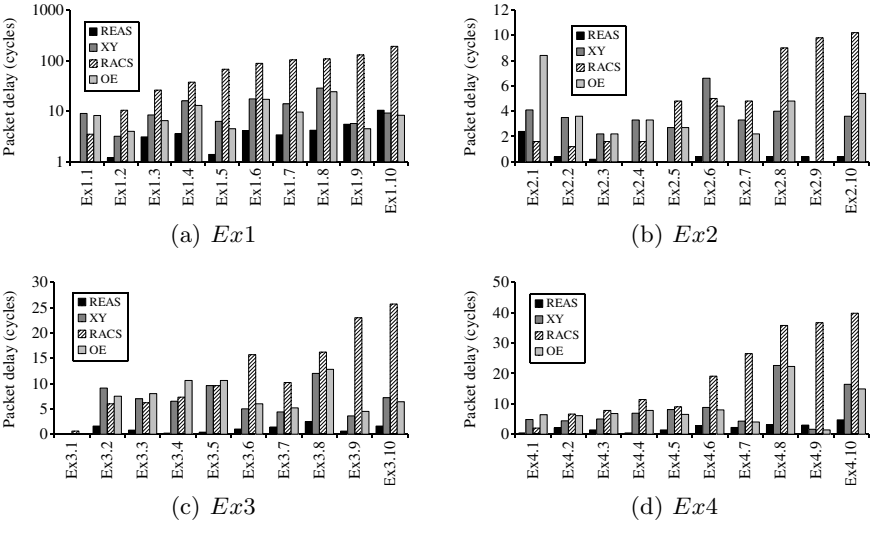


Fig. 1. Packet delay in 4 sets of applications

The results show the REAS getting the best results in the simulations when compared with other routing algorithms. The REAS is exceeded only in 5 of 40 tests. These low delay values show that the REAS is able to find good solutions to routing problem, independent of mapping or complexity of graph. For routing based on XY algorithm and Odd-Even turn model, there is a wide variation in average delays obtained for a given set of graphs. This large deviation in the delay values may suggest that XY and OE are very sensitive to mapping adopted, even more than the complexity of the application. The worst results were obtained with the second proposed routing. The values found by RACS are increasing due to the complexity of the used graphs.

5 Conclusion

Static routing is an efficient solution in NoCs designed to run always the same set of applications. This is because the communication paths need only be defined one time. In this paper we propose the use of ACO-based algorithms in the optimization of paths in the static routing step in NoC design. The performance of these algorithms was evaluated using randomly generated graphs, modeled to imitate applications with parallel tasks. Best results were obtained with REAS algorithm. Future work may study how to increase the performance of proposed algorithms and their evaluation with real-world applications.

Acknowledgment. We are grateful to the Brazilian agencies FAPERJ (*Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro,* `www.faperj.br`), CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico,* `www.cnpq.br`) and CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior,* `www.capes.gov.br`) for their continuous financial support.

References

1. Benini, L., Micheli, G.D.: Networks on chips: A new soc paradigm. Computer, 70–78 (2002)
2. Edwards, S., Lavagno, L., Lee, E., Sangiovanni-Vincentelli, A.: Design of embedded systems: Formal models, validation, and synthesis. Proceedings of the IEEE 85(3), 366–390 (1997)
3. Ni, L.M., McKinley, P.K.: A survey of wormhole routing techniques in direct networks. IEEE Tran. on Computers 26, 62–76 (1993)
4. Glass, C.J., Ni, L.M.: The turn model for adaptive routing. In: Proceedings of the 19th Annual International Symposium on Computer Architecture (ISCA 1992), pp. 278–287. ACM, New York (1992)
5. Chiu, G.: The odd-even turn model for adaptive routing. IEEE Transactions on Parallel and Distributed Systems 11(7), 729–738 (2000)
6. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 26(1), 29–41 (1996)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence From Natural to Artificial Systems. Oxford University Press, New York (1999)
8. Dorigo, M., Birattari, M., Stützle, T.: Ant colony optimization - Artificial ants as a computational intelligence technique. IEEE Computational Intelligence Magazine 1(4), 28–39 (2006)
9. Dorigo, M., Gambardella, L.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)
10. Dick, R.P., Rhodes, D.L., Wolf, W.: Tgff: task graphs for free. In: Proceedings of the 6th International Workshop on Hardware/Software Codesign, pp. 97–101. IEEE Computer Society (1998)

Segmentation of Mammography by Applying Extreme Learning Machine in Tumor Detection

Cordeiro F.R.[1], Lima S.M.L.[1], Silva-Filho A.G.[1], and Santos W.P.[2]

[1] Federal University of Pernambuco, Informatics Center, Recife, Brazil
{frc,smll,agsf}@cin.ufpe.br
[2] Federal University of Pernambuco,
Department of Biomedical Engineering, Recife, Brazil
wellington.santos@ufpe.br

Abstract. Locating regions of tumor in digital mammography images is a hard task even for experts. Consequently, due to medical experience, different diagnoses to an image are commonly found. Therefore, the use of an automatic approach for detecting tumor regions is important to avoid misdiagnosis. In this work, the Extreme Learning Machine (ELM) neural network was used to segment tumor regions of digitized mammograms available in the Mini-Mias database. A set of images were selected for training, while different images were used for testing. Results showed that ELM provides an over 81% classification rate, being able to segment the region of tumor with high accuracy. By comparing ELM with MLP network, it was possible to conclude that ELM has a faster learning time, with a higher training and testing accuracy.

Keywords: Breast Cancer, Segmentation, Neural Network, Extreme Learning Machine.

1 Introduction

The World Health Organization (WHO) estimates that 1,100,000 new cases of breast cancer appear yearly worldwide [1]. Recent changes in lifestyle in society in-crease obesity, which has a direct influence on breast cancer [2]. In developing countries, like Brazil, this type of cancer is one of the main causes of death among women [2][3].

It is estimated that the period between the beginning of the tumor and its growth until it becomes palpable, reaching around 1cm, is of about 10 years [4]. During this period, breast imaging is essential to the tumor attendance. The correct evaluation of the tumor size takes an important role in the planning of the breast cancer treatment, avoiding mutilating surgeries, such as mastectomy [5]. However, image devices determined by BMH (Brazilian Ministry of Health) [6] for the detection of breast cancer are quite inefficient at the evaluation of the nodule sizing. These methods depend substantially on professional examiner's experience.

Furthermore, the diagnosis through images is complex, mainly because of the large variability of cases. Many cases seen in clinic practice don't fit classic images and descriptions precisely [7]. Consequently, traditional techniques in image processing have been applied in medical field to turn diagnosis less susceptible to errors through accurate identification of anatomic anomaly [8][9].

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 92–100, 2012.

Usually, however, there isn't any concerning about the performance of these applications. Image processing is a task with a usually high computational cost. Therefore, the optimization time (learning) and the time for feedback to the user can be fairly long. Thus, techniques with reduced learning time have been prioritized in tumor detection.

In this work, is proposed the use of ELM (Extreme Learning Machine) neural network [10] to segment mammogram images. The ELM is a network with fast learning time and high accuracy, which makes it suitable to be applied in segmentation problems. The results of ELM network are compared with MLP (Multi-Layer Perceptron) network [11].

2 ELM Neural Network

Artificial Neural Networks (ANN) is one of the successful techniques used for classification problems [11]. The segmentation problem, however, also is a classification problem, where each pixel of image should be classified as being a pixel of a region of interest or not.

In most types of neural networks, such as MLP [12], knowledge about network parameters is necessary to obtain an optimal performance in the solution of the problem. A usual concern of this kind of network is to avoid being stuck in local minima, making it necessary to add methods to control the network to escape from those regions. Another common characteristic of this kind of network is the high training time necessary to make the network able to perform classifications correctly.

The ELM (Extreme Learning Machine) network [10], however, has as main characteristic the training speed and data prediction when compared with other nonlinear methods. The ELM network is a single hidden layer network, not recurrent, based on an analytical method to estimate the network output weights, in any random initialization of input weights. As the output weights are determined analytically, the network is obtained through a few steps and with low computational cost. Different from MLP, in the ELM network it is not necessary to define a training stopping criteria neither the use of cross validation, as the network presents a single iteration.

The learning process of ELM network is based on Moore-Penrose generalized inverse [x], where are calculated the output weights. The learning is performed in batch, where all data are presented to the network before the network tuning. The learning process of ELM involves a single interaction, turning the training step quite faster than conventional approaches. Furthermore, because it is not based on the descendent gradient, the network does not suffer the problem of local minima neither it is necessary the definition of the parameter of learning rate.

3 Proposed Methodology

3.1 Experimental Environment

The proposed work used the images from mini-MIAS database [13]. The ground truth image, which is the expected segmented image, was obtained by a built supervised auxiliary software, which uses the indications provided from database, being used for network testing and training.

3.2 Neural Network Topology

The ELM network used is composed of a topology with 25 input neurons, 300 hidden neurons and 2 output neurons. The activation function used on the network was the sigmoidal function. Those values were obtained empirically, selecting the ones witch had the best results. The ELM network topology is shown in Figure 1(a).

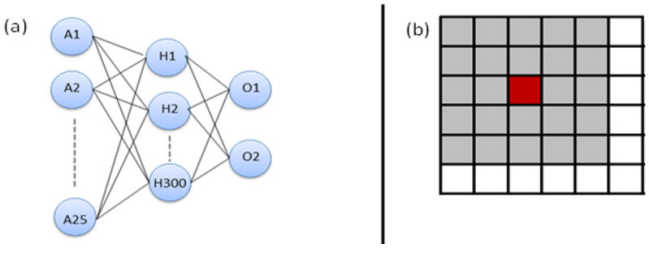


Fig. 1. (a) ELM Network Topology, (b) Achievement of input network data

The input neurons receive pixel values as input from the region that is being trained. The signal is then propagated through the intermediate layer and next to the output layer. The two output neurons represent the two classes of the problem: tumor and non-tumor. If neuron O1 presents a higher output value than O2, then the central pixel of the region being analyzed is classified as tumor. Otherwise, the pixel is classified as non-tumor.

Figure 1 (b) represents the choice of a pixel that will be trained by the network, represented by the darker pixel. The pixel should be classified as being a tumor pixel or not. Thus, a neighboring region is selected, in a 5x5 area, represented in the picture by a gray region. All the pixels in the neighboring region will be the input of ELM network. Each neighboring region is a network input, where the classification of the center pixel will be made.

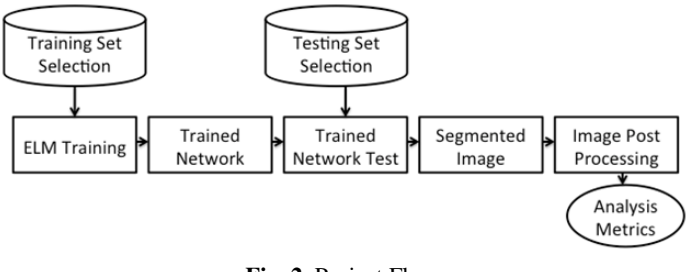


Fig. 2. Project Flow

3.3 Project Flow

Figure 2 describes the flow used in the proposed work, listing each one of its stages. First, a training set with samples from regions with and without tumor is defined.

The training set is obtained through the selection of the region of interest from different images. Each training image provided for the network is associated with the ground truth image, aiming to compare actual and estimated outputs.

In the selection of a training set, pixel samples from the region of interest are chosen, containing samples of pixels with tumor and non-tumor pixels. For each chosen pixel, the neighboring region is used for the network training. The selection of the testing set is analogous, selecting regions of interest from images that will be tested, where all the pixels from a region of interest are submitted to network classification.

Once all input training data are submitted to the network, the weights are tuned and the trained network is obtained. Next, test samples are submitted to the network, classifying pixel as tumor or non-tumor. From the classification it is possible to segment test images.

After the segmentation, the region may contain some fragmentations. Thus, the segmentation quality is improved through the elimination of external noises, through the technique of connected components [14]. The analysis of ELM results is made by using the metrics described as follow.

3.4 Analysis Metric

Two metrics were applied for the analysis of the ELM network classification: global accuracy index, from confusion matrix, and Kappa index [15], which evaluates the accuracy of the classifier. The Kappa index is used as a method to evaluate globally the quality of a classification process in a single metric, involving information of false positives, false negatives, true positives and true negatives, which are obtained from the confusion matrix. Table 1 shows the calculation made in the confusion matrix.

Table 1. Confusion Matrix

Class	C_1 Prediction	C_2 Prediction	Class Accuracy	Global Accuracy
C_1	True Positives T_p	False Negatives F_n	$\frac{T_p}{T_p + F_n}$	$\frac{T_p + T_n}{n}$
C_2	False Positives F_p	True Negatives T_n	$\frac{T_n}{F_p + T_n}$	

As illustrated in Table 1, the global accuracy index is obtained through T_p, T_n and n, which are the amount of true positive classifications, true negatives and the amount of set data, respectively. True positive classifications are done when the pixel of class C_1 (pixel with tumor) is predicted as being of class C_1. True negative classification is when the pixel of class C_2 (pixel without tumor) is predicted as being of class C_2. False Negative is when a pixel of class C_1 is predicted as class C_2, and false positive is when a pixel from class C_2 is predicted as class C_1. The Kappa index is calculated from the confusion matrix through the following equation:

$$K = \frac{n\sum_{i=1}^{c} x_{ii} - \sum_{i=1}^{c} x_{i+}x_{+i}}{n^2 - \sum_{i=1}^{c} x_{i+}x_{+i}}, \quad (1)$$

in which x_{ii} is the value of line i and column i; x_{i+} is the sum of line i and x_{+i} is the sum of column i of the confusion matrix; n is the total sample number and c is the total number of classes. The Kappa index takes on a value between 0 and 1, with the classification described in Table 2, according to the index.

Table 2. Kappa index and the corresponding classifier performance

Kappa Index	Performance
<0	Very Bad
0<k≤0,2	Bad
0,2<k≤0,4	Regular
0,4<k≤0,6	Good
0,6<k≤0,8	Very Good
0,8<k≤1,0	Excelent

Based on value of the Kappa index, represented by k, it is possible to assign a performance index to the classifier, rating it between very bad and excellent. The Kappa index for ELM network is analyzed in the results section of this article.

4 Results

In order to analyze the ELM network performance, it was considered a case study with separated images for training and testing of the network. The ELM network was compared with MLP network, which was chosen because it is a network widely used in many areas. In following the results are described for this scenery.

4.1 Case Study

In the case study analyzed, 25 images were separated in training and test images, using the k-fold method, where was used a *k* value equal to 5. The objective is that the network can be trained with pixels from a set of images and after that can generalize de tumor detection to different images. The training and testing regions of images are pre-defined through the selection of a region of interest where the tumor is located.

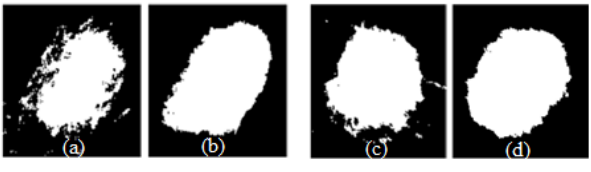


Fig. 3. (a) Image 1 segmented region; (b) Image 1 expected region; (c) Image 2 segmented region; (d) Image 2 expected region

In Figure 3 is shown two examples of images segmented by ELM network. The Figure 3 (a) is the first example of segmented region, while Figure 3 (b) is the expected region for the same image. Figure 3(c) and Figure 3 (d) represents another example of actual and expected segmentation to a different image.

Can be seen that results of segmentation were close to expected, achieving a good segmentation performance for all analyzed images.

Table 3 shows the results from Figure 3 (b). The network achieved a global accuracy of 92%, with accuracy of class 1 (pixel with tumor) of 91%, while the accuracy of class 2 (pixel without tumor) was of 94%. The Kappa index obtained was of 0.73, which corresponds to a performance described as "very good".

Table 3. Confusion matrix for Figure 3 (b) segmentation

Class	Class	Predicted as C1	Predicted as C2	Class Accuracy	Global Accuracy	Kappa
C1	C1	11739	1112	0.913	0.9255	0.8049
C2	C2	369	6661	0.948		

The results were also compared with the MLP network, with the performance shown in Table 4.

Table 4. Comparison between the results of the ELM and MLP networks for Figure 3(b) segmentation

	ELM	MLP
Training Time	63	130
Testing Time	1,67	0,83
Training Hit Rate	89%	77%
Testing Hit Rate	92%	66%

As demonstrated in Table 4 the ELM network required less than half of training time when compared to the MLP network. The ELM network, however, has a bit higher testing time than the MLP network. In relation to the training and testing hit rate, the EML network obtained better results, achieving an accuracy close of 92%. The MLP network was able to hit rates of 77% and 66% for training and testing, respectively.

Table 5. Performance Comparison between ELM and MLP networks

	MLP	*ELM*
Training Accuracy	75%	85%
Testing Accuracy	60%	81%
Kappa	0.2 (Bad)	0.49 (Good)
Training Time	11,01 min	9.96 min
Testing Time	11s	24s

On the whole, was possibly to verify that the ELM network presented good generalization results for image segmentation, presenting better results when compared to MLP network. Analyzing the average case, with the average result of all tested images, after using the k-fold method, could be obtained the results found in Table 5.

As shown above, the ELM network obtained a good training accuracy, with an accuracy rate of 85%. The accuracy for not trained images was of 81%, achieving a good generalization power.

The Kappa index, which evaluates the classifier performance, was of 0.49, classifying the network performance as "Good". The MLP, however, did not achieve good results for training and testing. Tunings were done in MLP, changing empirically the number of neurons, training rate and stop criteria, but there was not improvement in network performance. With relation to training time, the ELM network obtained lower training time, showing that besides having a higher training accuracy, the training stage was performed faster. With relation to testing time, however, MLP network was faster.

After the segmentation performed by the ELM, the segmented region may contain some fragmentation, as shown in Figure 4 (a). In this case, the segmentation quality is improved by the elimination of internal and external noises. This is achieved through the technique of connected components. The result is shown in Figure 4 (b). After that, the diameter of the tumor is calculated through the max distance between two points, as shown in Figure 4 (c).

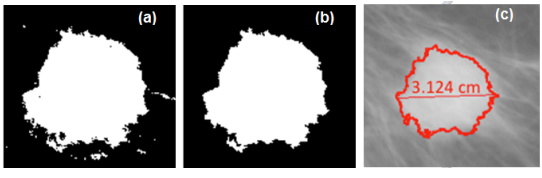


Fig. 4. (a) ELM segmentation. (b) Noise elimination by connected components. (c) The maximum diameter of a segmented region

The diameter corresponding to a perfect circle of the same area that the can be easily calculated. The diameter corresponding to the area is important because it avoids inaccuracies in the interpretation of the tumor. Thus, surgeries that mutilate, such as mastectomy, will only be employed when really necessary.

5 Conclusion

The presented work applied the ELM technique in tumor segmentation in mammography images. The ELM technique showed to have an advantage in relation to traditional networks because it has a fast learning phase, only having a need for a single iteration, besides requiring less configuration parameters.

Applying ELM network in tumor segmentation enabled observation of good results over 81% classification rate. The network also achieved good results for kappa index, which proved the network to be excellent or good in most cases. When compared with MLP network, it was possible to observe that ELM network obtained better results in relation to training time, training and testing accuracy rate.

The proposed work could evaluate the performance of ELM, when applied to the segmentation problem and compared to MLP. Thus, the proposed mechanism is expected to be used in the hospital network and to be used as a model in identification and interpretation of tumor sizing. Therefore, the efficiency and accuracy of breast exams can become less dependent of interpretation and experience on the professional examiner. Future work involves extending image database and comparing with another segmentation techniques.

Acknowledgment. The authors thank to UFPE (Reuni-UFPE nº 03/10) and FACEPE (ref. APQ-0341-1.03/10, PBPG-0527-1.03/10), both Brazilian agencies, for partial financial support.

References

1. World Health Organization. The Global Burden of Disease: 2004 update (2008)
2. Breast cancer: prevention and control from WHO (World Health Organization), http://www.who.int/cancer/detection/breastcancer/en/index.html (accessed January 05, 2012)
3. Statistical data of breast cancer in Brazil from INCA (National Cancer Institute), http://www.inca.gov.br/estimativa/2008/ (accessed January 05, 2012)
4. Clark, G.M.: Prognostic and predictive factors. In: Harris, J.R., Lippmann, M.E. (eds.) Diseases of the Breast, 2nd edn., pp. 489–514. Lippincott Williams and Wilkins, Philadelphia (2000)
5. Litière, S., Werutsky, G., Fentiaman, I.S., et al.: Breast conserving therapy versus mastectomy for stage I-II breast cancer? 20 year follow-up of the EORTC 10901 phase 3 randomised trial. The Lancet Oncology 13(4), 412–419 (2012)
6. Costa, H., Solla, J., Temporão, J.G.: Controle do Câncer de Mama: Documento de Consenso. Ministério da Saúde, Secretária de Atenção à Saúde e Instituto Nacional de Câncer (2004)
7. Juhl, J.H., Crummy, A.B., Kuhlman, J.E.: Interpretação Radiológica. In: Koogan, G. (ed.), 7th edn. (2000)
8. Da-Xi, H., Yuan, F.: An Algorithm for Medical Imaging Identification based on Edge Detection and Seed Filling. In: Computer Application and System Modeling (ICCASM), pp. 547–548 (2010)
9. Ye, S., Zheng, S., Hao, W.: Medical Image Edge Detection Method based on Adaptive Facet Model. In: Computer Application and System Modeling (ICCASM), pp. 574–578 (2010)
10. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of International Joint Conference on Neural Networks (2004)

11. Jeatrakul, P., Wong, K.W.: Comparing The Performance of Different Neural Networks for Binary Classification Problems. In: Eighth International Symposium of Natural Language Processing (2009)
12. Andrioni, V., Guingo, B.C., Santana, E.L., Pereira, W.C.A., Infantosi, A.F.C.: Comparison of artificial neural networks using texture parameters in the recognition of lesions in mammograms digitized. In: 2011 Pan American of Health Care Exchanges (PAHCE), March 28-April 1, vol. 1, pp. 426–430 (2011)
13. Suckling, J., et al.: The Mammographic Image Analysis Society Digital Mammogram Database Exerpta Medica. International Congress Series 1069, 375–378 (1994)
14. Řiha, L., Mareboyana, M.: GPU accelerated one-pass algorithm for computing minimal rectangles of connected components. In: 2011 IEEE Workshop on Applications of Computer Vision (WACV), January 5-7, pp. 479–484 (2011)
15. Thomas, S.J., Cowley, I.R.: A Comparison of Four Indices for Combining Distance and Dose Differences. International Journal of Radiation Oncology*Biology*Physics 82, 717–723 (2012)

An ANN Speed Observer Applied to Three-Phase Induction Motor

Tiago Henrique dos Santos, Alessandro Goedtel,
Sérgio Augusto Oliveira da Silva, and Marcelo Suetake

Electrical Engineering Department of the Federal Technological University of Parana,
Av. Alberto Carazzai. 1640, CEP 86.300-000, Cornélio Procópio, PR, Brazil
tiagohenrique.santos@gmail.com,
{agoedtel,augus}@utfpr.edu.br,
mclsuetake@gmail.com

Abstract. This work proposes an artificial neural network approach to estimate the induction motor speed applied to three-phase induction motor. The induction motor speed is the important variable in an industrial process. However, the direct measurement of speed compromises the driver system and control, besides increasing the implementation cost. The proposed strategy estimates the induction motor speed when it is driven by voltage source inverter with closed-loop scalar control. Simulation results are presented to validate the performance of the proposed method under motor load torque and speed reference point variations.

Keywords: Induction Motors, Neural Networks, Scalar Control, Speed Estimation.

1 Introduction

Three-Phase Induction Motors (TIM) are used in many industrial sectors as leading element to convert electrical into mechanical energy. The closed-loop control strategies of these machines make use of electronic drivers based on directly speed measurement or sensorless technologies.

Commonly the motor speed is measured by eletromechanical devices, with electromagnetic resolvers, optical encoders or brushless dc tachogenerators. However, the use of these devices present some limitation, such as the increasing driver cost, reduced mechanical robustness, low noise immunity and the requirement of a special attention with respect to applications in hostile environments [1]. Sensorless technique is mainly found in high performance applications, such as vector-controlled and direct torque controlled drivers. The main sensorless control strategies are based on open-loop estimators, e.g., the stator current and voltage monitoring, state observers, reference systems with adaptive models and estimator based on intelligent systems, such as Artificial Neural Network (ANN) and fuzzy logic [1].

Most speed estimators are obtained from the mathematical model of induction motor, where a precise knowledge of motor parameters is required [2]. Speed

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 101–108, 2012.

estimators based on State Observers (SO) need the exact values of the machine parameters for the correct operation in low-speed regions [3]. This method also requires a considerable computer effort, since the estimator algorithm demands differential equations solving. Recently, several methods for TIM speed estimation have been investigated [4–6].

The proposal of this work is the development of a parallel neural speed estimator applied to TIM. It is driven by a Voltage Source Inverter (VSI) using closed-loop scalar control with direct speed measurement. Two ANN are used in high speed and low speed range respectively. The input data of estimator, such as current and voltage are used to training and to validate the ANNs. The training data are generated from the simulation of the mathematical model of TIM. The VSI is modulated by means of Space Vector Pulse Width Modulation (SVPWM). The ANN training is carried out in a offline form. In order to validate the proposal method, simulation results of this speed observer are presented.

2 Modeling and TIM Control

The first step of an ANN supervised training is to compile the input data set, which are used to adjust the internal parameters of the network. In this procedure, the ANN must be exposed to a data set that satisfactorily describes the system behavior [7].

To generate the training data set of induction motor, several simulations are executed using Matlab/Simulink software at different speed operating points. Fig. 1 shows the block diagram which describes the input and output of the proposed model. The TIM is driven by a VSI with vector pulse width modulation (SVPWM). The adopted control strategy is based on voltage/frequency (V/f) ratio scalar control.

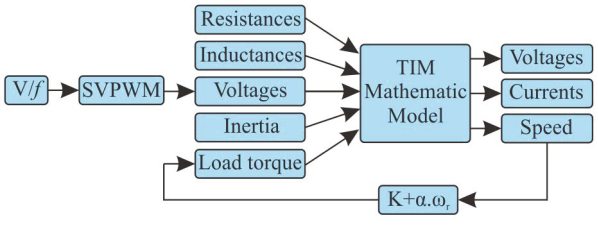


Fig. 1. Block diagram of system model

2.1 Aspects of the TIM Model

The induction motor model used in the simulations was developed in [8] and [9]. The machine parameters are obtained from a WEG manufacturing induction motor with 1 CV, 4-pole, 220/380V, IP55, stator resistance = 7.32 Ω, rotor

resistance = 2.78 Ω, stator leakage inductance = $8.95 \cdot 10^{-3}H$, rotor leakage inductance = $5.44 \cdot 10^{-3}H$, mutual inductance = $1.41 \cdot 10^{-1}H$, moment of inertia = $2.71 \cdot 10^{-3}kg.m^2$, Slip = 3.8 % and nominal torque = 4.1 Nm.

The voltage, current and rotor speed are the quantities used in the ANN training process. In this study, linear loads, which are mainly found in fans, rolling mills, piston pumps and wood saw applications, were coupled on the rotor axis to evaluate the proposed method. These loads are feature by presenting a linear relationship between the load torque and the rotor shaft speed [10].

2.2 The V/f Control

The proposed methodology used for adjusting TIM speed consist in keeping the voltage-frequency (V/f) constant, in order to maintain the magnetic flux in the air gap, which ensures operation of machine.

The scalar control method is applied in TIM speed control by changing frequency and amplitude of the machine voltage, in order to maintain the maximum torque produced by the machine constant. Thus, the electromagnetic flux produced by TIM also remains constant. In [3] presents the details of the scalar control. The block diagram of the open-loop scalar control used to generate de the training data set is shown in Fig. 2. In this figure, f^* is the set point frequency (Hz), ω_e^* is the set point angular speed (rad/s), Θ_e^* is the angular position of the proportional voltage reference and V^* is the proportional voltage reference. In closed-loop scalar control the speed is controlled by slip regulation, as the Fig. 3. The speed error between the f^* and the rotor frequency f_r generates the slip command f_{sl}^* through the proportional-integral (PI) controller and limiter. The slip command is added to de speed signal to generate the frequency command to scalar control [3].

In this control method, the slip speed limits, V_boost and the reference speed are externally adjustable quantities. These can be set from meet the TIM and load parameters.

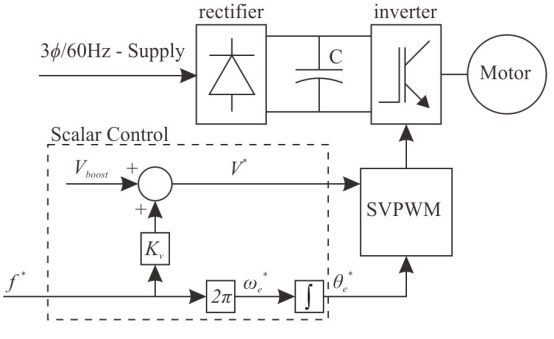


Fig. 2. Block diagram of open-loop scalar control

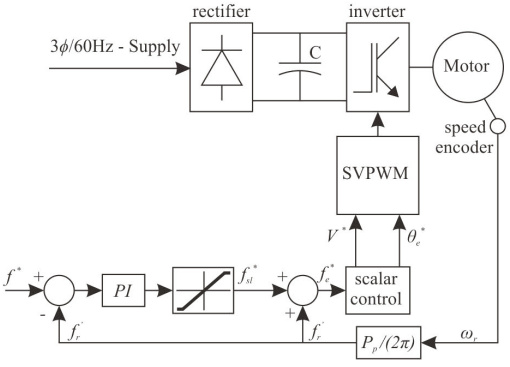


Fig. 3. Block diagram of closed-loop scalar control

3 Data Processing Model

The estimated machine speed through of currents and voltages, which increases the cost of drivers with their transducers. However, this cost is lower compared to the direct speed measurement. Following the quantities acquisition, there is a need for a signals processing, in order to extract information that can be used to mapping the rotor speed as a MIT voltage and current function.

The voltage supplied by the inverter has a switching characteristic, which passes through the low-pass filter (LPF) with 600 Hz cut off frequency before being processed in the Phase Locked Loop (PLL) system. Although the electric current are naturally filtered, due to the inductive characteristic of the machine, the same LPF is used, in order to obtain high frequency noise attenuation and provide the similar phase displacement compared with voltage signal.

In order to obtain the input patterns of the rotor speed function, stator voltage and current on synchronous reference frame $dq0$ were used. The Fig. 4 presents the block diagram of the voltages and currents treatment system. The PLL system used in this paper is based on the single phase p-PLL algorithm described in [11].

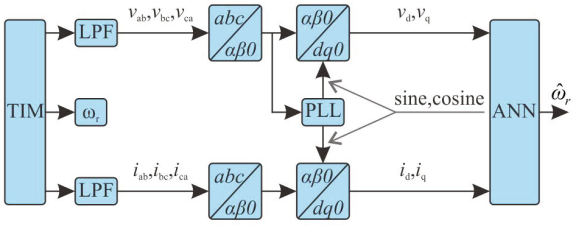


Fig. 4. Block diagram of the data processing

4 Neural Speed Estimator

The ANN has proved to be efficient in various engineering problems. In this paper the ANN is applied as a universal function approximator in order to estimate induction motor rotor speed driven by a VSI. The used neural network is the multilayer perceptron time delay neural network (TDNN) called by [12] with focused configuration (focused time-lagged feedforward network). For each input quantity are used four samples, in format $x(t), x(t-1), x(t-2), x(t-3), x(t-4)$, where x is one of the quantity at time t.

Two ANN are trained in offline form. Each ANN is trained to act in an region of the scalar control. The proposed estimator in this paper is structured as shown in Fig. 5. The operating range of each ANN and its configurations is presented in Table 1. The first ANN is trained for operation range of 25-60 Hz and the second one of 1-30 Hz. The voltage and current are presented to the ANNs after processing the simulation data, which output is the speed estimated, used in training and validation process. To select which ANN will be used to assess the V_d level, being directly proportional to frequency. To optimize

Table 1. ANN $'s$ Parameters

Network architecture	TDNN Perceptron multilayer
Number of layer	3
Neurons of the 1st hidden layer	6
Neurons of the 2nd hidden layer	21
Training algorithm	Levenberg-Marquardt backpropragation
Learning rate	$5 \cdot 10^{-2}$
Epochs	3000
Square error goal	$1 \cdot 10^{-2}$
Hidden layer activation function	Hyperbolic tangent
Output layer activation function	Linear

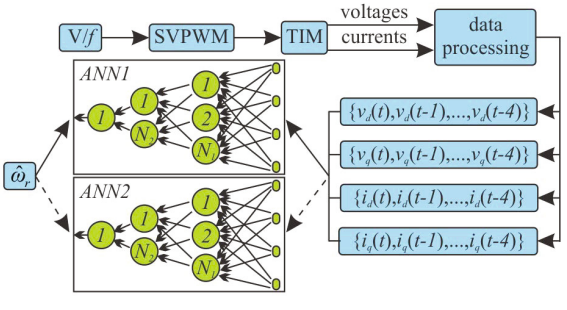


Fig. 5. Training and ANN test structure

the neural network training, a learning reinforcement process was applied in low speed regions. Therefore, data set which represents various aspects of the system dynamic in all operating domain is presented.

5 Generalization Results

The ANNs was trained from computer simulations data of TIM operating under scalar control in several operation points. To validate this approach, the neural speed estimator is tested as speed observer in eight simulations point containing speed and load torque changes, as show in Table 2.

Table 2. TIM Operation Point

Simulation number	Frequency operating point (Hz)		Load torque (N.m)	
	Initial	Final	Initial	Final
1	6		1.1	1.6
2	37		4.4	2.4
3	5	20	0.2	0.4
4	10	25	1.1	2.7

The PI controller implemented according to the following expression:

$$G_c(s) = \frac{(k_p s + k_i)}{s} \tag{1}$$

where k_p is the proportional gain and the integration gain. The PI controller parameters were set to $k_p = 1.8\Omega$ e $k_i = 2.3\Omega/s$. The maximum and minimum slip frequency limiter is 10 Hz and -10 Hz respectively.

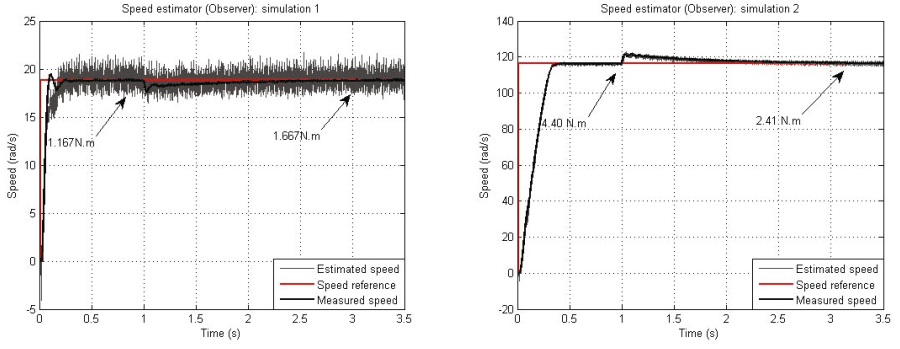
For the simulation it was considered the linear torque load in the following form:

$$T_l = k_t + \alpha \cdot \omega_r \tag{2}$$

where T_l is the load torque, k_t is the initial torque load constant in N.m, α is the constant of relation torque/speed given by N.m/rad/s and ω_r is the TIM rotor speed. In all simulations, it is considered the k_t with a value of 0.1 Nm and a load with inertia moment of $5.42 \cdot 10^{-3}\ kg.m^2$.

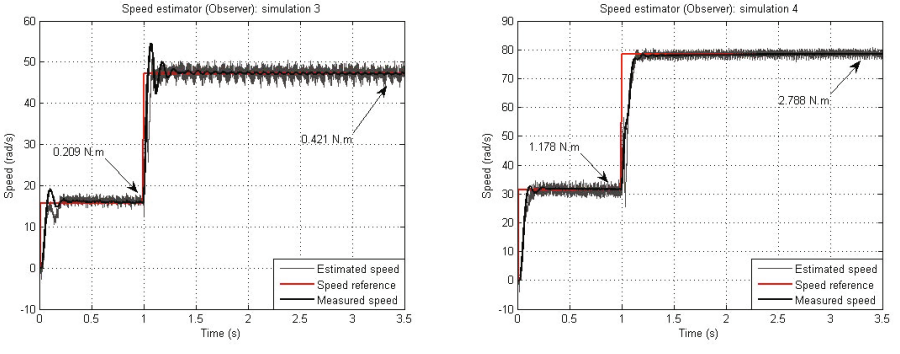
Figure 6 shows the result for simulations 1 and 2, considering a constant speed set point. Still considering load torque variation, now is presented the system behavior with a speed operation point step changing applied for each simulation (3 and 4). Generalization results are shown in Fig. 7. Table 3 shows the Relative Mean Error (RME) between the estimated and measured speed.

Simulations results demonstrate the ability of the ANNs to estimate the TIM speed in the transitory and steady state by scalar control even in low speed region.



a) b)

Fig. 6. Load change simulation results - a) Simulation 1, b) Simulation 2



a) b)

Fig. 7. Speed change simulation results - a) Simulation 3, b) Simulation 4

Table 3. Relative Mean Error of Speed Estimator

Simulation number	Speed Estimator RME (%) as observer
1	3.56
2	0.68
3	3.39
4	2.09

6 Conclusions

This work proposed an alternative methodology to estimate the induction motor speed driven a VSI with scalar control, based on two parallel TDNN artificial neural networks with supervised training process in off-line mode. The ANN selection was made through voltage range operations.

The methodology was applied to estimate the motor speed during transient and steady state, comprising the whole operation range of the scalar control. In each simulation it was applied a load torque variation when the TIM was operating in steady state and speed reference point change with the aim of demonstrate the robustness of the proposed method. Simulation results presented lower mean relative error obtained for all speed operating ranges. Nevertheless, the proposal can be applied to other methods of speed control or parameter estimator of induction motor using the proposed method.

Acknowledgments. The authors gratefully acknowledge the financial support under grand 474290/2008-5, 471825/2009-5, 473576/2011-2 and 552269/2011-5 received from CNPq and process 06/56093-3 received of Araucaria Foundation and CAPES.

References

1. Vas, P.: Sensorless Vector and Direct Torque Control. Oxford University Press (1998)
2. Vasic, V., Vukosavic, S.N., Levi, E.: A stator resistance estimation scheme for speed sensorless rotor flux oriented induction motor drives. IEEE Transactions on Energy Conversion 18(4), 476–483 (2003)
3. Bose, B.K.: Modern Power Electronics and AC Drives. Prentice-Hall, New Jersey (2001)
4. Jevremovic, V.R., Vasic, V., Marcetic, D.P., Jeftenic, B.: Speed-sensorless control of induction motor based on reactive power with rotor time constant identification. Electric Power Applications, IET 4(6), 462–473 (2010)
5. Yoksel, O., Mehmet, D.: Speed estimation of vector controlled squirrel cage asynchronous motor with artificial neural networks. Energy and Manangement 52(1), 675–686 (2011)
6. Gadoue, S., Giaouris, D., Finch, J.: Sensorless control of induction motor drives at very low and zero speeds using neural network flux observers. IEEE Transactions on Industrial Electronics 56(8), 3029–3039 (2009)
7. Goedtel, A., Graciola, C., Silva, S., Nascimento, C., Suetake, M.: A comparative study for single and multilayer neural networks applied to speed estimation in induction motors. In: 2010 XIX International Conference on Electrical Machines (ICEM), pp. 1–6 (September 2010)
8. Ong, C.: Dynamic Simulation of Electric Machinery: Using Matlab/Simulink. Prentice-Hall, Upper Sanddle River (1998)
9. Krause, P.C., Wasynczuk, O., Sudhoff, S.D.: Analysis of Electric Machinery and Drives Systems. Piscataway, New Jersey (2002)
10. Goedtel, A., da Silva, I.N., Serni, P.J.A.: Load torque identification in induction motor using neural networks technique. Electric Power Systems Research 77(1), 35–45 (2007)
11. da Silva, S., Campanhol, L., Goedtel, A., Nascimento, C., Paiao, D.: A comparative analysis of p-pll algorithms for single-phase utility connected systems. In: 13th European Conference on Power Electronics and Applications, EPE 2009, pp. 1–10 (September 2009)
12. Haykin, S.: Neural network: a comprehensive foundation. Prentice Hall, New Jersey (1999)

Opportunistic Sensor Interpretation in a Virtual Smart Environment

José M. Fernández-de-Alba[1], Pablo Campillo[2], Rubén Fuentes-Fernández[1], and Juan Pavón[1]

[1] Facultad de Informática de la Universidad Complutense de Madrid
Avda. Complutense, s/n. 28040 Madrid, Spain
{jmfernandezdealba,ruben,jpavon}@fdi.ucm.es
[2] Facultad de Informática de la Universidad de Murcia
Campus de Espinardo, s/n. 30100 Murcia, Spain
pablocampillo@um.es

Abstract. Smart environments interpret data from sensors to determine users' context. A relevant issue for their development is how to deal with the changes they experience at runtime. Components can be added, change or fail at runtime, modifying the system topology. These changes in turn modify the available information, providing new or redundant data or making impossible to calculate some pieces of the context. Opportunistic planning methods address these issues monitoring the context and suspending or resolving goals when the appropriate conditions are met. Using this approach, a smart room is able to wait and take advantage of changes in the context of the system to achieve goals that it was not able to accomplish before. This paper describes the development and testing of this context-aware behavior using FAERIE, a framework for context management intended to provide a general architecture. The sample application runs over UbikSim, which is a smart environment simulator for testing and validation.

Keywords: opportunistic planning, virtual smart environment, context-awareness, ambient intelligence.

1 Introduction

Ambient Intelligence (AmI) applications use embedded and unobtrusive technologies and devices [11]. Their elements should "vanish" in the environment so the users do not become aware of their existence unless it is completely necessary. For this reason, AmI systems must reduce the configuration effort to a minimum. This also implies being able to reconfigure themselves upon errors and to take advantage of new possibilities to fix the gaps that those errors may have caused. These changes prevent assuming a fixed topology of systems at design time. On the contrary, applications must be designed to be completely adaptive to changes in their components and needs of information.

The opportunistic planning paradigm addresses these requirements by selecting a suitable situation to solve an information goal. When the systems receives

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 109–116, 2012.

a goal, it does not try to solve it immediately, but suspends it. Then, it monitors the context changes to detect a correct moment when it can be solved, e.g. needed resources and information are available [8]. The observation of the context continues during the goal resolution, so the system can suspend the process if the goal becomes unreachable. This behavior implies the existence of mechanisms to facilitate context-awareness: the planning process is able to determine when to suspend or activate goals depending on the change of context conditions.

Several architectures and frameworks already deal with opportunistic features [3,7]. However, they still present several open issues for their extensive use. One of the most relevant is the difficulties to test and validate real applications with them due to their deployment costs. As most of times it is unfeasible to have all the possible environments and devices for a given target platform during development. Thus, testing capabilities are strongly limited. A solution to this problems is simulating most of the involved physical devices and their deployment within physical spaces. This greatly reduces testing and validation costs, and promotes better development practices.

This paper illustrates the development of context-aware systems with an opportunistic behavior using FAERIE (Framework for AmI: Extensible Resources for Intelligent Environments) [1]. Here, the testing of these systems relies to a large extent in the integration with the simulation framework UbikSim [4], thus addressing one of the common limitations of AmI developments. The example application is an artistic installation where the user interacts across several rooms with talking statues.

FAERIE [1] is a general purpose framework for AmI. It offers mechanisms to facilitate the implementation of workflow-based context-aware applications. FAERIE is based on a distributed blackboard model. Sets of *context observer* components observe and update shared *context container*s, which manage parts of the abstract representation of the real context. These *context container*s transparently coordinate among them to offer a virtual globally shared representation of the context. When a *context observer* modifies a piece of the context representation, every other *context observer* interested in that piece of information is made aware of the change, which triggers successive behaviors.

UbikSim [4] is a framework to run simulations of smart environments with different configurations. It allows defining maps of the physical spaces and their sensors. For these sensors, it can provide their signals either as introduced by users or from configuration files and actual external sensors. Engineers can run its simulations in batch (i.e. agents representing users have predefined behaviors), or interactive mode (i.e. agents are controlled via keyboard).

The rest of the paper is organized as follows. Section 2 describes the requirements of the example application. Then, Sections 3 and 4 respectively describe how the application is designed and its behavior in terms of FAERIE interactions. These results are discussed with related work in Sect. 5. Finally, Sect. 6 presents some conclusions and future work.

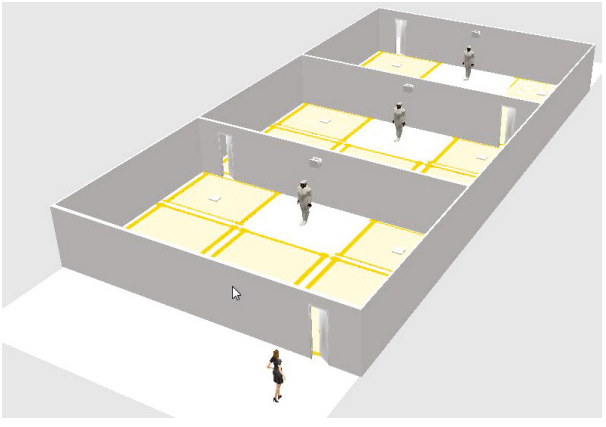


Fig. 1. Ubik Editor 3D model of the Talking Agents scenario

2 Case Study: Artistic Installation

The case study is based on the *Talking Agents* work [2]. This describes an interactive artistic installation where the spectator moves through different rooms. In each room, the system determines the spectator's position in order to trigger interactions with clay statues. These statues are the actual *talking agents*, as they have software and devices for speech recognition and dialog management. For the sake of brevity, this paper only deals with people tracking and considers several types of sensors (other works such as [10] only consider one type of information). Figure 1 shows an instance of this installation using UbikEditor 3D [4], a graphical editor of scenarios for the UbikSim simulator.

The design of the target AmI system needs to consider several requirements:

- Locations in rooms are defined as sectors of their surface. These divisions may change between deployments, and thus the potential users' locations.
- To determine the location, each room contains different sensors. The exact location and kind of sensors are determined at the moment they are deployed, which can occur during runtime. These sensors can be *distance sensors* and *video cameras* situated on the roof. Distance sensors determine location by triangulation, and camera sensors by detecting movement on video signal. The confidence of the recognition on the video signal depends on the lighting of the room.
- In case of redundancy or conflict of the locations calculated by the different alternatives, the system selects the one with the highest confidence score.
- Sensor failures and incomplete deployments may cause blind spots. The application will do its best to determine location at these circumstances.
- During execution, new sensors or alternative mechanisms to determine the location can be added.

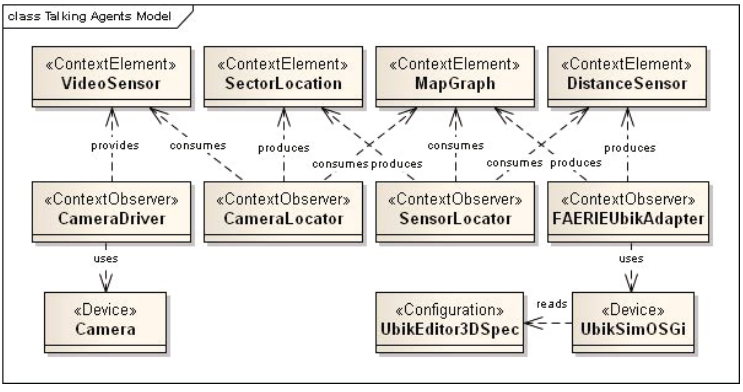


Fig. 2. System structure of the Talking Agents application with FAERIE and UbikSim

3 Design with FAERIE

The design of an application with FAERIE consists of two main tasks: identifying and modeling the structure of the context representation, and defining the rules that determine the context changes as a function of the changes on other context elements or the physical environment. Rules are implemented as *context observer* components that work on the representation of the context as *context elements*, consuming their values in order to provide the values for other elements.

The structure using FAERIE of the Talking Agents application described in the previous section appears in Fig. 2. It includes the following elements:

- **CameraDriver.** It is a context observer that accesses the *Camera* device and provides the *VideoSensor* context element. This element contains two pieces of information (i.e. *attributes*): the input stream of the video captured by the camera and the physical location of the camera provided at the moment of its deployment.
- **CameraLocator.** A context observer that provides the user's location. It uses the physical location of the *VideoSensor*, and superimposes the *MapGraph* on the video from the *VideoSensor* to detect movement.
- **FAERIEUbikAdapter.** It adapts the *UbikSimOSGi* component as an external device. This component simulates the external environment of a system and provides the *MapGraph* and *DisanceSensor* context elements. The *MapGraph* contains an attribute with a directed graph where each node represents a sector in a room of the space. The *DisanceSensor* includes two attributes, one with the current state of the sensor and another with its location, orientation and action range. The data of this last attribute are provided at the moment of its deployment.
- **SensorLocator.** A context observer that triangulates using the information from *DistanceSensor*s on location, orientation, action range and current state, and matches the result with the *MapGraph* to determine the spectator's *Location*.

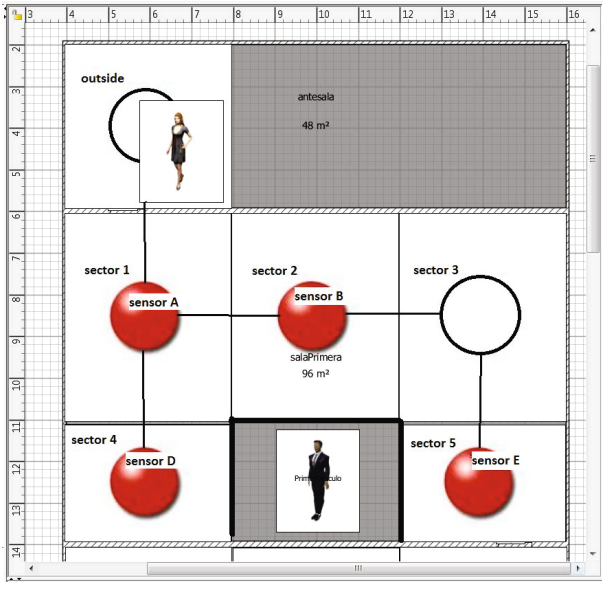


Fig. 3. Map of the first room, indicating the sectors and their adjacency

FAERIE activates only the *context observers* that are currently being used. Neither the *CameraLocator* nor the *SensorLocator* context observers will be activated, and the *context elements* needed by them will not be requested, if there is another context observer already providing the same information. In turn, the *context observers* responsible to calculate those *context elements* will not be activated either. The priority used is the confidence score: a number offered by each context observer to represent the exactitude of their measurement. As an improvement, the *SensorLocator* component only requests the *Sensor* elements necessary to determine the current location and those adjacent to it. In case that the spectator is in a blind spot, every location next to it is observed.

4 Deployment and Testing

This section shows how the proposed implementation of the *Talking Agents* with FAERIE proposed in the previous section works. It discusses the operations performed by the system as reaction to certain events in the first room.

Figure 3 shows the representation of the entrance and the first room. This room contains multiple distance sensors covering the different sectors, as well as a camera sensor in the roof, covering the entire room. White circles represent "blind spots", i.e sectors not covered by the distance sensors. Although in the real installation these are not blind spots at the beginning of its functioning, sometimes sensors may not work properly and the corresponding sectors become blind spots. The black lines represent sector adjacency.

Table 1. Changes in system state as a result of environment events. The asterisk represent which value is selected for location.

Env. event	**CameraLocator**		**CameraDriver**	**SensorLocator**		**UbikAdapter**
	confid.	*location*	*VideoSensors*	*confid.*	*location*	*DistanceSensors*
outside room	0.5	*null	first room	0.6	null	A, B, E
go to sector 1	0.5	null	none	0.6	*sector 1	A, B, D
go to sector 3	0.5	*sector 3	first room	0.6	null	A, B, E
deploy sensor at sector 3	0.5	null	none	0.6	*sector 3	B, C, E
room lighting increased	0.8	*sector 3	first room	0.6	null	none

Table 1 represents the events produced in the system, and how the different *context observers* change the context. The scenario runs as follows:

1. When the person is outside the room, it is considered within a "blind spot". The *SensorLocator* requests every *DistanceSensor* next to a blind spot. Since the *SensorLocator* cannot offer a *Location*, the system delegates on the *CameraLocator*, which requests the *VideoSensor* in first room. As it does not cover the entrance, it does not provides the *Location* neither, and it is left null.
2. The person enters into sector 1. The *SensorLocator* offers then the correct *Location*, and requests only *DistanceSensor*s covering the current and adjacent sectors. The *CameraLocator* is not used, because the *SensorLocator* is already offering a *Location* with a higher confidence score. As a consequence, the *SensorLocator* releases the *VideoSensor*.
3. The person enters into sector 3, which is a blind spot. The *SensorLocator* requests again every *DistanceSensor* next to that blind spot. Since it is not able to offer a *Location*, the system delegates on the *CameraLocator*, which requests again the *VideoSensor*. Now, it offers the correct *Location*.
4. A new sensor is deployed in sector 3, which was previously a blind spot. The *SensorLocator* reconfigures its map and gets a correct *Location* for the person. It renders the *CameraLocator*, and therefore the *VideoSensor* is released again, not needed at the current person's location.
5. The lighting on the room is increased, making the confidence score of the *CameraLocator* higher. Then, the system delegates on the *CameraLocator* for *Location*. It requests again the *VideoSensor* and gets a correct *Location*. Hence, the *SensorLocator* is deactivated, which also releases the *DistanceSensors*, as another component is already offering a more reliable result.

5 Related Work

AmI proposes adaptive and personalized applications. This is intrinsically related with context-awareness, which in turn is greatly supported by the theories of information fusion. The field studies the different methods to aggregate multiple sources of information. These methods are affected by dynamic topologies,

in which the sensor networks are constantly changing. Challa et al.' work [5] proposes new concepts and alternatives to enhance the information fusion methods in order to be opportunistic, and constitutes the starting point of multiple works related with this paper.

The OPPORTUNITY project [9], presents some activity and context recognition systems with opportunistic behaviors. It also presents an architecture focused on the information fusion mechanism, but does not specify the details on context modeling and handling. In FAERIE specification [1], it is defined the way to model context and how it is modified in order to develop a complete context-aware system, which later can use opportunistic features.

Kurz and Ferscha [7] propose a collection of sensor abstractions and mechanisms of self-description for them. Using this, they present an architecture for context and activity recognition that describes sensing goals in an abstract way. The architecture uses these descriptions to facilitate the reconfiguration of systems with the available sensors, producing an opportunistic behavior. This architecture can be extended by replicating the described processes in multiple abstraction layers, as described in the FAERIE framework architecture [1].

Boldrini et al. [3] describe a middleware to share information in opportunistic networks. These are networks where connections among nodes change along time, and the eventual connections are seized to establish information paths. The middleware infers social information of the users to assist the automated learning of the network topology, and to predict future changes. The opportunism in this case is, nonetheless, restricted to the eventual discovery of other nodes, and it does not consider detecting opportunities for a more general context management. This potential enhancement is later discussed in work of the same authors [6] and termed *opportunistic computing*. FAERIE supports this enhancement, as the processes are always triggered by a context change [1], independently of its nature.

6 Conclusions

The paper has described the design for testing of an application with opportunistic evaluation implemented with the FAERIE [1] context-aware framework. The application determines the user's location using different alternatives, chosen at each moment according to available sensors and their confidence level.

The opportunistic behavior is achieved using the context management mechanisms of FAERIE. These mechanisms are designed to reduce resource consumption (power in this case) and promote flexibility, minimizing configuration effort by users. These features produce less obtrusive applications, which is a requirement for AmI.

The paper also proposes the use of the UbikSim [4] simulator for virtual smart environments in order to reduce deployment and test costs. FAERIE integrates this environment by providing wrappers for its sensor components.

The integration of FAERIE and a simulation environment provides the basis to explore relevant extensions of our AmI architecture. A first issue is the

implementation of automated learning mechanisms that reduce the need of providing deployment information to components, therefore simplifying the system configuration. This learning processes frequently need a training phase or redundant sources of information, which can be provided using the simulator. Another improvement is taking advantage of the automatic behaviors present in the UbikSim Kit to validate properties of FAERIE systems in "batch mode".

Acknowledgments. This work has been done in the context of the project "Social Ambient Assisting Living - Methods (SociAAL)", supported by the Spanish Ministry for Economy and Competitiveness, with grants TIN2011-28335-C02-01 and TIN2011-28335-C02-02. Also, we acknowledge support from the "Programa de Creación y Consolidación de Grupos de Investigación" UCM-BSCH GR35/10-A.

References

1. Fernández-de-Alba, J.M., Fuentes-Fernández, R., Pavón, J.: Dynamic Workflow Management for Context-Aware Systems. In: Novais, P., Hallenborg, K., Tapia, D.I., Rodríguez, J.M.C. (eds.) Ambient Intelligence - Software and Applications. AISC, vol. 153, pp. 181–188. Springer, Heidelberg (2012)
2. Fernández-de-Alba, J.M., Pavón, J.: Talking agents: A distributed architecture for interactive artistic installations. Integr. Comput.-Aided Eng. 17(3), 243–259 (2010)
3. Boldrini, C., Conti, M., Delmastro, F., Passarella, A.: Context- and social-aware middleware for opportunistic networks. J. Net. Comp. App. 33(5), 525–541 (2010)
4. Campillo-Sanchez, P., Botia, J.A.: Simulation Based Software Development for Smart Phones. In: Novais, P., Hallenborg, K., Tapia, D.I., Rodríguez, J.M.C. (eds.) Ambient Intelligence - Software and Applications. AISC, vol. 153, pp. 243–250. Springer, Heidelberg (2012)
5. Challa, S., Gulrez, T., Chaczko, Z., Paranesha, T.: Opportunistic information fusion: a new paradigm for next generation networked sensing systems. In: 2005 8th International Conference on Information Fusion, vol. 1, 8 p. (2005)
6. Conti, M., Giordano, S., May, M., Passarella, A.: From opportunistic networks to opportunistic computing. Comm. Mag. 48(9), 126–139 (2010)
7. Kurz, M., Ferscha, A.: Sensor Abstractions for Opportunistic Activity and Context Recognition Systems. In: Lukowicz, P., Kunze, K., Kortuem, G. (eds.) EuroSSC 2010. LNCS, vol. 6446, pp. 135–148. Springer, Heidelberg (2010)
8. Patalano, A.L., Seifert, C.M.: Opportunistic planning: Being reminded of pending goals. Cognitive Psychology 34(1), 1–36 (1997)
9. Roggen, D.e.a.: Opportunity: Towards opportunistic activity and context recognition systems. In: 2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops, pp. 1–6 (2009)
10. Sánchez, A.M., Patricio, M.A., García, J., Molina, J.M.: A context model and reasoning system to improve object tracking in complex scenarios. Expert Systems with Applications 36(8) (2009)
11. Tapia, D.I., Abraham, A., Corchado, J.M., Alonso, R.S.: Agents and ambient intelligence: case studies. J. Ambient Intelligence and Humanized Computing 1(2), 85–93 (2010)

Comparison of PCA and ANOVA for Information Selection of CC and MLO Views in Classification of Mammograms

Ricardo de Souza Jacomini[1], Marcelo Zanchetta do Nascimento[1], Rogério Daniel Dantas[1], and Rodrigo Pereira Ramos[2]

[1] Centro de Matemática, Computação e Cognição, Universidade Federal do ABC (UFABC), Santo André-SP, Brasil
marcelo.nascimento@ufabc.edu.br
[2] Colegiado de Engenharia Elétrica, Universidade Federal do Vale do São Francisco (UNIVASF), Juazeiro, BA, Brasil

Abstract. In this paper, we present a method for extraction and attribute selection for textural features classification using the fusion of information from the mediolateral oblique (MLO) view and craniocaudal (CC) views. In the extraction step, wavelet coefficients together with singular value decomposition technique were applied to reduce the number of textural attributes. For the selection stage and reduction of attributes, an evaluation of the Analysis of Variance (ANOVA) technique and Principal Component Analysis (PCA) is performed when used for textural information reduction. In the final step, it was used the Random Forest algorithm for classifying regions of interest (ROIs) of the set of images determined as normal, benign and malignant. The experiments showed that ANOVA reached the higher proportional attributes reduction and featured the best results for information fusion of CC and MLO views. The best classification rates were obtained with ANOVA for normal-benign images (area under the receiver operating characteristic curve - AUC = 0.78) and benign-malignant images (AUC = 0.83) and with the PCA method for normal-malignant images (AUC = 0.85).

1 Introduction

Breast cancer is the most common type of cancer among women. For the 2012 year, it is expected more than 53,000 cases of this disease in Brazil [1]. The early detection of breast cancer increases treatment options and improves the chances for successful treatment and patient survival. The most effective tool available to clinicians for the early detection of breast cancer is screening mammography [2]. A screening mammographic examination usually consists of four images, corresponding to each breast scanned in two views: mediolateral oblique (MLO) view and craniocaudal (CC) view.During mammographic interpretation, the radiologist combines the information from the two views and evaluates prior examinations to confirm true positives and to reduce false positives [3–5].

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 117–126, 2012.

Computer Aided Diagnosis (CADx) technology can improve the performance of radiologists, by increasing sensitivity to rates comparable to those obtained by double reading, in a cost-effective manner [3, 6]. Algorithms for image processing together with artificial intelligence techniques are used in order to enhance, segment, extract features and classify abnormalities [7, 8]. Although CADx systems have provided a large number of research and high rates of sensitivity using several types of features, the majority of these works analyzes the MLO and CC views independently. In some situations, this system detects abnormalities in only one of the views. Radiologists believe that there is an inconsistency if a particular lesion is similar in both views and the system does not have the capability to find it. Studies have shown that these limitations have changed their impressions and radiologists are ignoring the results provided by these systems [9].

In this work, we present a comparison of PCA and ANOVA to discriminate lesions in mammograms using CC and MLO views merging information from both views. In the extraction step, the wavelet transform method was applied to provide texture-related attributes for the considered images. Following, the singular value decomposition (SVD) technique was used to reduce the number of wavelet coefficients. The PCA and ANOVA techniques were applied to the coefficients in order to select the most relevant attributes in both CC and MLO views. In the next step, it is made the fusion of the attributes obtained from the CC and MLO views. In the final step, it is used the Random Forest classifier to verify the algorithm performance after application of the feature selection techniques.The proposed computational tools were evaluated on a set of cases selected from the public Digital Database for Screening Mammography.

2 Materials and Methods

Data Set: the database used in this work was taken from the Digital Database for Screening Mammography (DDSM)[8]. The DDSM project is a joint effort of researchers from the Massachusetts General Hospital (D.Kopans, R Moore), the University of South Florida (K. Bowyer), and the Sandia National Laboratories – EUA (P. Kegelmeyer). We selected images digitized with a Lumisys laser film scanner at 50 mm and a resolution of 12 bits. In order to validate our methodology, we used a sub-sample of 720 ROIs from the mammogram images, composed of 480 samples of abnormal tissues, among which 240 were malignant and 240 were benign. Each set of images from a given type is divided in 50 % CC and 50 % MLO views. Images with no masses were taken from each of the images selecting a region with normal breast tissues. The selection of ROI with a nodule was implemented based on the .ics file from the DDSM database that defines the chain code of the lesion edge. The images used in the experiments were cuttings of size 512×512 pixels done in the sub-image, whose centers correspond to the centers of the presented abnormalities.

2.1 Textural Features Extraction

Wavelet Transforms: For a given function $f(x)$ in $\mathbb{R}$, the wavelet transform $Wf(a,b)$ on $\mathbb{R}$ is obtained from the inner product of $f(x)$ with a wavelet family, i.e.:

$$Wf(a,b) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} f(x)\psi_{a,b}(x)dx \tag{1}$$

where

$$\psi_{a,b}(x) = a^{-1/2}\psi\left(\frac{x-b}{a}\right) \tag{2}$$

are wavelets obtained from scaling and time shifting operations on a mother wavelet $\psi(x)$, with a and b being the scaling and translation factors, respectively. The mother wavelet $\psi(x)$ is an integrable function with zero mean, i.e.:

$$\int_{-\infty}^{\infty} \psi(x)dx = 0 \tag{3}$$

and centered in the neighborhood of $x = 0$. Examples of wavelets can be found on reference [11].

The discrete wavelet transform (DWT) is obtained from the discretization of parameters a and b. In [11], Mallat proved that a DWT of a signal is equivalent to its decomposition on a series of highpass and lowpass filter banks, followed by a downsampling of two samples, one bank for each desired resolution. The lowpass filters outputs give the approximation coefficients and the highpass filters outputs the details. For an image that is represented in 2D, the image is passed through a pair of filters on each row, followed by a downsampling of 2. Then, results are used as inputs of two filter banks, which are applied at the columns of the image, followed by downsampling. Four sub-images are generated in this process: the approximation LL, which represents the original image with a smaller resolution, and the details LH, HL, HH, which represents the horizontal, vertical and diagonal directions, respectively.

For each ROI, the 2D-DWT was applied using three different wavelet functions, namely Biorthogonal 3.7, Daubechies 8 and Symlet 8, and 2 resolution levels. The decomposition levels and wavelet functions were selected based on previous works [12, 13]. The first decomposition level yields the coefficient matrices LL_1, LH_1, HL_1 and HH_1. The second decomposition level applied in sub-band LL_1 resulted in the coefficient matrices LL_2, LH_2, HL_2 and HH_2.

Nonlinearity and smoothing operators: In this method, the first step consists of the introduction of non-linearities and smoothing operators, to make the subband images less sensitive to local variations [14]. For nonlinearity and smoothing operations, the total energy of wavelet transformation coefficients in each sub-band was calculated using:

$$E_i = \frac{1}{PQ}\sum_{j=1}^{P}\sum_{k=1}^{Q}|w_i(j,k)|^2 \tag{4}$$

where E_i is the overall energy in the i-th sub-band, $w_i(j,k)$ is the wavelet transformation coefficient at locations (j,k) in the i-th sub-band, and P and Q are the number of rows and columns of the i-th sub-band, respectively. Here, parameters P and Q have similar values. In this work, we consider also 3×3 neighborhoods, compute their average, and then normalize the average, as proposed by [14]. At each location $L_i(j,k)$, the local energy was computed in a 3×3 neighborhood using the following equation:

$$L_i(j,k) = \frac{1}{9} \sum_{u=0}^{2} \sum_{v=0}^{2} |w_i(j-1+u, k-1+v)|^2 \tag{5}$$

where the local energies $L_i(j,k)$ were normalized by

$$I_i(j,k) = \frac{L_i(j,k)}{E_i} \tag{6}$$

This process produced 196,608 wavelet coefficients for the first decomposition (256×256 in 3 sub-images) and 49,152 coefficients for the second decomposition (128×128 in 3 sub-images). Gathering together all detail coefficients from the sub-bands resulted on a feature vector of 245,760 attributes. Fig. 1 represents the process of wavelet decomposition for each ROI to obtain the wavelet coefficient features.

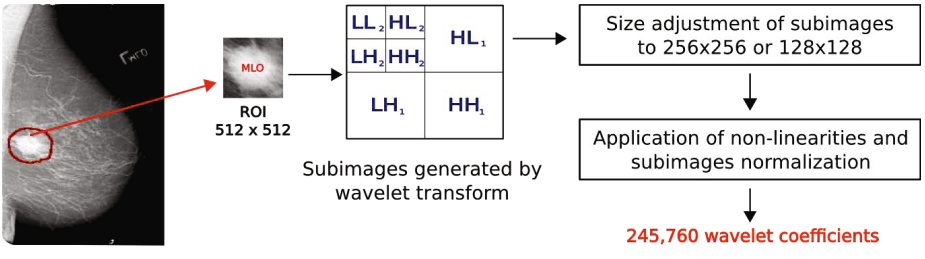


Fig. 1. Illustration of wavelet decomposition of a mammographic image

Singular Value Decomposition: Singular value decomposition (SVD) method was applied to each wavelet subimage, leading to a cardinality reduction of the feature vector [14, 16, 17]. To explain the application of method, consider the matrix $\mathbf{I_i}$ with size $P \times Q$, whose entries are the sub-band wavelet coefficients after the introduction of nonlinearity. The application of the SVD based method decomposes each wavelet subimage $\mathbf{I}i$ into the product of three matrices given by:

$$\mathbf{I}_i = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^T \tag{7}$$

where $\mathbf{U}_i$, with size $P \times Q$, and $\mathbf{V}_i$, with size $Q \times Q$, are orthogonal matrices whose columns are the eigenvectors of matrices $\mathbf{I}_i\mathbf{I}_i^T$ and $\mathbf{I}_i^T\mathbf{I}_i$, respectively, and $\mathbf{S}_i$, with size $Q \times Q$, is a diagonal matrix whose non-zero entries are the singular

values (square roots of the eigenvalues) of matrix $\mathbf{I}_i\mathbf{I}_i^T$. Once the SVD is unique for each matrix, the singular values completely represents the sub-band images. A method of truncation of the lower singular values, which is equivalent to a filter based approach, was applied to matrix $\mathbf{S}_i$ for dimensionality reduction with images with noise. In [14], the authors have shown that the effect of noise is more intense on singular values with lower magnitudes. Therefore, the diagonal matrix can be truncated to a dimension $K \times K$, where K is given empirically by:

$$K = Q\frac{\sigma_1^2}{\sum_n \sigma_n^2} \tag{8}$$

where σ_n is the n-th singular value and σ_1 is its highest value.

Since the wavelet transform in this work was considered for 2 resolution levels, there are sub-band images with sizes 256×256 and 128×128 pixels, what leads to a different number of truncated singular values (different K) for each mammogram. Therefore, Equation (8) was used to get a value of K_r for each resolution level, and the overall K was obtained by averaging the number of truncated singular values obtained. Having defined the average value K, singular values were extracted from each of the eight wavelet sub-images, resulting in a feature vector of $6K$ elements representing texture characteristics of the original mammograms.

2.2 Feature Selection

Analysis of Variance - ANOVA: is a statistical model that compares the averages of two or more experiments. This evaluation is performed to the extent that the differences between means are significant for the comparison of two estimates [18].

For our experiments, this comparison was carried out on benign masses, malignant masses and normal tissue present on mammographic images. The mean and standard deviation were calculated for sets of texture data and ANOVA technique was applied on each of the following cases: comparison of benign versus malignant masses, called benign-malignant; comparison of normal tissue versus benign masses, called normal-benign; and comparison of normal tissue versus malignant masses, called normal-malignant.

The one-way ANOVA method was firstly applied on the singular values attributes generated by the SVD procedure to determine the statistical significance of these values. In this method, the null hypothesis is that all attribute means are the same. The alternative hypothesis is that at least two of them are different. If any two groups are statistically the same, both are discarded since they do not contribute to the classification step [15, 19].

An F-test is applied to generate this test with a confidence interval of $1 * 10^{-11} < p < 1 * 10^{-16}$, this interval was defined according to the minimum number of textural attributes greater than or equal to 12. The whole process was performed using Matlab.

Principal Component Analysis -PCA: is a powerful technique for extracting a structure from potentially high-dimensional data sets, which corresponds to extracting the q eigenvectors that are associated with the largest q eigenvalues from the input distribution. Starting from an original set of l samples (features), which form the elements of a vector $\mathbf{x} \in \mathbb{R}^{\mathbf{l}}$, the goal is to apply a linear transformation to obtain a new set of samples [20]:

$$\mathbf{y} = \mathbf{A}^{\mathbf{T}}\mathbf{x} \tag{9}$$

so that the components of $\mathbf{y}$ are uncorrelated. In a second stage, one chooses the most significant of these components. The process is summarized here:
1. Estimate the covariance matrix $\mathbf{C}$. Usually, the mean value is assumed to be zero, $E[\mathbf{x}] = \mathbf{0}$. In this case, the covariance and autocorrelation matrices coincide, $\mathbf{R} \triangleq E[\mathbf{x}(\mathbf{x}^T)] = \mathbf{C}$. If this is not the case, the mean is subtracted. Given N feature vectors, $\mathbf{x}_i \in \mathbb{R}^l$, $i = 1, 2, ..., N$, the autocorrelation matrix estimate is given by

$$R \approx \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T \tag{10}$$

2. Perform the eigendecomposition of S and compute the l eigenvalues/ eigenvectors, λ_i, $\mathbf{a}_i \in \mathbb{R}^l$ $i = 0, 1, 2, ..., l-1$. **3.** Arrange the eigenvalues in descendind order, $\lambda_0 \geq, \lambda_1 \geq ... \geq \lambda_{l-1}$. **4.** Choose the m largest eigenvalues. In general, m is chosen so that the gap between λ_{m-1} and λ_m is large. Eigenvalues $\lambda_0 \geq, \lambda_1 \geq ... \geq \lambda_{m-1}$ are known as the m principal components. **5.** Use the respective (column) eigenvectores $\mathbf{a}_i$, $i = 0, 1, 2..., m-1$ to form the transformation matrix

$$\mathbf{A} = [\mathbf{a}_0\ \mathbf{a}_1\ \mathbf{a}_2 \ldots \mathbf{a}_{m-1}] \tag{11}$$

6. Transform each l-dimensional vector $\mathbf{x}$ in the original space to an m-dimensional vector $\mathbf{y}$ according to equation (9). Thus, the total variance of the elements of $\mathbf{x}$ is:

$$\sum_{i=0}^{l-1} E[x^2(i)] \tag{12}$$

and is equal to the sum of the eigenvalues

$$\sum_{i=0}^{l-1} \lambda_i \tag{13}$$

After the transformation, the variance of the ith elements, $E[y^2(i)]$, $i = 0$, 1, 2,..., $l - 1$, is equal to λ_i. Thus, selection of the elements that correspond to the m largest eigenvalues retains the maximum variance.

The PCA technique was applied in the SVD coefficients obtained from the ROIs wavelet transform using the WEKA (Waikato Environment for Knowledge Analysis) platform [23] for the following configuration: percentage of the variance in the original data 0.95 (95%), with the Ranker search method [21].

2.3 Fusion of Informations

Several studies have demonstrated that the fusion of information extracted from two views, CC and MLO, allows for a false positive reduction compared to the use of a single view [9, 22]. For the process of fusion of information, we applied the ANOVA and PCA technique to analyze the characteristics that are relevant in both views, at the same time. Then, for each mass, a new set of textural features was arranged with the features extracted from both mass views, and the whole processes of feature extraction and selection was performed and used in the next step for classification.

2.4 Classification Stage

In this paper, we have chosen to classify images using the Random Forest algorithm with two classes. The Random Forest algorithm is built from a collection of classification trees. It is a concept of regression trees, bootstrap samples induced by a set of training data, using features selected in the random process of tree induction [24]. The objects were classified from the data set in one of the three comparisons, namely normal or benign tissues, normal or malignant tissues and benign or malignant tissues. These algorithms were implemented in WEKA software. To train and test the proposed computerized method, a cross validation procedure was performed on a dataset To obtain the performance for each classification, we implemented a 10-fold cross validation procedure, in which the dataset were split into N parts. From these, $N-1$ parts served as the training data to fit the classification model. The remaining part was used as the test data for the estimation of performance measures. Each of the N parts was used as test data in turn. The resulting N estimates were averaged to obtain the final expected value. Performance evaluation was accomplished by means of the area under the ROC curve (AUC).

3 Experimental Results

Table 1 shows the obtained results for AUC when using Random Forest classifier for mammographic images obtained from the CC and MLO views. The testing result using the Random Forest classifier on images obtained from fusion of informations of the CC and MLO views is given in Table 2.

4 Discussions and Conclusion

In our proposed methods for classifying masses in mammograms, multiple features are obtained using the fusion of information from the CC and MLO views. Comparing the results of Tables 1, we verify that Symlet 8 and Daubechies 8 wavelets performed better for ANOVA (AUC = 0.83) and PCA (AUC = 0.82) techniques, respectively, for normal-malignant tissues in CC view. In MLO view,

Table 1. Classification of mammograms from CC and MLO views with Random Forest

		Normal-Benign		Normal-Malignant		Benign-Malignant	
		AUC	**Std**	**AUC**	**Std**	**AUC**	**Std**
CC							
ANOVA	**db08**	0.67	0.19	0.81	0.12	0.75	0.17
	sym8	0.69	0.21	0.83	0.11	0.74	0.17
	bo37	0.74	0.16	0.82	0.12	0.76	0.16
PCA	**db08**	0.67	0.20	0.82	0.10	0.80	0.15
	sym8	0.67	0.21	0.73	0.17	0.78	0.15
	bo37	0.71	0.18	0.81	0.14	0.76	0.14
MLO							
ANOVA	**db08**	0.75	0.15	0.78	0.14	0.71	0.16
	sym8	0.69	0.19	0.81	0.13	0.79	0.13
	bo37	0.70	0.20	0.82	0.11	0.79	0.14
PCA	**db08**	0.72	0.18	0.81	0.15	0.75	0.16
	sym8	0.70	0.18	0.82	0.12	0.82	0.12
	bo37	0.64	0.23	0.81	0.13	0.79	0.14

Table 2. Classification of mammograms from information fusion of CC and MLO views with Random Forest

CC+MLO		Normal-Benign		Normal-Malignant		Benign-Malignant	
		AUC	**Std**	**AUC**	**Std**	**AUC**	**Std**
ANOVA	**db08**	0.78	0.14	0.84	0.12	0.79	0.13
	sym8	0.74	0.16	0.84	0.13	0.83	0.10
	bo37	0.76	0.19	0.82	0.12	0.82	0.13
PCA	**db08**	0.71	0.20	0.85	0.11	0.78	0.16
	sym8	0.69	0.18	0.81	0.14	0.79	0.13
	bo37	0.74	0.16	0.84	0.10	0.80	0.13

the use of the ANOVA technique with the Daubechies 8 mother wavelet performed better with AUC = 0.75 for normal-benign tissues. However, results of discriminating breast tissue patterns from benign and malignant masses show that Symlet 8 wavelet with PCA method was 3% superior than ANOVA.

As seen on Table 2, we note that the use of information, obtained from the fusion of the CC and MLO views, might raise the evaluated rates of the descriptors. These rates were higher for ANOVA to all considered comparisons, being higher for almost all kinds of joint lesions for PCA. A similar behaviour occurred to the other other wavelets. This is a typical early case study setting and it gives useful indication. The performance evaluation conducted over the images showed that the best classification rates were obtained for ANOVA method with normal-benign tissues (AUC=0.78) and benign-malignant tissues (AUC=0.83),

while PCA method showed the best classification rates (AUC=0.85) comparing normal-malignant tissues. In future experiments more feature descriptors and classifiers should be investigated for the problem.

References

1. Brasil, Ministerio da Saude, Instituto Nacional do Cancer (INCA): Estimativa 2012: incidencia de cancer no brasil (2011)
2. Gupta, S., Markey, M.: Correspondence in texture features between two mammographic views. Medical Physics 32, 1598 (2005)
3. Rangayyan, R., Ayres, F., Leo Desautels, J.: A review of computer-aided diagnosis of breast cancer: Toward the detection of subtle signs. Journal of the Franklin Institute 344, 312–348 (2007)
4. Velikova, M., Samulski, M., Lucas, P., Karssemeijer, N.: Improved mammographic cad performance using multi-view information: a bayesian network framework. Physics in Medicine and Biology 54, 1131 (2009)
5. Wei, J., Chan, H., Sahiner, B., Zhou, C., Hadjiiski, L., Roubidoux, M., Helvie, M.: Computer-aided detection of breast masses on mammograms: Dual system approach with two-view analysis. Medical Physics 36, 4451 (2009)
6. Elter, M., Horsch, A.: Cadx of mammographic masses and clustered microcalcifications: a review. Medical Physics 36, 2052 (2009)
7. Jiang, J., Yao, B., Wason, A.: A genetic algorithm design for microcalcification detection and classification in digital mammograms. Computerized Medical Imaging and Graphics 31, 49–61 (2007)
8. Balleyguier, C., Ayadi, S., Van Nguyen, K., Vanel, D., Dromain, C., Sigal, R.: Birads (tm) classification in mammography. European Journal of Radiology 61, 192–194 (2007)
9. Gupta, S., Chyn, P.F., Markey, M.K.: Breast cancer cadx based on bi-rads descriptors from two mammographic views. Med. Phys. 33, 1810–1817 (2006)
10. Mallat, S.: Wavelets for a vision. Proceedings of the IEEE 84, 604–614 (1996)
11. Mallat, S.: A wavelet tour of signal processing. Academic Press (1998)
12. Eltoukhy, M.M., Faye, I., Samir, B.B.: Curvelet based feature extraction method for breast cancer diagnosis in digital mammogram. In: Proc. Int Intelligent and Advanced Systems (ICIAS) Conf., pp. 1–5 (2010)
13. Rashed, E., Ismail, I., Zaki, S.: Multiresolution mammogram analysis in multilevel decomposition. Pattern Recognition Letters 28, 286–292 (2007)
14. Selvan, S., Ramakrishnan, S.: Svd-based modeling for image texture classification using wavelet transformation. IEEE Transactions on Image Processing 16, 2688–2696 (2007)
15. Pedrini, H., Schwartz, W.: Análise de imagens digitais: princípios, algoritmos e aplicações. Thomson Learning, São Paulo (2008)
16. Ramakrishnan, S., Selvan, S.: Image texture classification using wavelet based curve fitting and probabilistic neural network. International Journal of Imaging Systems and Technology 17, 266–275 (2007)
17. Ramakrishnan, S., Selvan, S.: Multiwavelets domain singular value features for image texture classification. Journal of Zhejiang University-Science A 8, 538–549 (2007)
18. Vieira, S.: Análise de variância: ANOVA. Atlas (2006)

19. Susomboon, R., Raicu, D., Furst, J., Johnson, T.: A co-occurrence texture semi-invariance to direction, distance and patient size. In: Proc. SPIE Medical Imaging, vol. 6914. Citeseer (2008)
20. Theodoridis, S., Koutroumbas, K., Pikrakis, A., Cavouras, D.: Introduction to pattern recognition: a matlab approach. Academic Pr. (2009)
21. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2011)
22. Paquerault, S., Petrick, N., Chan, H., Sahiner, B., Helvie, M.: Improvement of computerized mass detection on mammograms: Fusion of two-view information. Medical Physics 29, 238 (2002)
23. Vibha, L., Harshavardhan, G., Pranaw, K., Shenoy, P., Venugopal, K., Patnaik, L.: Classification of mammograms using decision trees. In: 10th International Database Engineering and Applications Symposium, IDEAS 2006, pp. 263–266. IEEE (2006)
24. Ramos, R., Nascimento, M.: Comparação de extratores de características em imagens mamográficas. In: XXI Congresso Brasileiro de Engenharia Biomédica (2008)

A Multiobjective Analysis of Adaptive Clustering Algorithms for the Definition of RBF Neural Network Centers in Regression Problems

Rosana Veroneze, André R. Gonçalves, and Fernando J. Von Zuben

School of Electrical and Computer Engineering
University of Campinas
Campinas, São Paulo, Brazil
{veroneze,andreric,vonzuben}@dca.fee.unicamp.br

Abstract. A variety of clustering algorithms have been applied to determine the internal structure of Radial Basis Function Neural Networks (RBFNNs). k-means algorithm is one of the most common choice for this task, although, like many other clustering algorithms, it needs to receive the number of prototypes a priori. This is a nontrivial procedure, mainly for real-world applications. An alternative is to use algorithms that automatically determine the number of prototypes. In this paper, we performed a multiobjective analysis involving three of these algorithms, which are: Adaptive Radius Immune Algorithm (ARIA), Affinity Propagation (AP), and Growing Neural Gas (GNG). For each one, the parameters that most influence the resulting number of prototypes composed the decision space, while the RBFNN RMSE and the number of prototypes formed the objective space. The experiments found that ARIA solutions achieved the best results for the multiobjective metrics adopted in this paper.

Keywords: Radial Basis Function Neural Network, Adaptive Clustering Algorithms, Regression Problems.

1 Introduction

Radial Basis Function Neural Networks (RBFNNs) are universal approximators and have been successfully applied to deal with a wide range of problems. The main concept in this approach is to represent the function to be approximated by a linear combination of radial basis functions (RBFs).

RBFNNs can be trained by either a full or a quick learning scheme. In the former, nonlinear optimization algorithms (e.g. gradient-descent-based) are used to determine the whole set of parameters of an RBFNN: center and dispersion of each RBF, and the weights of the output layer. In this case, the number of RBFs is either defined a priori or estimated by a trial-and-error procedure. As for the quick learning scheme, the internal structure of an RBFNN (the number of RBFs, their centers and dispersions) is given a priori and the weights of the output layer can be determined by the *Least Squared* method.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 127–134, 2012.

Most of the proposed approaches to determine a priori the RBFNN internal structure are based on clustering methods [1]. A clustering algorithm widely used for this purpose is k-means [2]. Like many other clustering algorithms, k-means needs to receive the number of prototypes. However, in real-world applications little or no information is available concerning the input distribution, so it is hard to determine a priori the number of prototypes. Some proposals, like Cross-Validation [3] and MC^{2}SG [3], use regular clustering algorithms (usually k-means) and test several values for the number of prototypes. The computational cost of this approach increases with the number of values to be tested. An option to avoid these drawbacks is to use adaptive clustering algorithms that determine the number of prototypes automatically. In this paper, three of these algorithms were compared on the task of defining the internal structure of RBFNNs applied to regression problems, more specifically: (i) Adaptive Radius Immune Algorithm (ARIA) [4], (ii) Affinity Propagation (AP) [5], and (iii) Growing Neural Gas (GNG) [6].

In regression problems, a good RBFNN internal structure definition is parsimonious, i.e., leads to a low root mean squared error (RMSE) using the smallest possible number of prototypes. The smaller the number of prototypes, the lower the RBFNN computational cost, which is very important, particularly in real-world problems as on-line applications. Therefore, we have two conflicting objectives: low RMSE and low number of prototypes.

Despite ARIA, AP, and GNG define the number of prototypes automatically, they have some parameters that influence directly this number. In this way, to do a comprehensive and fair comparison, we performed a multiobjective analysis using the well-known NSGA-II algorithm [7], and compared the performance of them in terms of their Pareto front. The parameters of each algorithm that exert more influence on the final number of prototypes formed the decision space. The RMSE value and the number of centers formed the objective space. Thus, we will not provide a single solution to the user, but a set of non-dominated solutions. Therefore, the user can choose the best solution to his scenario, considering, for example, his computer resource.

The remaining of this paper is organized as follows. Section 2 presents a general explanation of RBFNN. Section 3 describes the three clustering algorithms compared in this paper. A detailed description of the experiments carried out in this study is given in Section 4. The obtained results are presented and discussed in Section 5. The concluding remarks of this study are outlined in Section 6.

2 Radial Basis Function Neural Networks

The architecture of an RBFNN is composed of an input layer, a hidden layer, and an output layer. The number of neurons in the input layer is equal to the number of attributes. The hidden layer is composed of an arbitrary number of RBFs (e.g. Gaussian RBFs), being each one defined by a center position and a dispersion parameter. The output layer is formed by neurons that promote a linear combination of the activations of the hidden layer neurons.

For a p-dimensional input vector $\mathbf{x} = (x_1, x_2, \ldots, x_p)$, where $\mathbf{x} \in \mathbf{X} \subset \Re^p$, the output of an RBFNN may be calculated as follows

$$y = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \tag{1}$$

where $\mathbf{w} = [w_1, \ldots, w_m]^T$ are the network weights and $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_m]^T$ contains the *basis functions*. Given the RBF center $\mathbf{c}_j \in \Re^p$, and the dispersion $\rho_j \in \Re$, the output of each basis function is

$$\phi_j = G(\|\mathbf{x} - \mathbf{c}_j\|, \rho_j), \quad j = 1, \ldots, m, \tag{2}$$

where $G(\cdot)$ is the RBF and $\|\cdot\|$ is a norm defined in the input space. In this work, Gaussian functions were used as the RBFs.

If the centers and dispersions of Eq. 2 are known, the RBFNN training becomes a minimization problem which may be solved by *least square* closed-form solution. Thus, the optimal set of weights at the output layer is given by

$$\mathbf{w}^* = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{s}, \tag{3}$$

where $\mathbf{H}$ is an $n \times m$ matrix, containing the outputs of the m hidden neurons (RBFs) for each of the n input data points, and $\mathbf{s}$ is the vector of desired outputs.

3 Clustering Algorithms

This section describes the three adaptive clustering algorithms used in this work: ARIA, AP, and GNG.

ARIA [4] is an immune inspired clustering algorithm which uses mechanisms of affinity maturation, clonal expansion, and network suppression to automatically define the position of a reduced number of prototypes. Associated to each prototype there is a radius that is inversely proportional to the local density of data points represented by the prototype, and is directly proportional to a parameter called ts. Once ts is related to the magnitude of the radii that will be generated, it will dictate the final number of prototypes. In this work, we computed the density and the radius of a prototype based on [8].

AP [5] is an exemplar-based clustering method that, given a set of similarities between pairs of data points, exchanges messages between them so as to determine a subset of exemplar points that best represent the data. The prior information about how likely each point is to be chosen as exemplar is a parameter called *preference*. At the end, exemplars are those data points for which the sum of their availabilities and *preferences* are positive. So, the final number of exemplars depend on the choice of this parameter.

GNG [6] is an unsupervised incremental clustering algorithm that uses competitive Hebbian learning [9] to direct the local adaptation of nodes and insertion of new nodes. It creates a graph of nodes, where each node k is associated with: (i) a prototype; (ii) a local cumulative error variable; and (iii) a set of edges defining its topological neighbours. To each edge, there is an associated age. An edge is removed if its age is greater than a_{max}. If a node has no edges, it is

removed. The value of the local cumulative error variables are used to indicate where a new node must be inserted. The number of nodes can not be greater than max_nodes. Although there are another parameters (see [6]), max_nodes and a_{max} are those that influence the most in the resulting number of prototypes.

4 Experimental Evaluation

In the following sections, we describe the experiments and multiobjective analysis carried out with the purpose of investigating how parcimonious each clustering algorithm is in defining the RBFNN internal structure for regression problems.

4.1 Experimental Settings

Six datasets were considered in the experiments, being collected from UCI [10] and StatLib [11] dataset libraries. Table 1 presents some information about these datasets. For the experiments, all datasets were normalized to avoid problems with attributes in different scales. The RBFNN output values is denormalized for the RMSE calculation.

Table 1. Datasets description

Dataset	# of features	# of samples
Auto-mpg	7	398
Bodyfat	14	252
Servo	4	167
Housing	13	506
Pollution	15	60
N_{O2}	7	500

A ten-fold cross-validation method was employed to evaluate the RBFNN efficiency. So, the fitness of each individual in the NSGA-II algorithm was the result of one execution of the ten-fold cross-validation. For NSGA-II, the population size was 30 and the maximum number of iterations was 100.

The interval of admissible values of each parameter was chosen so that the algorithm was able to yield solutions with different number of prototypes within the range $[1, N \cdot 0.5]$, where N is the number of samples in the dataset. The intervals of each parameter was defined as follows. For GNG, $a_{max} \in [1, 50]$ and $max\_node \in [N \cdot 0.05, N \cdot 0.5]$. For ARIA, $ts \in [0.2, 5]$. Finally, for AP, $preference \in [min(S), max(S)]$, where S is the similarity matrix.

The other parameters of the adaptive clustering algorithms, that exert less influence on the final number of centers, was set as follows. For ARIA, mutation rate $\mu = 1$, decay rate $\gamma = 0.9$ and neighborhood size $N_s = 3$. For a complete description of the whole set of ARIA parameters, refer to [4]. In GNG (see

parameters description in [6]), $\epsilon_b = 0.01$, $\epsilon_n = 0.004$, $\alpha = 0.5$, $d = 0.995$, $max\_it = 200$ and $\lambda = (N \cdot max\_it)/max\_nodes$. Thus, the number of neurons inserted will be equal to max_nodes. For AP, the negative of the Euclidean distance was used as similarity measure, damping factor $\lambda = 0.6$, convergence condition $n_{conv} = 50$ and $max\_it = 2000$ (see parameter description in [5]).

The results were obtained over 20 independent runs of NSGA-II.

Dipersions Definition: An important decision when projecting RBFNNs, besides the choice of the number and location of the RBFs, is the definition of their dispersions. Although they can be distinct (and even tunable) for each RBF, usually the same value is assigned for all, given by $\rho = d_{max}/\sqrt{2k}$ [12], where d_{max} is the largest distance among the prototypes and k is the number of prototypes. For ARIA, the adaptive radius obtained by each prototype can be used to calculate its dispersion. Using this information, the dispersion of the i-th RBF is $\rho_i = \eta \cdot r_i$, where η is a multiplicative factor. It was observed that η equals three has led to better results. So, to assess the efficiency of this approach, we will compare ARIA using radius based dispersion and ARIA employing fixed dispersion, named here as ARIA_d.

5 Results and Discussion

To compare the obtained results, we adopted the *Wilcoxon's Rank Sum test* [13] with significance 0.05. Since ARIA appeared to be the most promising approach (among the others analyzed here) to this problem, we chose it as the central algorithm in our analysis.

Table 2 shows the mean and standard deviation of the *hypervolume* (Hyp.), and *maximum spread* (MS) metrics [14]. The RMSE and the number of prototypes were rescaled to the interval [0,1] to compute the hypervolume, so it was limited in the same interval. The values in bold indicate the best results. We compared ARIA results with those of the other algorithms, and used the asterisks (*) to denote when the null hypothesis of Wilcoxon's test was rejected.

It is clear the superiority of ARIA in terms of the hypervolume metric. For all datasets, ARIA reached the highest values, while GNG achieved the worst hypervolume values, except for N_{O2} and *Pollution* datasets. In terms of maximum spread, ARIA also produced the best results for the majority of datasets.

Table 3 shows the mean and standard deviation of the *two-set coverage*, $C(A, B)$, metric [14]. Since it is not symmetric, we show the results of $C(A, B)$ and $C(B, A)$. For the Wilcoxon test, we show the p-values and the symbols "+", "–", or "∼" indicating when ARIA achieved significantly higher, lower or equivalent results than the other algorithms, respectively.

It is possible to see in Table 3 that ARIA is the algorithm that led to a set of final non-dominated solutions closer to the Pareto front of the problem, once its solutions covered (i.e. dominated) the solutions of the other algorithms for the majority of datasets.

Table 2. Mean and standard deviation of the *hypervolume* and *maximum spread* metrics

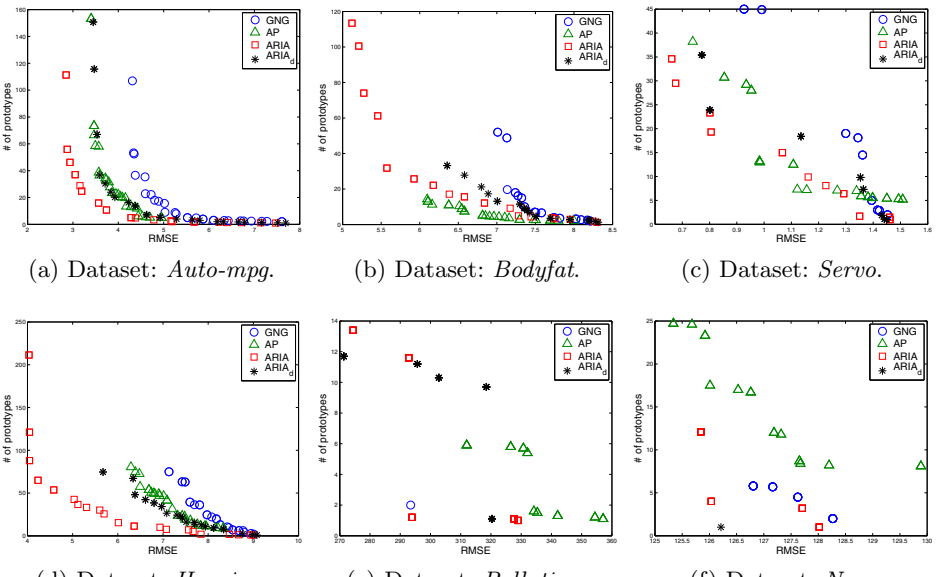
Dataset	ARIA		ARIA_d		GNG		AP	
	Hyp.	MS	Hyp.	MS	Hyp.	MS	Hyp.	MS
Auto-mpg	**0.91** (±0.02)	**88.00** (±25.64)	0.78* (±0.03)	**75.88** (±33.06)	0.63* (±0.02)	69.45* (±21.96)	0.79* (±0.02)	**88.38** (±34.20)
Bodyfat	**0.77** (±0.02)	**109.68** (±7.32)	0.52* (±0.02)	49.17* (±13.69)	0.35* (±0.02)	42.51* (±25.05)	0.62* (±0.03)	12.91* (±1.80)
Servo	**0.85** (±0.08)	43.79 (±16.19)	0.64* (±0.08)	**96.69*** (±49.15)	0.37* (±0.12)	32.96 (±14.23)	0.66* (±0.07)	63.64 (±45.80)
Housing	**0.81** (±0.04)	**134.32** (±38.70)	0.48* (±0.04)	95.83* (±18.13)	0.28* (±0.02)	75.79* (±13.73)	0.43* (±0.03)	79.98* (±21.52)
Pollution	**0.62** (±0.08)	**55.99** (±12.62)	**0.62** (±0.07)	45.45* (±13.34)	0.42* (±0.13)	12.83* (±9.85)	0.31* (±0.12)	28.84* (±12.87)
N_{O2}	**0.79** (±0.10)	**23.57** (±13.65)	0.35* (±0.22)	**36.00** (±30.41)	0.53* (±0.08)	9.05* (±5.20)	0.56* (±0.14)	**20.86** (±13.72)

Table 3. Mean and standard deviation of the Coverage metric, the bottom row shows the Wilcoxon test results

Dataset	$\text{ARIA/ARIA}_\text{d}$	ARIA/GNG	ARIA/AP
Auto-mpg	0.93(±0.03)/0.00(±0.00) + (0.000583)	1.00(±0.00)/0.00(±0.00) + (0.000583)	0.98(±0.05)/0.00(±0.00) + (0.000583)
Bodyfat	0.66(±0.13)/0.03(±0.04) + (0.000029)	0.72(±0.08)/0.00(±0.02) + (0.000014)	0.01(±0.03)/0.47(±0.07) - (0.000014)
Servo	0.87(±0.06)/0.02(±0.04) + (0.000000)	0.98(±0.05)/0.00(±0.01) + (0.000000)	0.61(±0.08)/0.19(±0.08) + (0.000000)
Housing	0.95(±0.01)/0.00(±0.00) + (0.000058)	1.00(±0.00)/0.00(±0.00) + (0.000016)	1.00(±0.00)/0.00(±0.00) + (0.000016)
Pollution	0.28(±0.23)/0.38(±0.25) ∼ (0.187691)	0.74(±0.39)/0.05(±0.11) + (0.000002)	0.90(±0.09)/0.00(±0.00) + (0.000000)
N_{O2}	0.32(±0.35)/0.12(±0.22) ∼ (0.140356)	0.73(±0.31)/0.04(±0.08) + (0.000209)	0.83(±0.16)/0.05(±0.11) + (0.000051)

To visualize one of the results obtained by the algorithms, Figure 1 shows the best Pareto front in terms of hypervolume obtained for each dataset.

It can be seen that no algorithm obtained a good Pareto front for *Pollution* dataset, possibly due to its small number of samples and high dimensionality. In some cases, we can note that ARIA_d, GNG and AP were not able to produce solutions with high number of prototypes (see plots for *Bodyfat* and *Housing* datasets). We can also see in Figure 1 that ARIA solutions dominated the other algorithm solutions mainly when it used a high number of prototypes. Employing a high number of prototypes and fixed dispersion (probably larger than the automatically defined ones obtained by ARIA), it is probably more likely to occur overfitting, leading to a poor performance on the test set.



(a) Dataset: *Auto-mpg*. (b) Dataset: *Bodyfat*. (c) Dataset: *Servo*.

(d) Dataset: *Housing*. (e) Dataset: *Pollution*. (f) Dataset: N_{O2}.

Fig. 1. Best Pareto front in terms of hypervolume

In general, GNG found unsatisfactory results when compared with ARIA, ARIA$_d$, and AP. ARIA was arguably the best among the analyzed algorithms. ARIA$_d$ was better than AP for some datasets, and AP was better for others, so it is difficult to choose an algorithm among these. It is worth mentioning that AP can be less computationally costly than the other algorithms. ARIA and GNG need to calculate the distance between data points and all prototypes at each iteration. AP chooses the prototypes from the data points, thus the similarity matrix needs to be computed only once. However, an efficient mechanism to store this similarity matrix should be employed for large datasets.

6 Conclusion

In this paper, we performed a multiobjective analysis to compare three adaptive clustering algorithms in the task of defining the internal structure of RBFNNs applied to regression problems. For each algorithm, the parameters that most influence the resulting number of prototypes composed the decision space, while the RMSE value and the number of RBFs formed the objective space.

As the reported experiments indicated, ARIA had the best performance. In most cases, its solutions achieved better results in terms of the multiobjective metrics adopted in this paper. Also, a great ARIA advantage is to determine the RBF dispersions automatically. This way of calculating the dispersions was more beneficial than the fixed dispersion, as we can see clearly by comparing the results of ARIA and ARIA$_d$. The greatest care in selecting the RBFs dispersions

possibly contributed to avoid overfitting when using a relative large number of prototypes.

The next step is to analyze these adaptive clustering algorithms for the definition of the RBFNN internal structure when applied to classification problems.

Acknowledgment. The authors would like to thank CNPq and CAPES for the financial support.

References

1. Guillén, A., Pomares, H., Rojas, I., González, J., Herrera, L.J., Rojas, F., Valenzuela, O.: Studying possibility in a clustering algorithm for RBFNN design for function approximation. Neural Computing and Applications 17 (1), 75–89 (2008)
2. MacQueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
3. Yeung, D., Ng, W., Wang, D., Tsang, E., Wang, X.Z.: Localized Generalization Error Model and Its Application to Architecture Selection for Radial Basis Function Neural Network. IEEE Trans. on Neural Networks 18(5), 1294–1305 (2007)
4. Bezerra, G.B., Barra, T.V., de Castro, L.N., Von Zuben, F.J.: Adaptive Radius Immune Algorithm for Data Clustering. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 290–303. Springer, Heidelberg (2005)
5. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315, 972–976 (2007)
6. Fritzke, B., et al.: A growing neural gas network learns topologies. In: Advances in Neural Information Processing Systems, vol. 7, pp. 625–632 (1995)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
8. Violato, R.P.V., Azzolini, A.G., Von Zuben, F.J.: Antibodies with Adaptive Radius as Prototypes of High-Dimensional Datasets. In: Hart, E., McEwan, C., Timmis, J., Hone, A. (eds.) ICARIS 2010. LNCS, vol. 6209, pp. 158–170. Springer, Heidelberg (2010)
9. Martinetz, T.: Competitive hebbian learning rule forms perfectly topology preserving maps. In: Proceedings of Int. Conf. on Artificial Neural Networks, pp. 427–434. Springer (1993)
10. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
11. StatLib–datasets archive, http://lib.stat.cmu.edu/datasets/ (downloaded in: March 22, 2012)
12. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice-Hall, New York (1999)
13. Moore, D.S., Mccabe, G.P., Craig, B.A.: Introduction to the Practice of Statistics, 6th edn. W.H. Freeman & Company (2007)
14. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation), 2nd edn. Springer (September 2007)

Mean Multiclass Type I and II Errors for Training Multilayer Perceptron with Particle Swarm in Image Segmentation

Michel M. dos Santos[1], Mêuser J.S. Valença[1], and Wellington P. dos Santos[1,2]

[1] Escola Politécnica de Pernambuco, Universidade de Pernambuco, Brasil
{mms2,meuser}@ecomp.poli.br
[2] Núcleo de Engenharia Biomédica, Centro de Tecnologia e Geociências, Universidade Federal de Pernambuco, Brasil
wellington.santos@ieee.org

Abstract. Image segmentation can be posed as a multiclass classification problem. In doing so, segmentation evaluation can be made through multiclass classification errors. Instead of being used for evaluation, in this work the mean multiclass type I and II errors are proposed for multilayer perceptron training via particle swarm optimization. Moreover, some relations involving mean multiclass errors and conditional errors are exposed. Applied to image segmentation, mean multiclass errors were compared to mean squared error as objective functions. The approach was effective and able to provide accuracy and precision gains, resulting in a lower number of function evaluations in a cross-validated experiment.

Keywords: Mean Multiclass Errors, Multilayer Perceptron, Classification, Image Segmentation, Particle Swarm Optimization.

1 Introduction

In image processing, segmentation is a crucial task since subsequent processing depends on it [1]. Magnetic resonance image segmentation requires high accuracy, such necessity remains a challenge for the automated segmentation methods [2]. Artificial neural networks (ANN) are one of possible methods applied to image segmentation, some advantages of such techniques are the ability to learn and generalize data and real time execution [3]. On a supervised approach, pre-segmented reference images are used for training a neural classifier, as for instance a multilayer perceptron (MLP), in order to assign class labels to pixels, in this way the network input layer is fed directly with intensities or features previously extracted from reference image [4].

In training process, the weights of the network elements are adjusted by an optimization procedure. One typical approach consists in applying a gradient descent based method called backpropagation that minimizes the mean squared error (MSE) [5]. Some drawbacks of backpropagation are the possibility to get

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 135–142, 2012.

stuck in local minimum and the restriction over error form. As alternative, evolutionary algorithms are global optimizers that enable the use of different error measures without being necessary to develop one specific procedure for each objective function form [6]. Particle swarm optimization (PSO) is a population based method inspired by collective behavior of bird flock, in such way it takes advantage of group information. One of the early applications of PSO was on training MLP network [7].

After training process, the neural classifier is applied over a test dataset for evaluation. Beyond binary classification, image segmentation can arise as a multiclass classification problem. Two quality metrics used for evaluation of multiclass classification are the errors of type I and II [8,9]. In the present work, a MLP is trained through PSO, with mean multiclass errors, instead of the MSE, aiming at improving classification accuracy. Moreover, some relations between mean multiclass errors and conditional errors are exposed.

In the following, some previous works are briefly discussed. An evolutionary algorithm was used to improve classification ability of MLP network through multiobjective minimization of type I and II errors, besides network size. However, the application was restricted to binary classification tasks [10].

A generalized error function, which can be used in the common backpropagation scheme, was proposed with a focus on MLP data classification in [11]. Such proposed error form was able to achieve superior performance in some cases, compared to other error functions. Aside issues involving backpropagation use, the minimization of generalized error does not ensure minimization of classification error.

In [12], feedforward networks trained with a variation of PSO were applied to classification. Nevertheless, the objective function minimized was the misclassification rate.

Further sections are organized as follows: in section 2, MLP training with PSO is reviewed, while is highlighted the approach for the present work. Conditional errors and some relations to mean multiclass errors are presented at section 3. Results and conclusions are in sections 5 and 6, respectively.

2 Multilayer Perceptron Learning and Particle Swarm Optimization

In MLP training, an evolutionary algorithm can be used to evolve different network features [6,13]. In the present work, the network structure is fixed and only the network weights are adapted, the same setting in which some studies compare PSO and backpropagation for training MLP network [14,15].

In PSO, the i-th particle is associated to a position vector $\mathbf{x}_i \in \mathbb{R}^n$ and a velocity $v_i \in \mathbb{R}$. By applying PSO algorithm to evolve a neural network, each particle position vector contains neural network's weights and represents a point in the search space. In this way, each position visited by a particle defines a network setting that is evaluated for all input patterns in training dataset. Thus, the error metric used to evaluate the network performance corresponds to the

value of objective function for the respective particle. A common metric for comparing particles (networks) is based on MSE [14,15], however other metrics are possible also [12].

During PSO evolution, the update on velocity of a given particle is influenced by its own best position $\mathbf{p}_i \in \mathbb{R}^n$ and by global best position $\mathbf{g} \in \mathbb{R}^n$, according to equation

$$v_{i,j} = wv_{i,j} + c_1 r_p(p_{i,j} - x_{i,j}) + c_2 r_g(g_j - x_{i,j}), \tag{1}$$

where w, c_1 and c_2 are real parameters, r_p and r_g are uniform random number in $[0, 1]$ and indexes i and j indicate, respectively, particles and coordinates in search space. Moreover, positions are updated as follows

$$x_{i,j} = x_{i,j} + v_{i,j}, \tag{2}$$

that is referred as global PSO with inertia weight w [13]. In the work [12], the particles' velocities and search space were limited. In the present work, a similar approach is carried out, when a particle exceeds maximum velocity, its current velocity is reset to maximum velocity, which is defined as a fraction of search space width. Besides, when some particle goes to outside of search space, its position is redefined to the border and its velocity is inverted and reduced.

Overfitting is a well known problem in neural networks, a way to tackle such problem is to use cross-validation to control generalization ability [5].

3 Image Segmentation and Classification Evaluation

Image segmentation can be posed as a supervised multiclass classification problem. In doing this, the methods used to evaluate supervised classification can also be used for segmentation evaluation. As long as the tasks are coincident, if the classification error is minimized, then the segmentation quality is improved.

Some classification quality measures are defined through the confusion matrix: let $t_{i,j}$ be the number of class j patterns assigned to class i, an element of an $N \times N$ confusion matrix, where N is the number of classes [16]. The global accuracy or correctness is

$$c_g = \frac{\sum_{k=1}^{N} t_{k,k}}{\sum_{i=1}^{N} \sum_{j=1}^{N} t_{i,j}}, \tag{3}$$

since maximum global correctness is equal to one, it follows that global error is $e_g = 1 - c_g$. Moreover, two conditional metrics are particularly important in following presentation:

- correctness conditioned on true (T) class k,

$$c_k^T = \frac{t_{k,k}}{\sum_{i=1}^{N} t_{i,k}}, \tag{4}$$

is the accuracy given the true class k or percentage of correct detection among elements of true class k, it is related to sensitivity and specificity concepts in binary classification. Also, the conditional correctness is at most equal to one, so the conditional error on true class k is $e_k^T = 1 - c_k^T$;

- correctness conditioned on assigned (A) class k,

$$c_k^A = \frac{t_{k,k}}{\sum_{i=1}^{N} t_{k,i}}, \tag{5}$$

is the accuracy given the assignment to class k or percentage of true class k elements among elements attributed to class k, that is related to predictive value. Likewise, the conditional error on assigned class k is $e_k^A = 1 - c_k^A$.

While c_k^T can be used to evaluate algorithm's ability to distinguish class k items from items belonging to another classes. On the other hand, c_k^A is the percentage of items really belonging to class k, among elements attributed to class k, what can be used as the expected correctness when a classifier assigns label k.

Following a similar path [8,9], two multiclass error measures can be defined for each class k: type I error is the number of class k patterns do not classified as k, divided by the total number of class k patterns, that is precisely the conditional error on true class k rewritten as

$$e_k^I = e_k^T = \frac{\sum_{i=1}^{N} t_{i,k} - t_{k,k}}{\sum_{i=1}^{N} t_{i,k}}; \tag{6}$$

type II error is defined in a slightly different way as the number of items pertaining to other classes and assigned to class k, divided by the total number of patterns that are not of class k, as follows

$$e_k^{II} = \frac{\sum_{i=1}^{N} t_{k,i} - t_{k,k}}{\sum_{i=1}^{N}\sum_{j=1}^{N} t_{i,j} - \sum_{i=1}^{N} t_{i,k}}. \tag{7}$$

Ideally, a classifier must have a minimum mean error per class. Let the mean type I error (MTIE) be

$$\bar{e}^I = \frac{1}{N}\sum_{k=1}^{N} e_k^I = \frac{1}{N}\sum_{k=1}^{N}\left(1 - \frac{t_{k,k}}{\sum_{i=1}^{N} t_{i,k}}\right) = 1 - \frac{1}{N}\sum_{k=1}^{N}\frac{t_{k,k}}{\sum_{i=1}^{N} t_{i,k}}, \tag{8}$$

from previous expression is straightforward conclude that minimizing mean multiclass type I error is equivalent to maximize mean conditional correctness on true classes. The mean type II error (MTIIE) is also a function of a conditional error, however it appears in a weighted sum, in the form:

$$\bar{e}^{II} = \frac{1}{N}\sum_{k=1}^{N} e_k^{II} = \frac{1}{N}\sum_{k=1}^{N}\left(1 - \frac{t_{k,k}}{\sum_{i=1}^{N} t_{k,i}}\right)\left(\frac{\sum_{i=1}^{N} t_{k,i}}{\sum_{i=1}^{N}\sum_{j=1}^{N} t_{i,j} - \sum_{i=1}^{N} t_{i,k}}\right). \tag{9}$$

Instead of being used for evaluation, such mean multiclass errors are proposed here as objective functions to be minimized for training a multilayer perceptron through PSO algorithm.

4 Experimental Setup

Two main experiments were conducted, the experimental setting common to both is presented as follows. Multilayer perceptrons were trained through PSO with MSE, MTIE and MTIIE as objective functions. The dataset employed was the BrainWeb that provides simulated resonance magnetic images [17]. Three types of images were selected, these are characterized accordingly with acquisition parameters by the weighing in T1 and T2 relaxation times and in proton density (PD). All images had 1mm of slice thickness, 3% of noise level, 20% of intensity non-uniformity and resolution of 181 x 217 x 181 pixels. The three types of images correspond to the number of network inputs for each pixel. On the other hand, the number of network outputs was determined by the three classes considered (white and gray substances, besides liquor). It was employed the 1-of-C coding, in which the output class is indicated by the neuron in last layer yielding the maximum value. A hidden layer with six neurons and logistic activation function for all neurons were selected empirically. Each algorithm execution (training process) was repeated 30 times for statistical evaluation, in each execution a random sample of size 3000 pixels was generated with the three classes balanced. In PSO, a number of 20 particles was used, further parameters were $w = 0.65$, $c_1 = 3$, $c_2 = 1$, the search space at each coordinate was limited to range [-20,20], maximum velocity was set as 20% of search space width, when some particle went outside the search space its velocity was inverted and reduced multiplying by -0.01.

The purpose of first experiment (A) is to study the performance of MSE, MTIE and MTIIE as a function of the number of evaluations. The experiment (B) aims at practical application and employs a cross-validation procedure in order to control overfitting. These two experiments differ mainly in the stopping criterion and performance comparison.

In experiment A, each sampled pixel set was divided into training (66%) and test (33%) datasets, the maximum number of function evaluations (NFE) on training set or cycles was fixed at 4000. During each training process, for each particle evaluated on training set, the current best particle was used to calculate the global accuracy on the test set. For evaluation, statistics of the global accuracy summarizing executions along cycles are used in a graphical comparison.

In experiment B, each sampled pixel set was divided into training (50%), validation (25%) and test (25%) sets, it was employed a cross-validation with early stopping criterion determined by reaching either a maximum number of cycles equal to 4000 or error without decreasing per 1000 cycles on validation set.

Statistics for NFE in training set as well as for final global accuracy in test set are used to compare objective functions performance. For statistical analysis, Wilcoxon's rank sum test was carried out to test the null hypothesis that samples come from the same underlying distribution [18].

5 Results

5.1 Results for Experiment A

For the first experiment, the mean global accuracy on 30 executions and along cycles is presented in Figure 1, for each case in which MSE or MTIE or MTIIE was set as objective function.

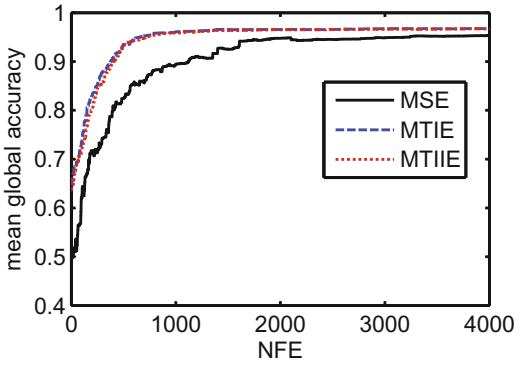


Fig. 1. Average global accuracy in test set along cycles

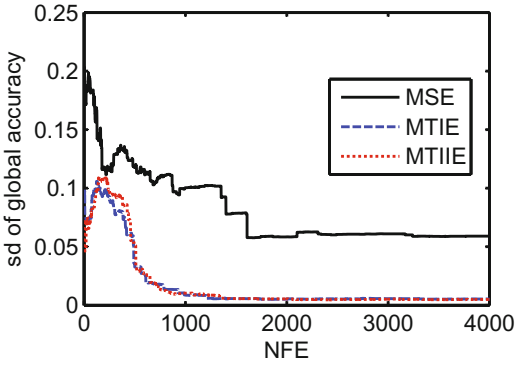


Fig. 2. Dispersion of global accuracy in test set

On average, the optimization of mean multiclass errors provided a higher global accuracy for the same NFE. On the same condition, when looking at dispersion of global accuracy as given by standard deviation in Figure 2. It is possible conclude that a higher precision was obtained when using mean multiclass errors, mainly after 1000 function evaluations and coincidently with mean global accuracy stabilization.

5.2 Results for Experiment B

In the cross-validated experiment, statistics for each dataset and for each case in which MSE, MTIE and MTIIE was set as objective function are presented in Table 1. Estimates for global accuracy and NFE over 30 executions are summarized in terms of means and standard deviations (sd), besides p-values (pv) resulting from statistical comparison for each one of multiclass errors versus MSE.

Table 1. Statistics for the cross-validated experiment

	Global accuracy in test set			NFE over training set		
	mean	sd	pv	mean	sd	pv
MSE	0.950	0.057	-	3147	837	-
MTIE	0.966	0.009	0.013	2497	680	0.006
MTIIE	0.965	0.007	0.039	2353	794	<0.001

On left hand side of Table 1, greater and more precise global accuracy was obtained when the objective function was one of two mean multiclass errors, as can be noted at means and deviations, respectively. Moreover, the use of MSE required more function evaluations than each one of the mean multiclass errors as objective functions, as can be seen in mean column on the right hand side of Table 1. These differences were significant at the 5% level.

6 Conclusion

Mean multiclass type I and II errors were proposed for multilayer perceptron training through PSO. Applied to an image segmentation problem, the mean multiclass errors were compared to MSE as objective function. The use of multiclass errors was able to produce accuracy and precision gains, also a lower number of function evaluations in a cross-validated experiment. Mean multiclass errors could be easily weighted in order to penalize misclassification into a given class, what is particularly useful in image segmentation. Furthermore, the experiments were restricted to one dataset of three balanced classes, the performance of mean multiclass errors for other datasets and in situations with increased class number or in unbalanced datasets is a matter for future work. This study was essentially experimental, a theoretical analysis would be interesting also.

Acknowledgments. Research partially supported by CAPES, Brasil.

References

1. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall (2002)
2. Balafar, M.A., Ramli, A.R., Saripan, M.I., Mashohor, S.: Review of Brain MRI Image Segmentation Methods. Artif. Intell. Rev. 33, 261–274 (2010)
3. Sharma, N., Aggarwal, L.M.: Automated medical image segmentation techniques. Journal of Medical Physics 35(1), 3–14
4. Egmont-Petersen, M., Ridder, D., Handels, H.: Image Processing with Neural Networks - a Review. Pattern Recognition 35(10), 2279–2301 (2002)
5. Haykin, S.: Neural Networks: a Comprehensive Foundation. Prentice-Hall (2001)
6. Yao, X.: Evolving Artificial Neural Networks. Proceedings of the IEEE 87, 1423–1447 (1999)
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
8. Yasnoff, W.A., Mui, J.K., Bacus, J.W.: Error Measures for Scene Segmentation. Pattern Recognition 9(4), 217–231 (1977)
9. Zhang, Y.: A Survey on Evaluation Methods for Image Segmentation. Pattern Recognition 29, 1335–1346 (1996)
10. Castillo, P.A., Arenas, M., Merelo, J.J., Rivas, V.M., Romero, G.: Multiobjective Optimization of Ensembles of Multilayer Perceptrons for Pattern Classification. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN IX. LNCS, vol. 4193, pp. 453–462. Springer, Heidelberg (2006)
11. Silva, L.M., de Sá, J.M., Alexandre, L.A.: Data Classification with Multilayer Perceptrons Using a Generalized Error Function. Neural Networks 21, 1302–1310 (2008)
12. Zamani, M., Sadeghian, A.: A Variation of Particle Swarm Optimization for Training of Artificial Neural Networks. In: Computational Intelligence and Modern Heuristics, Al-Dahoud Ali (2010)
13. Eberhart, R.C., Shi, Y.: Computational Intelligence - Concepts to Implementations. Morgan Kaufmann (2007)
14. Mendes, R., Cortez, P., Rocha, M., Neves, J.: Particle Swarms for Feedforward Neural Network Training. In: 2002 International Joint Conference on Neural Networks IJCNN 2002, vol. 2, pp. 1895–1899 (2002)
15. Gudise, V.G., Venayagamoorthy, G.K.: Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks. In: 2003 IEEE Swarm Intelligence Symposium - SIS 2003, pp. 110–117. IEEE Press (2003)
16. Egmont-Petersen, M., Talmon, J.L., Brender, J., McNair, P.: On the Quality of Neural Net Classifiers. Artif. Int. Artif. Int. in Med. 6(5), 359–381 (1994)
17. Kwan, R.K.S., Evans, A.C., Pike, G.B.: MRI Simulation-Based Evaluation of Image-Processing and Classification Methods. IEEE Transactions on Medical Imaging 18(11), 1085–1097 (1999)
18. Fay, M.P., Proschan, M.A.: Wilcoxon-Mann-Whitney or T-test? On Assumptions for Hypothesis Tests and Multiple Interpretations of Decision Rules. Statistics Surveys 4, 1–39 (2010)

An Improved ABC Algorithm Approach Using SURF for Face Identification

Chidambaram Chidambaram[1,2], Marlon Subtil Marçal[2], Leyza Baldo Dorini[2], Hugo Vieira Neto[2], and Heitor Silvério Lopes[2]

[1] State University of Santa Catarina-UDESC, Brazil
chidambaram@udesc.br
http://www.sbs.udesc.br
[2] Federal University of Technology - Paraná - UTFPR, Brazil
marlon8968@gmail.com, {leyza,hvieir,hslopes}@utfpr.edu.br
http://www.utfpr.edu.br

Abstract. Face recognition is being intensively studied in the areas of computer vision and pattern recognition. Working on still images with multiple faces is a challenging task due to the inherent characteristics of the images, the presence of blur, noise and occlusion, as well as variations of illumination, pose, rotation and scale. Besides being invariant to these factors, face recognition systems must be computationally efficient and robust. Swarm intelligence algorithms can be used for object recognition tasks. Based on this context, we propose a new approach using an improved ABC implementation and the interest point detector and descriptor SURF. To assess the robustness of our approach, we carry out experiments on images of several classes subject to different acquisition conditions.

Keywords: Face Identification, Image Variations, Interest Point Detectors, Swarm Intelligence, ABC algorithm, SURF, Still Images.

1 Introduction

Face recognition systems are generally classified into two categories: face verification and face identification. The first performs the matching between a query face image and a template, and the second compares the query image against several templates of a database [1]. In other words, the identification requires a onetomany matching process, while the verification is performed considering a onetoone image matching. In real world applications, the input images may be subject to different degradation problems during the acquisition procedure. These problems may occur due to several reasons, including environmental conditions (non-uniform illumination, occlusion and changes in pose and scale) and intrinsic face image characteristics (expression, hair styles, cosmetics and aging). Over the last two decades, several works have been proposed to make face recognition systems less sensitive to some of these problems, but only a few consider still images with multiple faces, focus of this work.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 143–150, 2012.

Still images with multiple faces may have complex background and comprise two main issues on a face identification process: the searching and detection of each face under different image conditions and the matching of a face object image against the faces obtained from still images. Real-life face recognition applications typically require accurate and fast search algorithms and matching methods at all stages [2]. According to Chellappa [3] in geometrical features based systems, window-based local processing to locate feature points using iterative search algorithms can be applied. Hence, the searching process in still images can be done using swarm intelligence optimization algorithms, such as Particle Swarm Optimization (PSO) [4] or ABC (Artificial Bee Colony) [5], for example. These approaches are computationally inexpensive, decentralized, adaptive and are not based on an explicit model of the environment [6]. Metaheuristic population-based optimization algorithms, such as those above mentioned were successfully applied in the development of fast algorithms for recognition problems [7] and are the basis of the approach proposed in this paper.

To reduce the associated computational cost of the second issue, instead of matching all image pixels (as in the traditional template matching approach), an alternative could be to match some features extracted from invariant image locations [8]. In this context, interest points can be considered to describe image features like color, texture and shape. Since its development, interest points have been mainly applied for object recognition and other related tasks [9].

The main objective of this work is to present a novel approach using the improved ABC algorithm (iABC) [7] to search and identify faces in still images, based on features generated by the interest point detector and descriptor, SURF (Speeded-UP Robust Features). Although there are many other ways to search and recognize faces in still images using SURF, the computational experiments carried out in this work show that its combination with a swarm intelligence approach yields accurate and efficient results during an iterative search and matching process. Since the focus of the present work is to identify faces in still images using iABC with SURF, we have also formulated this problem as an optimization problem where optimal values for image parameters have to be determined. To evaluate the proposed approach we conduct several computational experiments with real digital images acquired under different image conditions. This is the main contribution of the present work.

This paper is organized as follows. In Section 2, related topics and works regarding interest point detectors, SURF and ABC algorithm are explained. Experimental details and results are discussed in Section 3. Finally, Section 4 outlines some conclusions and future work.

2 Related Topics and Works

2.1 ABC Algorithm for Face Identification

In the ABC algorithm [5], each food source is considered as a possible solution for an optimization problem. The amount of employed bees represents the number of solutions (SN) in the population. Each solution is represented by

X_i, $i \in (1, ..SN)$, a d-dimensional vector where d corresponds to the number of parameters to be optimized. Once the employed bees are created, the search process starts and it is repeated by a predefined number of cycles, defined by the Maximum Cycle Number (MCN) value. Thus, the basic ABC algorithm has three control parameters determined at the beginning of the search process: SN, $Limit$ and MCN. The parameter $Limit$ is the product of dimension (d) by the number of initial solutions (SN).

For the face identification problem using ABC algorithm, a face image is represented by a 4-tuple, denoted by (x, y, s, θ), representing the coordinate values of column and row of the central point, scale and rotation, respectively. Such transformation parameters are optimized to determine the most similar face in a still image. In our image context, the search space is limited by restricting the range of each parameter as follows: $0 \leq x \leq n$, $0 \leq y \leq m$, $0.5 \leq s \leq 1.5$ and $-\pi/2 \leq \theta \leq \pi/2$. A bee (or solution) corresponds to a position in the still image, represented by a 4D vector $X_{id}(X_{i1}, X_{i2}, X_{i3}, X_{i4})$, where $d = 4$ and $i \in 1, ..SN$.

The following pseudocode summarizes briefly the basic ABC algorithm [5] used for the definition of improved iABC:

1. Initialize the population of employed bees (positions) of size SN;
2. Evaluate the fitness of initial population;
3. Repeat until the stopping criteria is met(cycle = MCN):
   (a) *Produce new neighborhood solution (positions) and evaluate their fitness;*
   (b) *Move the employed bees to new solutions (positions);*
   (c) *Calculate the probability values for the solutions;*
   (d) *Produce the new solutions for the onlookers using the probability values and compute their fitness;*
   (e) *Move the onlooker bees to new solutions (positions);*
   (f) *Produce scout bees;*
   (g) *Increment cycle;*

As shown in the pseudocode, during the steps 3(a) and 3(d), employed and onlooker bees perform the local search process to find out the optimal solutions, meanwhile, in step 3(f), by evaluating the Limit value, scout bees are produced to perform global search, aiming at finding new unexplored solutions. The basic ABC algorithm perturbates only one parameter at time when a new neighborhood solution is generated for both employed and onlooker bees, during the steps 3(a) and 3(d). At the end of step 3(f), if possible, when the limit counter value of a specific solution exceeds the $Limit$, one scout will be generated even though it can be more than 5% of the population [10]. If none of the solution exceeds the $Limit$ value, no scout bee will be produced. Based on this context, three different mechanisms for improving the ABC were studied, such as the generation of scout bees, perturbation of all variables and explosion of stagnated population.

To define the iABC version, the basic ABC algorithm [5] was tested with the combination of three improvement strategies. According to the iABC, for the object recognition problem using landscape images, the generation of randomly created scout bees produced no significant effect in performance, even though scout bees mechanism can improve the global search ability of the ABC

algorithm. On the other hand, the use of explosion of stagnated population mechanism was shown to accelerate the convergence of the algorithm without losing quality of solutions. Based on the experimental analysis, the iABC was defined by selecting the best strategy which consists of perturbation of multiple variables, explosion and without generation of scout bees. Overall, the improved ABC, proposed by Chidambaram and Lopes [7], can be a good alternative to real-world object identification problems. Hence, in this work, we use the iABC algorithm, to search and identify faces in still images.

2.2 Interest Point Detectors

An interest point detector is an algorithm that uses an image as input and outputs a set of points that can be identified with high repeatability in location. Interest points can be defined as a set of image pixels that have high level of variation in reference to a predetermined local measure [11]. Compared to low-level features like color, interest points are considered more stable and reliable [12]. Most of the detectors finally generate descriptor vectors which contain the information regarding the neighborhood of every interesting point in an image. Object recognition can be considered as one of the main application of interest point detectors [9]. Among many detectors and descriptors, two of them can be mentioned as most known recently: Scale Invariant Features Transform (SIFT) [9] and Speeded-Up Robust Features (SURF) [8].

In 2008, Bay and colleagues [8] developed a novel scale and rotation-invariant detector and descriptor, called SURF. Although SURF can be conceptually similar to SIFT, SURF is less sensitive to noise, invariant to scale and rotation and outperforms SIFT. SURF builds a descriptor vector of 64 dimension using relevant feature information arround every interest point. SURF descriptors reduce the time for feature computation and matching, and also increases robustness. Repeatability rate is the only measure of stability which is strongly accepted as a standard computer vision performance metric for interest points [11]. Measurements of repeatability will quantify the number of repeated points detected under varying conditions, i.e, the percentage of matched points that are repeated in both images.

3 Computational Experiments and Results

The search and identification using iABC with SURF for finding image correspondences is repeated as an iterative optimization process. This process is similar to the template matching in which maximum correlation (the best possible match) is possible when the incoming object image is identical to the imaget cut from the still image. According to the nature of this work, face detection is not done explicitly rather it is indirectly associated with the identification step. As the initial step, the interest points of the still image and face object image are calculated using SURF. In the next step, interest points of still image, mainly descriptors of 64-D vector, are stored in a separate matrix structure (the same size

of still image) at the corresponding keypoints (coordinates) of interest points. This matrix structure is created to avoid the calculation of interest points for each image cut from the still image during the iterative process. Using a 4-D vector for each individual of the population, a face image is obtained from the still image and its interest points (descriptors) are obtained from the corresponding coordinates in the matrix. Thus, this procedure can reduce the computational effort that will be spent on interest points calculation. It is important to mention that each cut of image from still image could possibly be an optimal solution or the exact face image that has to be identified. During the matching stage, the interest points correspondences between images are identified using distance measures of keypoints or coordinates and descriptors.

The iABC algorithm was implemented C programming language with OpenCV functions. All experiments were run on a cluster of computers using Pentium quad-core processors under Linux. For our tests, we have defined the number of employed bees (or initial solutions) SN = 80, MCN = 100 and *Limit* was not used, as defined in Section 2.1. The number of runs was set to 30. During the search, when stagnation occurred for 30% of the MCN, explosion was performed. In addition to these ABC parameters, to evaluate the matching of interest points, two more control parameters were used to determine the matching points between the face object image and the face obtained from the still image using descriptors and coordinates: (1) keypoints or coordinates distance threshold (set to 50); (2) descriptor distance threshold (set to 0.09). Both values were determined empirically. The fitness was calculated using the number of matched interest points against the number of interest points of face object image.

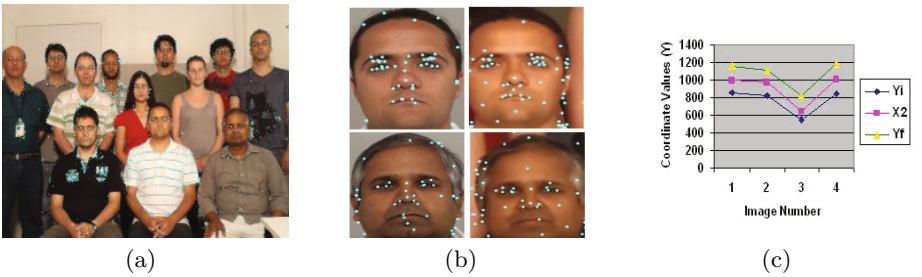
We have performed nine experiments to show the robustness of the proposed approach. These experiments widely cover the different image conditions that occur in real-world. The most relevant results are discussed in this section. All face object images that were used in the experiments were obtained separately and are different from those in the still images. Images were captured under two different illumination conditions: (1) Under controlled illumination condition using a specfic lighting system; (2) uncontrolled illumination condition. Under uncontrolled illumination, the image aquisition was done in two ways: under natural lighting conditions (flourescent lights) of the room, and also, with partially controlled illumination. The latter is used only for still images. All face object images used in this work were obtained under controlled illumination.

The first experiment was done using a still image obtained under controlled condition and without any variation. In Table 1, some result data are shown such as central coordinate values of the identified face in still image (X_1 and X_2), average fitness of best solutions, number of times the face was correctly found in still image within 30 runs (Num. Id), the left-top corner (X_i and Y_i) and the right-bottom corner (X_f and Y_f) coordinate values of face in still image, and the average number of evaluations per run (Num. Eval). The execution time and the number of evaluations are included in this work for comparison purposes of performance with other approaches.

Table 1. Solutions found by iABC using still image without any variation obtained under controlled condition

	Coordinates of face found by iABC					Num.	Coordinates of face in still image				Exec.	Num.
Image	X_1	X_2	*Angle*	*Scale*	*Fitness*	*Id*	X_i	Y_i	X_f	Y_f	*Time*	*Eval*
1	2102	1003	−0.008	0.946	0.3196	29	1980	860	2250	1170	861	25348
2	1446	971	−0.013	0.999	0.4667	30	1310	820	1540	1120	837	31240
3	1287	650	−0.064	0.898	0.2474	29	1164	560	1354	820	1028	25511
4	751	1013	−0.173	0.915	0.3222	29	640	850	880	1190	1058	34022

From the results of the first experiment shown in Table 1, it can be observed that the angle values are relatively close to zero, since the faces in still image are almost without any rotation. The scale values varies from 0.9 to 0.99 which indicates that the face obtained from still image is almost of the same size of face object image presented to the algorithm. In the same table, the fitness values vary from 0.24 to 0.47 and represents the matched interest points of face region from still image and face object image. In addition to these data, the images with interest points are shown in the Figure 1 (a) and (b). The main goal of this work is to locate and identify the face object image within the region of face in still image. Hence, no exact match of coordinates is expected. This can be observed from X_1 and X_2 whose values are within the range of X_i and Y_i, and X_f and Y_f. It is demonstrated in Figure 1(c) by the y coordinate value (X_2- middle line) of the identified face image by iABC. To determine the identification rate of face image, this condition was observed in all other experiments.



(a) (b) (c)

Fig. 1. Images with interest points - Still image (a), Face object images (1st Column) and Faces from still image (2nd Column) (b), Coordinate value Y (X_2) of identified face by iABC within the region of face in the still image (between Y_i and Y_f)) (c)

To test the robustness of the iABC algorithm with SURF, we have also conducted experiments using images obtained under different conditions and artificially manipulated images as shown in Table 2. The identification rate is the ratio of the number of times the face object image was identified by the number of runs in each experiment. For example, in the first experiment, even though the still image is obtained under controlled condition, the average identification rate is about 88% among eight experiments. This is because of one face object image which has significant variation in comparison with the face in still image. In the next two experiments, using the same face object images as in the first experiment and still images obtained under partially controlled and uncontrolled illumination, the average identification rate reached 72% and 13%, respectively. Such results confirm the well-known influence of illumination conditions in the recognition process.

Table 2. Result data of all experiments

Experiment Description (Image Type)	Num. of Images	Num. of Experiments	Average ID. Rate
Controlled Illumination	08	08	88%
Uncontrolled Illumination	08	08	13%
Partially Controlled Illumination	08	08	72%
Face Images with occlusion	04	04	54%
Rotation (-30 to +30 degrees)[1]	02	20	100%
Scale Variation (80% to 120%)[1]	02	16	97%
Gaussian Blur (04 levels)[1]	02	08	33%
RGB Noise (04 levels)[1]	04	16	100%
Illumination Variation (04 levels)[1]	04	16	99%

[1]Artificially Manipulated Images

The identification rate in the experiment with occluded faces reached about about 54%. This type of images may require more investigation using different levels of occlusion. However, in our study, the results show that the proposed approach is capable of identifying faces in such conditions. All other experiments were conducted using artificially manipulated images. The SURF is considered invariant to scale and rotation and is also less sensitive to noise [8]. Based on this point, the still image variations are divided into the following categories: RGB noise, Gaussian blur, changes in lighting, rotation and scaling. Increasing levels of Gaussian blur and RGB noise were added to the still image. Rotated images were generated in both clockwise and counterclockwise directions, varying from -30 to +30 degrees in increments of 5 degrees. Likewise, scaled still images were obtained by varying its original size from 80% to 120% in increments of 5%. Among these categories, only in Gaussian blurred images, the identification rate is about 33%, meanwhile, in the other categories, the iABC algorithm was robust enough to identify the face images in almost all experiments.

4 Conclusions

In this work, a new approach using iABC with the interest point detector SURF is proposed for the face identification problem. The promising results obtained from some relevant experiments showed that the proposed iABC with SURF can effectively work with images under different conditions and it can be suitable for face identification and related tasks. Thus, the proposed approach is robust to work with still images with mutiple faces. This approach differs from other similar works in the literature by the way the face identification process is conducted with aid of iABC algorithm. Besides this point, face detection is not done explicitly and it is indirectly associated with the face identification process which is driven by the evolution of the population using the interest points matching function. According to the experimental results, it can be observed that the illumination variation influences the face identification process. Consequently, this issue requires more research effort and deep study. In addition to this, future work will focus on reducing the number of evaluations and, consequently, the execution time.

References

1. Abate, A., Nappi, M., Riccio, D., Sabatino, G.: 2D and 3D face recognition: A survey. Pattern Recognition Letters 28, 1885–1906 (2007)
2. Pawar, V.N., Talbar, S.N.: An investigation of significant object recognition techniques. International Journal of Computer Science and Network Security 9(5), 17–29 (2009)
3. Chellappa, R., Wilson, C.L., Sirohey, S.: Human and machine recognition of faces: a survey. Proceedings of the IEEE 83(5), 705–741 (1995)
4. Kennedy, J., Eberhart, R.C.: Particle swarm optimisation. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
5. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 214(1), 108–132 (2009)
6. Keshtkar, F., Gueaieb, W.: Segmentation of dental radiographs using a swarm intelligence approach. In: Proceedings of the Canadian Conference on Eletrical and Computer Engineering, pp. 328–331 (2006)
7. Chidambaram, C., Lopes, H.S.: An improved artificial bee colony algorithm for the object recognition problem in complex digital images using template matching. International Journal of Natural Computing Research 1(2), 54–70 (2010)
8. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Computer Vision and Image Understanding 110(3), 346–359 (2008)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 2(60), 91–110 (2004)
10. Tereshko, V., Loengarov, A.: Collective decision-making in honey bee foraging dynamics. Computing and Information Systems 9(3), 1–7 (2005)
11. Trujillo, L., Olague, G.: Synthesis of interest point detectors through genetic programming. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 887–894 (2006)
12. Pimenov, V.: Fast image matching with visual attention and SURF descriptors. In: Proceedings of the 19th International Conference on Computer Graphics and Vision, pp. 49–56 (2009)

A Model Based on Genetic Algorithm for Investigation of the Behavior of Rats in the Elevated Plus-Maze

Ariadne A. Costa[1], Antonio C. Roque[1], Silvio Morato[2], and Renato Tinós[3]

[1] Department of Physics, FFCLRP, University of São Paulo (USP) 14040-901, Ribeirão Preto, S.P., Brazil
ariadne.ad@pg.ffclrp.usp.br
[2] Department of Psychology, FFCLRP, University of São Paulo (USP) 14040-901, Ribeirão Preto, S.P., Brazil
[3] Department of Computing and Mathematics, FFCLRP, University of São Paulo (USP) 14040-901, Ribeirão Preto, S.P., Brazil

Abstract. In this paper we propose the use of an artificial neural network associated to a genetic algorithm to develop a behavioral model of rats in elevated plus-maze. The main novelty is the fitness function used, which is independent of prior known experimental data. Our results agree with experimental tests, demonstrating that open arms exploration evoke greater avoidance. The perspective of the results are increased by analyzing Markov chains obtained by experiments with real rats and by computational simulations, suggesting that the general fitness function proposed summarizes the main relevant characteristics for the study of the rats behavior in the elevated plus-maze.

Keywords: Genetic algorithm, Artificial neural network, Elevated plus-maze, Rat, Markov Chain.

1 Introduction

The elevated plus-maze (EPM) is a simple experimental apparatus widely used to measure anxiety responses of rodents and even as a general research tool in neurobiological anxiety and defense study [1]. It consists of a plus-shaped maze composed of two open arms diametrically opposed and two enclosed arms also in opposite positions. These four arms are connected at a central area and the set is elevated from the floor. The great interest in this model occurs due its ease of use, since it is not necessary training the animals before the tests and they are not deprived of any basic resources. This experimental model is derived from some early work of Montgomery [2] to study the relation between anxiety and fear, based on the exposition of rats to this new environment, evoking simultaneously fear and curiosity. These feelings can be viewed as adaptive defense mechanisms of animals to dangerous stimuli and untried situations [3].

There are many studies proposed to understand the mechanisms evolved in the control of the behavior of rats in EPM [4–6]. Most of them are based on

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 151–158, 2012.

the results first obtained by Montgomery [2], that the rat spend significantly more time in the enclosed arms than in the open arms of the plus-maze, because the fear of exposure is larger than the anxiety of explore the new environment. Nevertheless, only three studies are based on computational models [7–9]. All these three models try to simulate a single naive rat exploring the EPM for five minutes and rely on the assumption that the rat has never previously been exposed to the plus-maze, because the habituation with the environment would probably affect the results, since after some time in the EPM the rats fail to explore it [2]. In these studies the space contained in the EPM is discrete, with three [8] or five [7, 9] positions (squares) in each arm.

The approach-avoidance conflict of rats in the plus-maze is the key concept of the model proposed in [7]. This model uses the exploratory motivation and the aversion as variables to define the dynamic probability of a position in the EPM be occupied by the computational agent, called virtual rat and represented by an artificial neural network (ANN). In contrast, in [8], the probability of occupation is given according to the direction of the movement. A different approach is adopted in [9], where the problem of obtaining an agent capable of reproducing the behavior of the rat in is viewed as an optimization problem. For this purpose, ANNs optimized by evolutionary computation (EC) are used to define the navigation strategy for a robot in an EPM replica.

In [10] the authors discuss that the implemented fitness function in EC can be chosen following two different methods. In the first one, the fitness function carries explicitly the constraints and capabilities of the intelligent agent, restricting its range of actions. In this case, the agent learns a specific function determined by the programmer. On the other hand, in the second case, the fitness function is based on a general survival criterion "that is automatically translated into a set of specific constraints by the characteristics of interactions between the organism and the environment" [10], actually simulating natural evolutionary processes. In [9] the first method was employed; its fitness function directly compares the output results obtained using EC with the intended results, i.e., prior known experimental data. The present work shows a variation of [9], in which we also combine ANN with EC, however use a generalized fitness function, as implied by the second method. The proposed fitness function is composed of two terms: one that stimulates the agent to explore new positions of the EPM and another that represent possible dangerous situations in which the agent is subjected in each position of the plus-maze. This function does not depend of comparisons with data of experiments with real rats.

The models developed in [7–9] are able to simulate the preference of the rats for the enclosed arms than for the open arms of the plus-maze, but the behavior of the rats is more complex than that. Therefore [11] encouraged the use of Markov chains [12], that have been already used to describe the behaviors of animals in other works [13–15]. The Markovian process considers time and space as discrete states and at each step of time the transition from the current position to another (or eventually the same) is done with a probability that depends only on the last

k states. This paper confront Markov chains exposed in [11], obtained by experiments with real rats, with Markov chains coming from our simulations.

The methodology proposed here is described in Sect.2. The experimental results obtained in simulations are presented in Sect.3. Finally, the conclusions of the paper are in Sect.4.

2 Methods

The agent implemented in this work (the virtual rat) is controlled by a recurrent multilayer perceptron (MLP) with Elman's architecture [16]. The advantage of using recurrent connections is that previous inputs are preserved in the network's internal state (internal memory), which is important because the next movement of the rat depends on the past visited positions. The ANN has 10 inputs, 4 hidden neurons and 4 outputs. The outputs indicate the next position to which the robot will move (one of the positions in its neighborhood). The inputs are given by the perception of the walls around the agent, i.e, they are equivalent to the reading of obstacle sensors placed around a robot (agent).

The synaptic weights of the ANN are optimized by a genetic algorithm (GA). In this way, each individual of the GA's population is given by a chromosome (an array of integers) that encodes a subset of possible solutions for the weights of the ANN. The initial population is randomly chosen. The selection operators of the GA are elitism and tournament. The elitism select two individuals with the best fitness in the population to pass unchanged the next population, while in the tournament two random individuals compete and that one with better fitness is selected with probability 0.75. The individuals selected by tournament are transformed by reproduction operators. Here, the GA applies single point crossover (according to a crossover rate) and integer mutation with uniform distribution (according to a mutation rate) to single elements.

The individual (set of weights of the ANN) is evaluated by letting the corresponding agent to navigate the virtual EPM during a number of iterations equivalent to the duration of a standard experiment with real rats (five minutes). The trajectory performed by the agent is recorded and used to compute the fitness of the individual according to the fitness function described in the next section.

2.1 Fitness Function

We developed a fitness function based on the general behavior of rats in the EPM. It is worth highlighting that the proposed function does not require experimental data. It is basically composed of two terms: reward and punishment. Reward, which increases the fitness of the individual, represents the interest of the rat of exploring not recently visited positions of the EPM. Punishment decreases the fitness of the individual and is related to the danger of occupying different positions of the plus-maze (e.g., open arms expose the agent to more danger than enclosed arms).

Table 1. Definition of all parameters involved in the fitness function

Parameter	Definition
n	number of time steps for the agent in the maze
p_t	EPM's position occupied in time step t, in which $p_t = 1, ..., p_{max}$
p_{max}	number of positions of the EPM
$r(p_t)$	reward $= \begin{cases} 1, \text{ if } p_t \text{ was not visited in the last } \gamma(n_{p_t}) \text{ steps} \\ 0, \text{ otherwise} \end{cases}$
$\gamma(n_{p_t})$	parameter related to the agent's "memory" of visiting the position p_t
n_{p_t}	number of time steps in which the position p_t was occupied
$s(p_t)$	punishment $= \begin{cases} -1, \text{ if } z_i < \alpha(p_t) \\ 0, \text{ otherwise} \end{cases}$
z	random number with uniform distribution in the interval [0,1]
$\alpha(p_t)$	$\begin{cases} \alpha_o \in [0,1], \text{ if } p_t \text{ is in an open arm} \\ \alpha_e \in [0,1], \text{ if } p_t \text{ is in an enclosed arm} \\ \alpha_c \in [0,1], \text{ if } p_t \text{ is in the central position of the EPM} \end{cases}$
α_o, α_e, α_c	probability of undergoing punishment respectively in the open arms, enclosed arms and center
β	weight for the punishment of the rat

The fitness function is given by:

$$f(\boldsymbol{x}) = \sum_{t=1}^{n} r(p_t) + s(p_t).\beta \ . \tag{1}$$

The definition of the parameters above can be seen in Table 1. As mentioned before, this fitness function simulates the rat's adaptation due to natural evolution.

2.2 The Virtual EPM

The virtual EPM is the same proposed in [7]. The plus-maze is divided into 21 positions being five in each arm and one in the central position (Fig.1(a)). The virtual rat can walk in all 21 positions, but to analyze the results we consider both open arms as if they were the same and proceed in the same way with the enclosed arms. It is because the plus-maze is symmetrical and its all parts are subject to uniform physical conditions. Thereby we have the reduced maze with 11 positions shown in the Fig.1(b), which decreases the behavioral variability in the model. The positions 1-5 consist the open arm, 7-11 consist the enclosed arm and 6 is the central position.

2.3 Evaluation of the Best Individuals

In order to evaluate the quality of the model, we need a method to compare the trajectories in the EPM obtained by the agent (virtual rat) and by real rats.

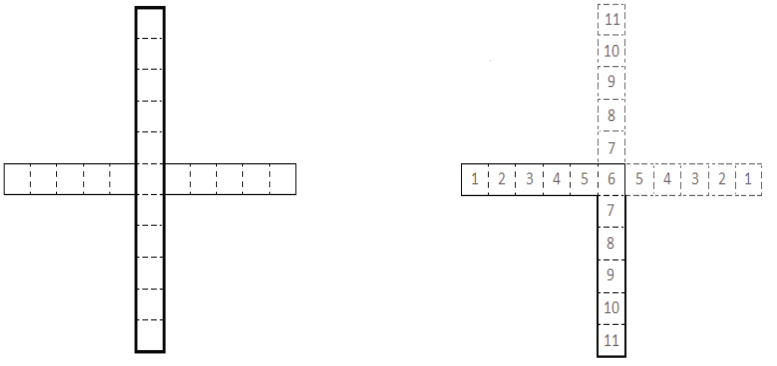


Fig. 1. (a) Representation of the elevated plus-maze considered in the simulations. Arms in bold correspond to closed arms. (b) Reduced EPM used for the analysis of the results, composed of 11 positions, outlined by solid line in the representation.

As the computational models simulate rat behavior in the EPM as a sequential probabilistic it is possible to use Markov chains to describe it [11]. We represent the firs-order homogeneous (independent of time) Markov chain with a stochastic matrix (P) where each element p_{ij} is the probability of displacement from the position i to the adjacent position j of the EPM. The values are normalized so that the sum of each row in the matrix is 1. The auto-transitions (steps in which the rat remains in the same state) are disregarded because our objective in using Markov chains is analyze the behavior contained in the motion. Although the transition probabilities change during the experiment, we work with homogeneous Markov chains because we are interested in the pattern of rat exploratory behavior during the test, and not in reproduce its exact path. Five minutes is the standard duration for the EPM's experiment since after that rats normally lose interest in exploring the maze. The stochastic matrix elements are calculated as [11]:

$$p_{ij} = \frac{f_{ij}}{\sum_{j=1}^{11} f_{ij}} \quad , \tag{2}$$

where $i, j = 1, 2, ..., 11$, $i \neq j$ (that are the positions of the reduced EPM). The frequencies f_{ij} are obtained by the trajectories of the rat during the period of test. From the stochastic matrix P we can calculate the probability of the vector of stationary states probabilities, called π:

$$\pi P = \pi \quad . \tag{3}$$

The 11 elements of π provide the probability of finding the rat in the respective positions of the EPM in a test. We still computed the time spent in each arm for comparison of the behavior of virtual and real rats.

3 Simulation Results

All the experimental results presented in this section are the average of 10 rats (control group analyzed in [11]) while the results based on computational simulations are the average over 30 executions of the program, with 500 individuals for 600 generations in the genetic algorithm. We have tested many sets of parameters and the selected for this work's simulations were: $\gamma(n_{p_t}) = 3$, $\beta = 5$, $\alpha_o = 0,015$, $\alpha_c = 0,010$ and $\alpha_e = 0,005$.

The most examined behavioral characteristic of rats in EPM is the time spent in each arm and in the central position during the experiment. The time spent in each arm obtained by experiments with real and virtual rats are shown in Fig.2. We consider each time step of the simulations as one second. Virtual rats spend similar proportions of time in each arm of the EPM in relation to real rats, remaining substantially longer in the enclosed arms.

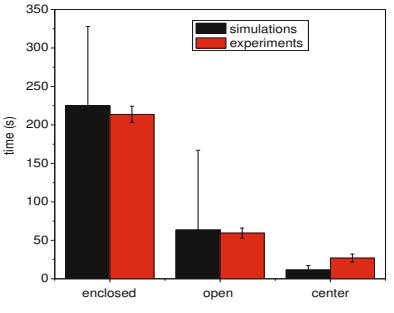


Fig. 2. Time spent in enclosed and open arms and in central position of the EPM for real rats (experiments) and for virtual rats (simulations)

Fig.3 presents the corresponding Markov chains to the data from real and virtual rats. The vector π, for the experiments with real rats and for computational simulations, that store the probability of finding the rat in each position of the plus-maze, can be seen in Fig.4.

One can observe that the behavior of the virtual rat is very close to the behavior of the real rats. As expected, the enclosed arms are more visited. Differences between simulated and experimental probabilities are higher for the open arms positions, which have the largest standard deviations in the experiments with real rats [11]. This implies that even those differences are acceptable for real rats.

$$\begin{bmatrix}
0 & 1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.35 & 0 & 0.65 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.46 & 0 & 0.54 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.44 & 0 & 0.56 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.45 & 0 & 0.55 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.34 & 0 & 0.66 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.42 & 0 & 0.58 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.47 & 0 & 0.53 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.48 & 0 & 0.52 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.54 & 0 & 0.46 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.00 & 0
\end{bmatrix}$$

(a)

$$\begin{bmatrix}
0 & 1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.51 & 0 & 0.49 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.39 & 0 & 0.61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.43 & 0 & 0.57 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.56 & 0 & 0.44 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.14 & 0 & 0.86 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.52 & 0 & 0.48 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.51 & 0 & 0.49 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.52 & 0 & 0.48 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.48 & 0 & 0.52 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.00 & 0
\end{bmatrix}$$

(b)

Fig. 3. (a) Experimental and (b) simulated Markov chains for rats in a 5-minute test in the EPM

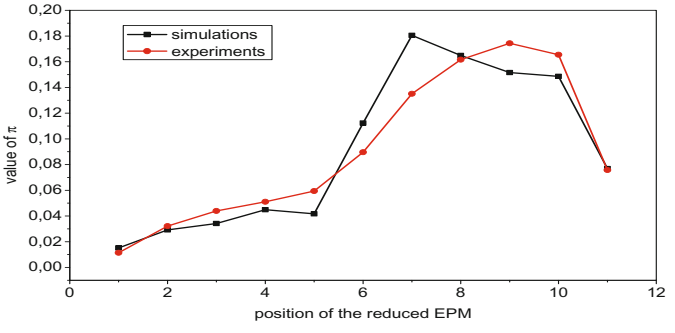


Fig. 4. Vector π for (a) real and (b) virtual rats

4 Conclusion

In this paper, we propose a new fitness function based on survival criteria, and exploring the conflict of fear and anxiety in rats in an EPM. The behavior in rats is acquired by natural selection, i.e., by an optimization process. In a similar way, we propose that the behavior of the virtual rat should be obtained by an optimization process too (in this case, artificial evolution).

The results indicate that the simulation results significantly agree, both in a qualitative and in a quantitative way, with data from real rats. Our model consists of a large simplification of the reality but it seems to be a good approximation for describing the behavior of rats in the EPM. As future work, we should study the influence of the parameter β (that is the weight of the rat's punishment) in the exploratory behavior of rats under different conditions (e.g., under the effects of anxiogenic and anxiolytic drugs), as well as the role performed by the neural network and by each neuron of its hidden layer.

Acknowledgments. This research is supported by FAPESP and CAPES. The authors are grateful to Helder Ken Shimo and Diego Henrique Ferreira for the contribution to the simulator development, and to Julián Tejada, Andrea M. G. Becerra, Ana Cristina M. Costa, Javier Leonardo R. Rodriguez for providing us the experimental data of their work.

References

1. Hogg, S.: A review of the validity and variability of the elevated plus-maze as an animal model of anxiety. Pharmacol. Biochem. Behav. 54(1), 21–30 (1996)
2. Montgomery, K.C.: The relation between fear induced by novel stimulation and exploratory behavior. J. Comp. Physiol. Psychol. 48, 254–260 (1955)
3. Graeff, F.G.: Brain defense systems and anxiety. In: Burrows, G.D., Roth, M., Noyes, R. (eds.) Handbook of Anxiety. The Neurobiology of Anxiety, vol. 3, pp. 307–354. Elsevier, Amsterdam (1990)
4. Pellow, S., Chopin, P., File, S.E., Briley, M.: Validation of open closed arm entries in an elevated plus-maze as a measure of anxiety in the rat. J. Neurosci. Methods 14(3), 147–167 (1985)
5. Salum, C., Roque-da-Silva, A.C., Morato, S.: Conflict as a determinant of rat behavior in three types of elevated plus-maze. Behav. Processes 63, 87–93 (2003)
6. Walf, A.A., Frye, C.A.: The use of the elevated plus maze as an assay of anxiety-related behavior in rodents. Nature Protocols 2, 322–328 (2007)
7. Salum, C., Roque-da-Silva, A.C., Morato, S.: Anxiety-like behavior in rats: a computational model. Neural Netw. 13(1), 21–29 (2000)
8. Giddings, J. M.: Modeling the Behavior of Rats in an Elevated Plus-Maze. Master's thesis. Acadia University (acess in April 11, 2012)
9. Shimo, H.K., Tejada, J., Roque, A.C., Morato, S., Tinós, R.: Use of evolutionary robots as an auxiliary tool for developing behavioral models of rats in an elevated plus-maze. In: 10 Proceedings of the 2010 Eleventh Brazilian Symposium on Neural Networks (2010)
10. Nolfi, S., Floreano, D.: Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press/Bradford Books, Cambridge (2000)
11. Tejada, J., Bosco, G.G., Morato, S., Roque, A.C.: Characterization of the rat exploratory behavior in the elevated plus-maze with Markov chains. J. Neurosci. Methods 193(2), 288–295 (2010)
12. Kemeny, J., Snell, J.: Finite Markov chains. Springer, NJ (1976)
13. Haccou, P., Dienske, H., Meelis, E.: Analysis of time-inhomogeneity in Markov chains applied to mother-infant interactions of rhesus monkeys. Animal Behaviour 31(3), 927–945 (1983)
14. Garcia-Perez, E., Mazzoni, A., Zoccolan, D., Robinson, H.P.C., Torre, V.: Statistics of decision making in the leech. J. Neurosci. 25, 2597–2608 (2005)
15. Yang, H., Chao, A.: Modeling Animals' Behavioral Response by Markov Chain Models for Capture–Recapture Experiments. Biometrics 61(4), 1010–1017 (2005)
16. Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the theory of neural computation. Addison-Wesley Publishing Company (1991)

Data Clustering Using Hybrid Particle Swarm Optimization

Ahmed A.A. Esmin[1,2] and Stan Matwin[2,3]

[1] Department of Computer Science University of Lavras (UFLA),
Lavras, MG, 37200-000, Brazil
ahmed@dcc.ufla.br
[2] School of Electrical Engineering and Computer Science,
University of Ottawa (uOttawa), Ottawa, ON, Canada
[3] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
stan@eecs.uottawa.ca

Abstract. Clustering is an important data mining task and has been explored extensively by a number of researchers for different application areas, such as text application and bioinformatics data. In this paper we propose the use of a novel algorithm for clustering data that we call hybrid particle swarm optimization with mutation (HPSOM), which is based on PSO. The HPSOM basically uses PSO and incorporates the mutation process often used in GA to allow the search to escape from local optima. It is shown how the PSO/HPSOM can be used to find the centroids of a user-specified number of clusters. The new algorithm is evaluated on five benchmark data sets. The proposed method is compared with the K-means (KM) clustering technique and the standard PSO algorithm. The results show that the algorithm is efficient and produces compact clusters.

Keywords: Data Cluster, PSO, Hybrid PSO, Data Mining.

1 Introduction

Clustering is an important problem that often must be solved as part of more complicated tasks in pattern recognition, image analysis, and other fields of science and engineering. Clustering is one of the main tasks in knowledge discovery from databases (KDD) and consists in finding groups within a certain set of data, where each group contains objects similar to each other and different from those of other groups [1].

In the clustering process, the learning algorithm is provided with just the data points and no labels; the task is to find a suitable representation of the underlying distribution of the data (data vectors are grouped based on distance from one to another). Some approaches are based on hybridization of different clustering techniques and involve optimization in the process.

K-means (KM) algorithm is one of the most popular and widespread partitioning clustering algorithms because of its superior feasibility and efficiency in dealing with a large amount of data. The main drawback of the KM algorithm is that the cluster result is sensitive to the selection of the initial cluster centers and may converge to the local optima [2,3].

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 159–166, 2012.

The particle swarm optimization (PSO) algorithm is an optimization method developed by Eberhart et al. [4,5]. PSO tries to find the optimal solution through the simulation of some ideas drawn from fish schooling, bird flocking, and other social groups. One such idea is that an agent can effectively achieve his objective using the information that is owned by him and the information that is shared among the group. This means that PSO is an optimization method that uses the principles of social behavior. PSO has proved to be competitive with genetic algorithms in several tasks, mainly in optimization areas [5,6].

PSO has been successfully applied in several areas such as clustering problem [2,3], function optimization [6,7] etc. PSO finds the best value with interaction of particle, solves the problem of initialization of the KM algorithm, but it also can trapped in local optima [6,12].

Different variants of the PSO algorithm have been proposed. Some of these variants have been proposed to incorporate the capabilities of other evolutionary algorithms, such as hybrid versions of PSO or the adaptation of PSO parameters, creating the adaptive PSO versions. Many authors have considered incorporating selection, mutation, and crossover, as well as differential evolution, into the PSO algorithm. As a result, hybrid versions of PSO have been created and tested, including a hybrid of genetic algorithm and PSO (GA-PSO), evolutionary PSO (EPSO) [6-10] and hybrid particle swarm optimization with mutation (HPSOM) algorithm [6,7].

In this paper we explore the HPSOM algorithm to solve the PSO stagnation problem and to prevent the particles from being trapped in local minima [6,7]. The main contribution of this paper is to describe a strategy for cluster data by using the HPSOM algorithm and comparing its results with those obtained by KM and standard PSO. Experimental results indicate the superiority of the HPSOM algorithm.

The rest of the paper is organized as follows: Section 2 provides an overview of PSO, Section 3 presents the HPSOM algorithm, Section 4 introduces the HPSOM clustering algorithm, and Section 5 shows the tests performed with the different variants of the algorithm. The conclusions are presented in Section 6.

2 An Overview of Particle Swarm Optimization

The particle swarm optimization algorithm (PSO) is a population-based optimization method that tries to find the optimal solution using a population of particles [4,5]. Each particle is an individual, and the swarm is composed of particles. In PSO, the solution space of the problem is formulated as a search space. Each position in the search space is a potential solution of the problem. Particles cooperate to find the best position (best solution) in the search space (solution space). Each particle moves according to its velocity. At each iteration, the particle movement is computed as follows:

$$x_i(t+1) \leftarrow x_i(t) + v_1(t), \tag{1}$$

$$v_i(t+1) \leftarrow \omega v_i(t) + c_1 r_1 (pbest_i(t) - x_i(t)) + c_2 r_2 (gbest(t) - x_i(t)) \tag{2}$$

In Eqs. (1), (2), $x_i(t)$ is the position of particle i at time t, $v_i(t)$ is the velocity of particle i at time t, $pbest_i(t)$ is the best position found by particle itself so far, $gbest(t)$ is the best position found by the whole swarm so far, ω is an inertia weight scaling the previous time step velocity, c_1 and c_2 are two acceleration coefficients that scale the influence of the best personal position of the particle ($pbest_i(t)$) and the best global position ($gbest(t)$), r_1 and r_2 are random variables within the range [0,1]. The process of PSO is shown as Fig. 1.

```
Initialize a population of particles with random
positions and velocities in the search space.
While (termination conditions are not met)
{
   For each particle i do
   {
      Update the position of particle i according to
      equation (1).
      Update the velocity of particle i according to
      equation (2).
      Map the position of particle i in the solution
      space and evaluate its fitness value according to
      the fitness function.
      Update pbest_i(t) and gbest_i(t) if necessary.
         }
  }
```

Fig. 1. The process of the PSO algorithm

3 The Hybrid PSO with Mutation Algorithm

Since the presentation of PSO [4,5], its performance has been investigated in several papers. The work presented in [11] describes the complex task of parameter selection in the PSO model. Comparisons between PSOs and the standard genetic algorithm (GA) formulation have been carried out in [11], where the author points out that PSO performs well in the early iterations but presents problems in reaching a near-optimal solution.

The behavior of PSO in the *gbest* model presents some important aspects related to the velocity update. If a particle's current position coincides with the global best position, the particle will only move away from this point if its inertia weigh (ω) and previous velocity are different from zero. If their previous velocities are very close to zero, then all the particles will stop moving once they catch up with the global best particle, which may lead to a premature convergence of the algorithm. In fact, this does not even guarantee that the algorithm has converged on a local minimum. It means that all the particles have converged at the best position discovered so far by

the swarm. This phenomenon is known as stagnation [12]. The solution presented in [12] is based on adding a new parameter and additional equations. Another solution is presented in [13] by introducing *breeding and subpopulation*.

In [6] we proposed hybrid particle swarm optimization with mutation (HPSOM) by incorporating the mutation process often used in GA into PSO. The stagnation is alleviated by this technique and introduces diversity into the population. This process allows the swarm to escape from local optima and to search in different zones of the search space.

This process starts with the random choice of a particle in the swarm and moves to different positions inside the search area. The mutation process is implemented by the following equation (3):

$$mut(p_k) \leftarrow -p_k + \beta \tag{3}$$

where, p_k is the random choice *kth* particles from the swarm, and β is randomly obtained within the range $[0, 0.1*(x_{\max} - x_{\min})]$, representing 0.1 times the length of the search space. Comparisons between standard PSO and HPSOM, which show the HPSOM model as better than the standard PSO model, are presented in [6,7].

4 PSO/HPSOM Clustering

Among all the efforts in the literature to modify the particle swarm optimization algorithm for data clustering, [14-15] seem to be the ones closest to the original idea of PSO since each particle comprehends a whole candidate solution to the problem. A particle p_i is constructed as follows:

$$p_i = (m_{i1}, m_{i2}, \ldots, m_{ij}, \ldots, m_{iNc})$$

where Nc is the number of clusters to be formed, and m_{ij} corresponds to the jth centroid of the ith particle, the centroid of the cluster C_{ij}. Thus, a single particle represents a candidate solution to a given clustering problem.

Each particle is evaluated using the following equation (fitness function):

$$f = \frac{\sum_{j=1}^{Nc} [\sum_{\forall X_k \in C_{ij}} d(x_k, m_{ij}) / | C_{ij} |]}{Nc} \tag{4}$$

where x_k denotes the k^{th} data vector, $| C_{ij} |$ is the number of data vectors belonging to the cluster C_{ij}, and d is the Euclidian distance between x_k and m_{ij}.

The stopping criterion (termination conditions) mentioned in the algorithm depends on the type of problem being solved. Usually, the algorithm is run for a fixed number of iterations (objective function evaluations) or until a specified error bound is reached. In this study, the algorithm is stopped when a user-specified number of iterations has been exceeded. The proposed cluster algorithm is shown in Fig 2:

```
Initialize the cluster centroids of each particle randomly  //allocate data to each particle randomly with the centroid vector values
Repeat
 { For each particle i do
   { For each data vector x_k do
     {Calculate the d(x_k, m_ij) to all cluster centroids m_ij
       Assign x_k to cluster C_ij such that:
                 d(x_k, m_ij) = min ∀l = 1,..., Nc {d(xk, mil)}
                  //assign x_k to the cluster C_ij with the minimum distance
                  // (from all the Nc clusters of the particle_i)
      }
   }
 Calculate the fitness using equation (4)
 Update the swarm particles (centroids) as in (Fig 1)
 Execute mutation process (for HPSOM) using equation (3)
Until a stopping criterion is satisfied.
```

Fig. 2. The cluster HPSOM algorithm

5 Results and Discussion

To evaluate the performance of the proposed algorithm, five benchmarks were used: iris, wine, glass and breast cancer, taken from the UCI Repository of Machine Learning Databases [16] and a synthetic data sets (artificial problem). The main features of these benchmarks are presented as follows:

- **Artificial problem:** This problem follows the following classification rule:

$$class = \begin{cases} 1 & if(z_1 \geq 0.7) or((z_1 \leq 0.3) and(z_2 \geq -0.2 - z_1)) \\ 0 & otherwise \end{cases} \quad (5)$$

A total of 500 data vectors were randomly created, with $z_1, z_2 \sim U(-1,1)$.

- **Iris:** This is a well-understood database with 4 inputs (attributes), 3 classes and 150 data vectors (instances).
- **Wine:** This is a classification problem with well-behaved class structures. There are 13 inputs, 3 classes and 178 data vectors.
- **Glass:** This is a classification problem with 9 inputs, 7 classes and 214 data vectors.
- **Breast cancer:** The Wisconsin breast cancer database contains 9 attributes, 2 classes and 286 data vector. The objective is to classify each data vector into benign or malignant tumors.

For each data set, PSO/HPSOM was run 30 times, with 20 iteration, 10 particles, and the parameters ω = 0.72, c_1 = 1.49, and c_2 = 1.49, after several simulations these

parameters ensures a good convergence. The relatively small number of iterations (function evaluations) was chosen due to the high convergence rate of PSO/HPSOM. HPSOM has an additional parameter related to the mutation rate, which was set to 10%. Each benchmark class was represented by the cluster (of the best particle) with the largest number of data of that class; data of different classes within this cluster were considered misclassified. Other measures that can be used to evaluate the performance are the intra-cluster and the intercluster distance.

The intra-cluster distance measures the density of the created clusters, i.e., how compact these clusters are, since the data in the same cluster should be similar. In this work, the intra-cluster distance was measured by the average distances between data in the same cluster. The intercluster distance measures the separation between created clusters, given that the clusters should be as far as possible from each other. Here the intercluster distance was measured by the average distances between the mass centers of the clusters.

Table 1. Comparison of the results by fitness, correctly cluster instances, intra- and intercluster distance

Prob.	Algo.	Fitness Equation	Correctly clustered (%)	Average Intra-cluster	Average Inter-cluster
Iris	KM	0.0842± 0.0035	81.464 ± 6.6106	3.3126±0.247	0.8981±0.092
	PSO	0.0869±0.00484	79.234 ± 6.9871	3.8954±0.183	0.8915±0.87
	HPSOM	**0.08203±0.00289**	**86.037± 5.0426**	**3.0727±0.178**	**0.8532±0.096**
Wine	KM	0.06155± 0.00145	71.217± 0.5254	4.443±0.265	**1.156 ± 0.14**
	PSO	0.05903± 0.00153	68.712± 2.2641	5.143±0.156	2.989+0.203
	HPSOM	**0.00289±0.00148**	**73.872± 0.5725**	**4.185**±0.132	2.789 ±0.187
Glass	KM	0.01502 ±.00260	41.025± 3.7600	1.7903±0.143	3.8945±0.237
	PSO	0.01911± 0.00108	42.205± 5.3687	1.8353±0.129	**3.4551**±0.157
	HPSOM	**0.01442±0.00123**	**44.108 ±4.6933**	**1.6264**±0.121	5.2453±0.109
Breast-Canc.	KM	1.989± 0.064	71.402± 3.013	6.981± 0.324	**1.986 ±0.252**
	PSO	2.606± 0.084	65.140± 4.413	7.571±0.343	3.443±0.216
	HPSOM	**1.795± 0.139**	**73.230 ±5.573**	6.752±0.402	3.295±0.96
Artif.	KM	0.997±0.042	51.183 ± 5.103	3.678±0.087	1.83±0.044
	PSO	0.781±0.028	54.174 ±6.265	3.826±0.89	1.I92±0.51
	HPSOM	**0.772±0.027**	**57.174 ±5.662**	**3.801±0.81**	**1.160 ±0.43**

Table 1 summarize the results obtained from the three clustering algorithms for the benchmark problems above. The values reported are the averages from over 30 simulations, with the standard deviations indicating the range of values at which the algorithms converge. First, consider the fitness of solutions, i.e., the equation (4). For all the problems, the hybrid algorithm had the smallest average quantization error (fitness functions). For the iris and glass problems, KM clustering was not significantly worse than the PSO and HPSOM algorithms (the difference in not high).

However, for the wine problem both KM and the PSO algorithm were significantly worse than the hybrid algorithm (HPSOM is 2 to 3 times better).

When considering the inter- and intra-cluster distances, the latter ensures compact clusters with little deviation from the cluster centroids, while the former ensures larger separation between the different clusters. With reference to these criteria, PSO approaches succeeded most in finding clusters with larger separation than did the KM algorithm; the HPSOM algorithm succeeded in four of the five problems. HPSOM formed the most compact clusters for the five problems. Figures 3 illustrate an example of the clustering found.

Figure 4 summarizes the effect of varying the number of clusters for the three algorithms for the artificial problem. In this case, it is expected that the quantization error should go down when the number of clusters increases. Figure 4 also shows that the HPSOM algorithm consistently performs better than the other two algorithms when the number of clusters increases.

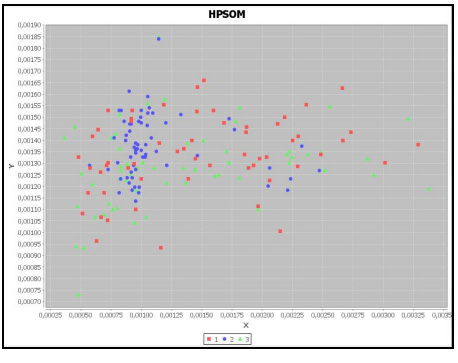


Fig. 3. Clustering found HPSOM Wine

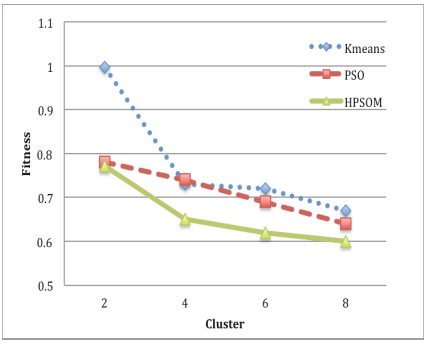


Fig. 4. Effect of the different number of clusters on artificial problem

6 Conclusion

This paper investigates the application of a hybrid PSO called HPSOM to cluster data vectors. Two algorithms were tested, namely, standard PSO and a hybrid PSO approach where the individuals of the swarm are seeded by the result of the KM algorithm. Comparison of the two PSO approaches with KM clustering showed that HPSOM algorithm approaches have better convergence with lower fitness errors and, in general, larger intercluster and smaller intra-cluster distances.

Future works will extend the fitness function to also explicitly optimize the intercluster and intra-cluster distances, and work toward the development of a hybrid PSO clustering method capable of handling large and complex data set and determining the optimal number of clusters.

Acknowledgements. We would like to thank CNPq, FAPEMIG (Brazilian agencies) and NSERC (Canada) for partial financial support. The authors also thank the anonymous reviewers for useful remarks and suggestions.

References

1. Jiawei, H., Micheline, K.: Data Mining, Conecpts and Techniques. Morgan Kaufmann Publishers (2001)
2. Kao, Y.T., Zahara, E., Kao, I.W.: A hybridized approach to data clustering. Expert Systems with Applications 34(3), 1754–1762 (2008)
3. Feng, H.M., Chen, C.Y., Ye, F.: Evolutionary fuzzy particle swarmoptimization vector quantization learning scheme in image compression. Expert Systems with Applications 32(1), 213–222 (2007)
4. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE Internal Conference on Neural Networks, Perth, Australia, vol. 4, pp. 1942–1948 (1995)
5. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
6. Esmin, A.A.A., Lambert-Torres, G., Zambroni de Souza, A.C.: A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization. IEEE Transactions on Power Systems 20(2), 859–866 (2005)
7. Esmin, A.A.A., Lambert-Torres, G.: Fitting Fuzzy Membership Functions using Hybrid Particle Swarm Optimization. In: 2006 IEEE World Congress on Computational Intelligence & IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2006, pp. 9954–9961. IEEE Press, Vancouver (2006)
8. Miranda, V., Fonseca, N.: EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems. In: Proc. of the Asia Pacific IEEE/PES Transmission and Distribution Conference and Exhibition, vol. 2, pp. 745–750 (2002)
9. Fan, S.S., Liang, Y., Zahara, E.: Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions. Engineering Optimization 36(4), 401–418 (2004)
10. Shi, Y., Eberhart, R.: Parameter Selection in Particle Swarm Optimization. In: Proc. 7th Annual Conference on Evolutionary Programming, pp. 591–600 (1998)
11. Angeline, P.: Evolutionary Optimization Versus Particle Swarm Optimization Philosophy and Performance Differences. In: Proc. 7th Annual Conference on Evolutionary Programming, pp. 601–610 (1998)
12. Van den Bergh, F., Engelbrecht, A.P.: A New Locally Convergent Particle Swarm Optimiser. In: Proc. of the IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia (October 2002)
13. Løvbjerg, M., Rasmussen, T.K., Krink, T.: Hybrid Particle Swarm Optimiser with Breeding and Subpopulations. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO), USA (July 2001)
14. van der Merwe, D.W., Engelbrecht, A.P.: Data Clustering using Particle Swarm Optimization. In: Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003), pp. 215–220. IEEE Computer Society, Caribella (2003)
15. Esmin, A.A.A., Pereira, D.L., Araujo, F.P.A.: Study of Different Approach to Clustering Data by Using The Particle Swarm Optimization Algorithm. In: 2008 IEEE Congress on Evolutionary Computation (IEEE CEC 2008), Proceedings of IEEE Congress on Evolutionary Computation, Hong Kong, pp. 1817–1822 (2008)
16. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, Department of Information and Computer Science, Irvine (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

A Learning-to-Rank Algorithm for Constructing Defect Prediction Models

Xiaoxing Yang[1], Ke Tang[1], and Xin Yao[1,2]

[1] NICAL, Joint USTC-Birmingham Research Institute in Intelligent Computation and Its Application, School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China
apricot@mail.ustc.edu.cn, ketang@ustc.edu.cn
[2] CERCIA, School of Computer Science, University of Birmingham, U.K.
X.Yao@cs.bham.ac.uk

Abstract. This paper applies the learning-to-rank approach to software defect prediction. Ranking software modules in order of defect-proneness is important to ensure that testing resources are allocated efficiently. However, prediction models that are optimized for predicting explicitly the number of defects often fail to correctly predict rankings based on those defect numbers. We show in this paper that the model construction methods, which include the ranking performance measure in the objective function, perform better in predicting defect-proneness rankings of multiple modules. We present the experimental results, in which our method is compared against three other methods from the literature, using five publicly available data sets.

Keywords: Learning-to-rank, software defect prediction, differential evolution.

1 Introduction

Software defect prediction employs the attributes of software modules to predict their defect-proneness in order to support software testing activities[1-5]. We focus on the defect prediction models with the goal of predicting a ranking of the modules based on their defects[2-5]. To illustrate this task clearly, we give a simple example in Table 1. LinesOfCode, PreviousDefects and LinesAdded are the attributes (independent variables), according to which we obtain data from the software modules. We use modeling approaches such as generalized linear regression[2-3] to construct a model based on the data from A, B and C (known defect number (dependent variable)), and then use the model to predict D and E (unknown defect number). Thus we can obtain a ranking of D and E based on the prediction and allocate testing resources according to the ranking (for instance, more resources for modules with more defects).

For this kind of defect prediction, it is the ranking of the modules according to the defects instead of the specific number of defects that is important and helpful for guiding the assignment of testing resources, so models are usually used as ranking models[2]. However, most existing methods obtain defect prediction

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 167–175, 2012.

Table 1. A Simple Example of Defect Prediction

ModuleName	LinesOfCode	PreviousDefects	LinesAdded	DefectNumber
A	456	4	45	4
B	123	1	0	0
C	156	1	23	2
D	321	3	23	unknown
E	211	2	45	unknown

models by optimizing for predicting explicit defect numbers such as maximum likelihood or minimizing least square errors, etc. This might cause the potential problem that a good model according to these indirect loss functions could not give a good ranking according to the ranking performance measures such as the fault-percentile-average[2]. Therefore, we investigate constructing models by directly optimizing the ranking performance measures in order to improve the general performance of the models. Considering the effectiveness of linear models[2-3][6], we investigate constructing a linear defect prediction model. In order to directly optimize the non-differentiable model performance measure, we use the optimization approaches that are able to deal with non-differentiable loss functions, to obtain the coefficients of the linear models. To be specific, we use composite differential evolution (CoDE)[7] as the optimization approach. We compare our approach with three popular approaches in the literature, including generalized linear regression[2-3], logistic regression[4] and random forest[2] over five public datasets. The main contributions of this paper include first application of the learning-to-rank approach in defect prediction and a comparison study of four model constructing algorithms over five public datasets.

2 The Learning-to-Rank Approach

Our goal is to build a defect prediction model by directly optimizing the model performance measure in order to give a better ranking of the software modules based on their defects.

2.1 Dependent and Independent Variables

Normally, the dependent variable is the defect number. However, what we need is the order of the modules according to the defect number instead of the explicit defect number. Therefore, the dependent variable in this paper is the relative defect number, which can be a real number.

Many previous studies[2-5][8-12] are based on private data. In order to facilitate others to reproduce the results, we use the public benchmark[1] presented in[13], including data over a five year period from five open-source software

[1] http://bug.inf.usi.ch/

systems—Eclipse JDT Core (eclipse), Eclipse PDE UI (pde), Equinox framework (equinox), Mylyn, and Apache Lucene (lucene). Because the change metrics have been found to be very effective for constructing defect prediction models[13-15], we use the change metrics of the five systems. The independent variables contain the number of revisions, number of times a file has been refactored, number of times a file was involved in bug-fixing, number of authors who committed the file, lines added, max lines added, average lines added, lines removed, max lines removed, average lines removed, Codechurn, max Codechurn, average Codechurn, age and weighted age. All bugs are seen as the predicted defects and seen as equally important. Detailed information of the five systems is given in Table 2.

Table 2. Information of the Five Systems

System (url)	Prediction Release	Time Period	File Number	Defect Number
Eclipse JDT Core(www.eclipse.org/jdt/core/)	3.4	1.01.2005-6.17.2008	997	369
Eclipse PDE UI (www.eclipse.org/pde/pde-ui/)	3.4.1	1.01.2005-9.11.2008	1497	306
Equinox framework (www.eclipse.org/equinox/)	3.4	1.01.2005-6.25.2008	324	241
Mylyn (www.eclipse.org/mylyn/)	3.1	1.17.2005-3.17.2009	1862	339
Apache Lucene (lucene.apache.org)	2.4.0	1.01.2005-10.08.2008	691	97

2.2 Our Approach

Given a vector of independent variables of the software module $\mathbf{x}=(x_1, x_2, ..., x_d)$ (x_i: the i^{th} independent variable value and d: number of independent variables), the goal of defect prediction models is to predict its relative defect number, which is denoted as $f(\mathbf{x})$. For the effectiveness of linear models for ranking problems[2-3][6] and the suitability of linear and additive models for defect prediction[2], we study the simple linear model:

$$f(\mathbf{x}) = \sum_{i=1}^{d} \alpha_i x_i \tag{1}$$

where α_is are the corresponding coefficients obtained by training. Once the α_is are fixed, the model is fixed.

The coefficients are often obtained by optimizing functions such as least square errors, which are related with the explicit defect number. The potential problem is that a good regression model according to these functions might not give a good ranking, because they focus on the fitting of each sample instead of the whole ranking. A simple example is: we are predicting modules A and C in Table 1. One model predicts that A and C have respectively 1 and 2 defects, and the other model predicts that A and C have respectively 1 and 0 defects. The former model might be better according to the least square errors, although the latter model is more desired because it could give the expected ranking. Therefore, we try to directly optimize the performance of defect prediction models to find

the coefficients. Since the performance measures are usually non-differentiable, we employ intelligent optimization, which can deal with non-differentiable functions. Differential evolution (DE)[16] is one of the famous intelligent optimization methods and exhibits noticeable performance in a wide variety of problems. As a population-based stochastic search technique, it uses selection, crossover and mutation operators to move the population toward the optimum point. One of the improvements of DE is proposed lately by Wang et al.[7]—CoDE. It combines several trial vector generation strategies with suitable control parameter settings and overcomes the troubles caused by parameter settings of previous differential evolutions. Therefore, we use CoDE to obtain the coefficients of the linear models. The details are as follows:

1. **Input**: training vectors: $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{id})$, i: 1 to M, d: number of attributes, M: number of training vectors; objective function: **fault-percentile-average**
2. **Initialize**: Set the population size (number of solutions v_is ($v_i = (\alpha_1, \alpha_2, ..., \alpha_d)$) at each generation) to N. Randomly generate N solutions that compose P_0 and compute their objective function values. Set the generation number $t = 0$.
3. **While** $t < t_{max}$ (maximal generation number to decide termination), **do**
   (a) **For each solution** v_i **in** P_t, do crossover and mutation to generate new solutions (trial vectors). Three generation strategies include
      i. **rand/1/bin**:

$$u_{i,j,G} = \begin{cases} \alpha_{r1,j,G} + F \cdot (\alpha_{r2,j,G} - \alpha_{r3,j,G}), & if\ rand < C_r, or j = j_{rand} \\ \alpha_{i,j,G}, & otherwise \end{cases} \quad (2)$$

      ii. **rand/2/bin**:

$$u_{i,j,G} = \begin{cases} \alpha_{r1,j,G} + F \cdot (\alpha_{r2,j,G} - \alpha_{r3,j,G}) + F \cdot (\alpha_{r4,j,G} - \alpha_{r5,j,G}), \\ if\ rand < C_r\ or\ j = j_{rand} \\ \alpha_{i,j,G}, \quad otherwise \end{cases} \quad (3)$$

      iii. **current-to-rand/1**

$$\boldsymbol{u}_{i,G} = \boldsymbol{\alpha}_{i,G} + rand \cdot (\boldsymbol{\alpha}_{r1,G} - \boldsymbol{\alpha}_{i,G}) + F \cdot (\boldsymbol{\alpha}_{r2,G} - \boldsymbol{\alpha}_{r3,G}) \quad (4)$$

   where $\boldsymbol{u}_{i,G}$ is a new trial vector, r1-r5 are distinct integers randomly selected from the range [1, N],*rand* denotes a uniformly distributed random number between 0 and 1, C_r is the crossover control parameter and the first scaling factor $\boldsymbol{F}$ in Equ.(3) is randomly chosen from 0 to 1. Three control parameter settings are $[F = 1.0, C_r = 0.1]$,$[F = 1.0, C_r = 0.9]$ and $[F = 0.8, C_r = 0.2]$. Each generation strategy creates a new vector with the control parameter randomly chosen from three settings. Thus three vectors are generated for v_i. The best vector among the three vectors and v_i according to the objective function value is saved as the solution in the new population P_{t+1}.
   (b) **t++**.
4. **Return** the best solution in P_{t+1} and **output** the model: $f(\mathbf{x}) = \sum_{i=1}^{d} \alpha_i x_i$.

Considering k modules $f_1, f_2, \ldots, f_k$, listed in increasing order of predicted defect number, n_i as the actual defect number in the module i, and $n = n_1 + n_2 + ... + n_k$ as the total number of defects, the proportion of actual defects in the top

m predicted modules to the whole defects is $\frac{1}{n}\sum_{i=k-m+1}^{k} n_i$. Then **the fault-percentile-average** is defined as follows[2]:

$$\frac{1}{k}\sum_{m=1}^{k}\frac{1}{n}\sum_{i=k-m+1}^{k} n_i$$

Actually, the fault-percentile-average is the average of the proportion of actual defects in the top i(i: *1* to k) predicted modules, so larger fault-percentile-average means better performance.

3 Experimental Studies

3.1 Methodology

In order to evaluate the performance of the learning-to-rank approach, we compare it with three popular methods: generalized linear regression[13], logistic regression[4] and random forest[2]. Our approach is implemented using Java while the other methods are implemented based on WEKA[17]. Following the work of[2], *fault-percentile-average* and the percentages of defects contained in the top 20% of modules are used as the model performance measures. As pointed above, *fault-percentile-average* is also used as the objective function for CoDE. Detailed methodology is as follows:

Preprocessing the data: Useless attributes such as "name" and modules with 0 line of code are deleted and the whole data is normalized using "Normalize" in WEKA.

Parameter setting: For the learning-to-rank approach, the feasible solution space is set to $\Omega = \prod_{i=1}^{d}[-20, 20]$, and the population size and maximal generation are set to 100. For linear regression and random forest, we use 10-folds cross-validation to find the optimal parameters. The adjustable parameters for random forest are the number of trees (10, 50, 100, 250, 500, 1000) and the number of attributes ((0.5, 1, 2)*$\sqrt{d}$, where d is the total attribute number)). Parameters for linear regression include attribute-selection (false or true) and the attribute selection method (no method, M5 (step through the attributes removing the one with the smallest standardized coefficient until no improvement is observed in the estimate of the error given by the Akaike information criterion)[17], greedy method). The parameters for logistic regression are set as default.

Ten-folds cross-validation: We do 10-fold cross-validation to evaluate the methods. That is, we split the dataset in 10 folds and use 9 folds as training set to build the prediction model and the remaining fold as a validation set to evaluate the models, with each fold used once as a validation set. We conduct 10 times of 10-folds cross-validation.

3.2 Results

Table 3 shows the mean calculated over 100 testing fault-percentile-average results in 10 times of 10-folds cross validation. We also record the corresponding training

results (using 9 folds of data for both training and testing) in Table 3. In order to further compare the results, Wilcoxon rank-sum test at the 0.05 significance level is used to assess the relationship between our approach and other methods. The letters "W" or "E" after the number respectively means that the corresponding testing results (only testing results are useful for comparison) are statistically worse (smaller) than or equal to that of the learning-to-rank approach.

Table 3. Mean Fault-Percentile-Average for 100 Testing and Training Results. ("W" or "E": Worse or Equal Testing Results for that Approach Compared to Our Approach According to Wilcoxon Rank-Sum Test.)

datasets	Learning -to-Rank		Generalized Linear Regression			Logistic Regression			Random Forest		
	testing	training	testing	training		testing	training		testing	training	
eclipse	0.8162	*0.8341*	0.8110	*0.8273*	E	0.8110	*0.8382*	E	0.8149	*0.9162*	E
pde	0.7455	*0.7722*	0.7398	*0.7584*	E	0.7339	*0.7629*	W	0.7335	*0.9353*	W
equinox	0.7578	*0.7754*	0.7354	*0.7625*	W	0.7508	*0.7760*	E	0.7606	*0.8254*	E
lucene	0.8180	*0.8700*	0.7824	*0.8380*	W	0.7963	*0.8757*	E	0.8281	*0.9589*	E
mylyn	0.7858	*0.8037*	0.7642	*0.7806*	W	0.7724	*0.7888*	W	0.7864	*0.9376*	E

It can be seen from Table 3 that the learning-to-rank approach is statistically better than or the same as the three methods for all datasets, which implies the usefulness of the learning-to-rank approach for defect prediction. The generalized linear regression and logistic regression have very similar results, which are inferior to random forest. This seems to conflict with the conclusion by comparison study in[2], which shows that the generalized linear regression and random forest have similar results. However, for two systems in[2], random forest has better average percentile of faults. Therefore, the results are actually consistent. In addition, we tune more parameters for random forest, which might give more chances for it to obtain better results.

Random forest has almost the same testing results as our approach but larger training fault-percentile-average results. This accords with our intuition: simple models reduce the chance of over-fitting on training data. With the above experimental setting, random forest takes longer time than our approach to build models because the best number of trees for random forest are large. With fixed 500 trees , which was the best number of tree in[2], and 8 attributes for random forest, the model building time ratio of random forest to the learning-to-rank approach (100 for both population size and maximal generation) is about three to one. A drawback of random forest is its difficulty to interpret the model. Contrarily, it is easy to tell the important factors from our linear models. From the coefficients of the defect prediction models constructed by our approach, we find that the first, fifth, eighth, ninth and twelfth coefficients are larger for most models, and the first coefficient is large for all models on all datasets. This means that number of revisions, lines added, lines removed, max lines re-moved and max Codechurn play a significant role in deciding the models, especially the number of revision.

Table 4. Mean Calculated over 100 Testing Results of Defect Percentages Contained in the Top 20% of Modules. ("W", "E" or "B": Worse, Equal or Better Results for that Approach Compared to Our Approach According to Wilcoxon Rank-Sum Test.)

datasets	Learning -to-Rank	Generalized Linear Regression	Logistic Regression	Random Forest
eclipse	0.6636	0.6473 E	0.6347 W	0.6358 W
pde	0.5424	0.4896 W	0.5068 E	0.4829 W
equinox	0.5376	0.4902 W	0.5245 E	0.5061 E
lucene	0.6601	0.6240 W	0.6354 E	0.6571 E
mylyn	0.5534	0.5124 W	0.5703 E	0.5978 B

Because of the practical significance of the percentages of defects contained in the top 20% of modules, we record the average defect percentages contained in the top 20% of modules in Table 4. While the fault-percentile-average focuses on the whole ranking given by the model, this measure focuses on the ranking of the former 20% of modules. The results are not as good as that in [2], where more than 76% of defects were contained in the top 20% of modules for three commercial sub-systems, which implies difficulties for different systems are different. Nevertheless, our approach can predict over 54% of defects in the top 20% of modules for all datasets, which is also appealing. From the Table 4, the learning-to-rank models optimized according to the fault-percentile-average are still better than other models in most cases. The exception is for mylyn. The letters "B" means that the corresponding results are statistically better (larger) than that of our approach. This shows that models with good fault-percentile-average still might have comparably worse defect percentages contained in the top 20% of modules. When the specific measure is known, it is better to directly optimize the measure.

4 Related Work

Ohlsson and Alberg[3] showed that linear regression models, based on design data, could successfully predict the number of defects in software modules. Denaro and Pezze[4] constructed a variety of logistic regression models and found that suitably selected multivariate models could successfully predict defect-proneness. D'Ambors et al.[13] presented a benchmark data sets and a comparison study, including prediction of rankings as well as defect-proneness. They compared several sets of variants and constructed models by linear regression. These studies focused more on the choice of module attributes, e.g. design metrics or historical changes, while in this paper we focused instead on the methods for model construction.

Closely related work includes that of Ostrand, Bell and Weyuker[2][8-12], who constructed models to identify the modules with most defects. They applied several kinds of generalized linear models and also a very simple model based on

lines of code to predict the defect number or defect density for three commercial systems and found that about 80% of defects were contained in the top 20% of the files identified by the models. In [2], they presented a benchmark that compared the effectiveness of four modeling methods for predicting the files likely to contain the most defects and found that the generalized linear model performed well for predicting defects, thus supporting the effectiveness of linear and additive models. Our work is different from theirs in three ways: firstly, they focus more on the data collection and the choice of metrics, while we focus instead on the model constructing approach; secondly, their data sets are derived from commercial systems and are not public while we have used public data sets so that our experiments can be reproduced; finally, they optimized their models to best predict defect numbers, whereas we directly optimized our models to best predict ranking, which led to improved ranking predictions.

5 Conclusions

Predicting the order of modules according to defect number is very important because it can help to allocate the testing resources effectively and efficiently although testing resources allocation itself is a separate study[18]. Most methods take it as a regression problem and try to obtain models by optimizing some differentiable loss functions. The potential problem is that a good model according to those loss functions might not be good according to the ranking performance, which is important for defect prediction models. Therefore, we use the evolutionary optimization method to directly optimize the model performance measure—fault-percentile-average, which is different from those loss functions. Compared with linear regression and logistic regression, the proposed learning-to-rank method can obtain better models according to the fault-percentile-average for most datasets. It has similar results to random forest for most datasets. However, our model is simpler and easier to interpret. Comparing the defect percentages in the top 20% of modules of these models, we found that the models trained by learning-to-rank are better for four datasets except for mylyn. This might imply that fault-percentile-average is consistent with defect percentages in the top 20% of modules in most cases, but not absolutely.

Although CoDE[7] was used in this paper, other meta-heuristic algorithms[19-20] could also be used. This is left as future work. The other direction of future research is to evaluate our approach on more real-world data sets and analyze its performance in depth.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant No 61028009 and 61175065, the Natural Science Foundation of Anhui Province under Grant No 1108085J16, the European Union 7th Framework Program under Grant No 247619, and an EPSRC Grant (No. EP/J017515/1).

References

1. Briand, L.C., Wuest, J.: Empirical Studies of Quality Models in Object-Oriented Systems. Advances in Computers 59, 97–166 (2002)
2. Weyuker, E.J., Ostrand, T.J., Bell, R.M.: Comparing the Effectiveness of Several Modeling Methods for Fault Prediction. Empir. Software Eng. 15, 277–295 (2010)
3. Ohlsson, N., Alberg, H.: Predicting Fault-Prone Software Modules in Telephone Switches. IEEE Transactions on Software Engineering 22(12), 886–894 (1996)
4. Denaro, G., Pezze, M.: An Empirical Evaluation of Fault-Proneness Models. In: Proc. of the 24th International Conf. on Software Engineering, pp. 241–251 (2002)
5. Succi, G., Pedrycz, W., Stefanovic, M., Miller, J.: Practical Assessment of the Models for Identification of Defect-Prone Classes in Object-Oriented Commercial Systems Using Design Metrics. J. Syst. Softw. 65(1), 1–12 (2003)
6. Karimzadehgan, M., Li, W., Zhang, R., Mao, J.: A Stochastic Learning-To-Rank Algorithm and its Application to Contextual Advertising. In: Proceedings of the 20th International Conference on World Wide Web, pp. 377–386 (2011)
7. Wang, Y., Cai, Z., Zhang, Q.: Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters. IEEE Transactions on Evolutionary Computation 15(1), 55–66 (2011)
8. Ostrand, T.J., Weyuker, E.J., Bell, R.M.: Where the Bugs Are. In: Proc. ACM/ISSTA 2004, pp. 86–96 (2004)
9. Ostrand, T.J., Weyuker, E.J., Bell, R.M.: Predicting the Location and Number of Faults in Large Software Systems. IEEE Transaction on Software Engineering 31(4), 340–355 (2005)
10. Bell, R.M., Ostrand, T.J., Weyuker, E.J.: Looking For Bugs in All the Right Places. In: ISSTA 2006, pp. 61–72 (2006)
11. Ostrand, T.J., Weyuker, E.J., Bell, R.M.: Automating Algorithms for the Identification of Fault-Prone Files. In: ISSTA 2007, pp. 219–227 (2007)
12. Weyuker, E.J., Ostrand, T.J., Bell, R.M.: Comparing Negative Binomial and Recursive Partitioning Models for Fault Prediction. In: Proceedings of the 4th International Workshop on Predictor Models in Software Engineering, pp. 3–10 (2008)
13. D'Ambros, M., Lanza, M., Robbes, R.: Evaluating Defect Prediction Approaches: A Benchmark and an Extensive Comparison. Emp. Soft. Eng., 1–47 (2011)
14. Graves, T.L., Karr, A.F., Marron, J.S., Siy, H.: Predicting Fault Incidence Using Software Change History. IEEE Trans. on Soft. Eng. 26(7), 653–661 (2000)
15. Arisholm, E., Briand, L.C.: Predicting Fault-prone Components in a Java Legacy System. In: Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, pp. 8–17 (2006)
16. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–359 (1997)
17. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Expl. 11(1), 10–18 (2009)
18. Wang, Z., Tang, K., Yao, X.: Multi-objective Approaches to Optimal Testing Resource Allocation in Modular Software Systems. IEEE Transactions on Reliability 59(3), 563–575 (2010)
19. Yang, Z., Tang, K., Yao, X.: Scalability of Generalized Adaptive Differential Evolution for Large-Scale Continuous Optimization. Soft. Computing 15(11), 2141–2155 (2011)
20. Li, X., Yao, X.: Cooperatively Coevolving Particle Swarms for Large Scale Optimization. IEEE Transactions on Evolutionary Computation 16(2), 210–224 (2012)

Multilayer Perceptrons as Classifiers Guided by Mutual Information and Trained with Genetic Algorithms

Antonio Neme, Sergio Hernández, Antonio Nido, and Carlos Islas

Universidad Autónoma de la Ciudad de México
San Lorenzo 290, México, D.F. México
neme@nolineal.org.mx, cheko@fciencias.unam.mx

Abstract. Multilayer perceptrons can be trained with several algorithms and with different quantities that correlate the expected output and the achieved state. Among the most common of those quantities is the mean square error, but information-theoretic quantities have been applied with great success. A common scheme to train multilayer perceptrons is based in evolutionary computing, as a counterpart of the commonly applied backpropagation algorithm. In this contribution we evaluated the performance of multilayer perceptrons as classifiers when trained with genetic algorithms and applying mutual information between the label obtained by the network and the expected class. We propose a classification algorithm in which each input variable is substituted by a function of it such that mutual information from the new function to the label is maximized. Next, those approximated functions are fed as input to a multilayer perceptron in charge of learning the classification map, trained with genetic algorithms and guided by mutual information.

Keywords: classification, multilayer perceptrons, mutual information, genetic algorithms.

1 Introduction

Multilayer perceptrons (MLP) are general approximators [1]. However, unfortunately existence proof does not offer a constructive mechanism. Several algorithms have been proposed to train them, and also several quantities that reflect the behavior of the MLP with respect to the expected output.

One of the most applied guiding quantities that evaluate the performance of, for example, classification is the mean square error (MSE). This is a good choice when data is supposed to be described by a Gaussian distribution and first and second order momentum are enough to describe the distribution. However, many problems in pattern classification are described by relations more complex and subtle than first and second order momentum between attributes and labels. An alternative to the MSE should be able to detect non-linear correlations between random variables (input and class, for example).

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 176–183, 2012.

In fig. 1 it is shown a general panorama of an adaptive system F able to modify some of its parameters guided by a feedback quantity Q. That quantity is mainly represented by MSE, but in a group of recent proposals, Principe et al (see [2,3]) have considered the use of guiding quantities from information theory.

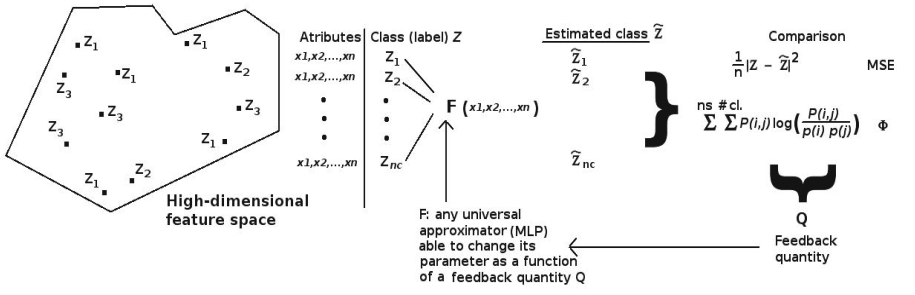


Fig. 1. A general perspective of an adaptive system able to modify some of its parameters guided by a feedback quantity that measures some aspects of the system's output and the expected output

In this contribution, we test the capabilities of MLP as classifiers when the guiding quantity of performance is mutual information between the estimated class and the actual label, but with one major modification. Attributes are substituted by implicit functions of them such that the guiding quantity is maximized. Since Shannon's seminal paper [4] all fields in which data analysis pervades have been benefited with the gradual substitution of first and second momentum statistics with concepts related to mutual information. In next section we describe the details of the model, and then experiments and results are presented.

2 The Proposed Algorithm

We propose a two-stage classification algorithm guided by mutual information and trained by genetic algorithms. In the first stage, each input variable is replaced by a function of that variable that maximizes a certain quantity (mutual information). In the second stage, a multilayer perceptron is trained with the function of the variables obtained in the previous stage, and the MLP intends to maximizes mutual information between its output and the expected results. We call this algorithm *Multilayer perceptron fed with unfolded variables and trained with genetic algorithms and guided by mutual information* (MLPfXGAgMI for short). We proceed to the detailed description of MLPfXGAgMI.

First, the guiding quantity that tells the MLP how well it is doing its classification task is based in the concept of entropy. The entropy H of the training set is defined as:

$$H = -\sum_{i=1}^{\#classes} p_i log(p_i) \tag{1}$$

where the number of classes is defined as $\#classes$, and p_i is the probability of randomly chose an input vector whose class is i. The mutual information between two random variables quantifies how much information is gained about the possible state one of them once we know the actual state of the other variable. It is a measure of correlation [5]. Mutual information between two random variables X and Z is expressed as $\Phi(X;Z)$, where X is in this work one of the attributes of the input vectors and Z is the class or label of those vectors. It is defined as:

$$\Phi(X;Z) = \sum_{i}^{\#statesinX} \sum_{j}^{\#statesinZ} P(i,j) log \frac{P(i,j)}{P(i)P(j)} \tag{2}$$

For the case of classification, the number of states in Z is just the number of classes, and ns is the number of states in X. If X is a continuous variable, then it can be discretized into ns different states. In information-theoretic algorithms the guiding quantity to be minimized is the Renyis Quadratic Entropy of the error between the output of the system and the actual label [6].

In general, for artificial datasets with no noise, all entropy in the label can be removed from the list of attributes $\bar{X}$ that define the high-dimensional feature space. That is, $\Phi(\bar{X};Z) = H$. Mutual information between the compound system of all attributes or variables and Z ($\Phi(\bar{X};Z)$) tends to disipate all entropy in the label. That is, when $ns \to \infty$, $\Phi(\bar{X};Z) \to H$. When the guiding quantity of an adaptive system F (such MLP) is the mutual information, we are considering high-order momentum able to capture non-linear correlations in data [7,5].

A system F is trained such that $\Phi(F(X_1,...,X_n);Z)$ is maximized. F can be, for example, a MLP trained with retropropagation-like algorithms as proposed by Príncipe [2], and systematically studied also by Marqués da Sá[7] and several other groups.

The main idea behind our proposal is described as follows. The mutual information between a variable X_i and the target variable or class Z is expressed by $\Phi(X_i;Z)$. We are interested in finding a function F_i such that $\Phi(F_i(X_i);Z) >= \Phi(X_i;Z)$. Once we have computed F_i for each X_i, we train a MLP such that: $\Phi(F(F_i(X_1),...,F_n(X_n));Z)$ is maximized. Note that now the system does not work directly with the input variables but with mapped versions of them.

In this contribution, we approximate F also with a MLP, but the training algorithm is a genetic algorithm. There is an extensive debate about the performance of genetic algorithms (and evolutionary algorithm in general) vs. retropropagation for MLP training but we are not interested in such debate here. We simply specify that genetic algorithm (ga) are valuable schemes for MLP training.

As a first step, we replace each variable X_i with a function of it $F_i(X_i)$ such that $\Phi(F_i(X_i);Z) >= \Phi(X_i;Z)$. In words, this means that we want to unveil X_i to increase the information it maintains about Z. This is related to feature extraction [8], but applied only to one attribute at the time. We obtain independently F_i for all input variables X_i, and this can of course shade the relevance of other variables. But, as we show in the experiments section, results are encouraging.

As a simple example, define $F = cos(X) \times Y$ for $x \in [-1, 1], y \in [-1, 1]$. Mutual information between X and F is $\Phi(X, F) = 0.454$, whereas $\Phi(cos(X), F) = 0.616$ when the number of states is 10. Indeed, for any number of states greater that 1, $\Phi(cos(X), F) > \Phi(X, F)$. This is the heuristics behind our proposal, that a transformed version of a variable (attribute) may help in the classification task as it reduces the entropy of the class. Equivalently, an increase in mutual information between a transformed attribute and the class is observed.

We define a new set of variables $\tilde{X}$ in an iterative process. We start with the most informative variable $X_g(\arg\max_i \Phi(X_i; Z))$ and obtain F_g. Then, we continue with the second-most informative variable k and we approximate an F_k such that $\Phi(F_k(X_k); Z)$ is maximal. We continue this iteration process until the less informative variable is included. If there is no F_r such that $\Phi(F_r(X_r); Z) > \Phi(X_r; Z)$, then X_r is not substituted and the compound system is constructed with X_r instead of $F_r(X_r)$. The algorithm to construct the compound system is sketched in fig. 2.

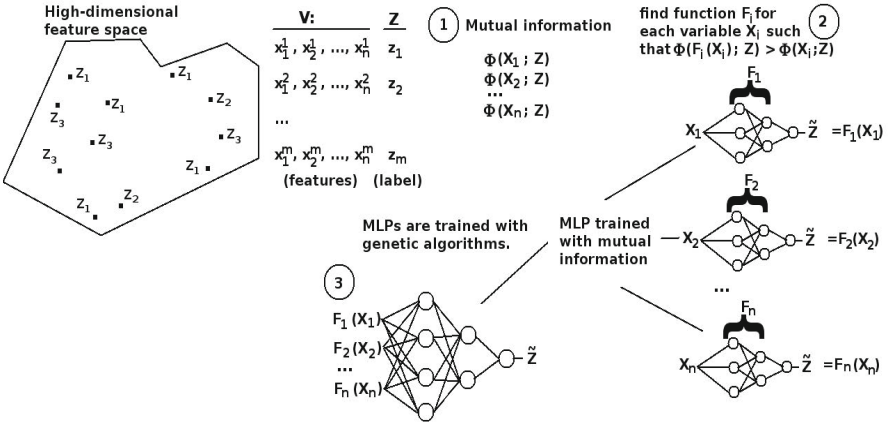


Fig. 2. The sketch of the proposed algorithm MLPfXGAgMI. In the initial step, mutual information between each variable X_i and labels are obtained, $\Phi(X_i; Z)$. Next, each variable is substituted by a new variable $F_i(X_i)$ such that $\Phi(F_i(X_i); Z) > \Phi(X_i; Z)$. The function F_i is approximated by a MLP trained with genetic algorithms and guided by mutual information maximization. In the last step, a MLP is fed with the transformed variables $F_i(X_i)$, and trained also by a genetic algorithm and guided by maximization of mutual information between the output and the actual label.

Note that $\tilde{X}$ is the list that contains the new variables that will replace the list of original variables $\bar{X}$. Now a MLP will be trained with $\tilde{X}$ and the quantity that will evaluate the performance of MLP is not MSE, but $\Phi(MLP(\tilde{X}); Z)$. The evaluation of a solution is dictated by mutual information between the obtained class and the expected class.

3 Experiments

Several datasets from the UCI repository [9] were analyzed. The iris, ionosphere, red wine, cardiotocography, liver and breast tissue datasets were selected and in table 1 a general description of the analyzed datasets is presented. In all cases, entropy and mutual information is expressed in bits (log_2).

Table 1. Datasets for classification

	Iris	Ionosphere	Red wine	Cardiotocography	Liver	Breast tissue
Entropy (H)	1.098	0.652	1.184	2.019	0.681	1.777
MIF (ns = 3)	0.428	0.048	0.032	0.171	0.014	0.688
MIF (ns = 30)	1.054	0.521	0.698	1.091	0.586	1.569
dimension	4	34	11	35	6	9
No. classes	3	2	6	10	2	6
no. vectors	150	351	1599	2126	345	106

The evaluation of classifiers was under a regression approach, meaning that the order of the class is relevant. To evaluate the classification achieved by MLP we applied two error measures. The first one is the mean absolute deviation (MAD):

$$MAD = (1/N) \sum_{i}^{vectors} |Z_i - \tilde{Z}_i| \tag{3}$$

Where Z_i is the class of input vector i and $\tilde{Z}_i$ is the class estimated by the system, and the second error measure is the fraction of misclassified vectors (MV):

$$MV = (1/N) \sum_{i}^{vectors} \delta(Z_i, \tilde{Z}_i) \tag{4}$$

Where $\delta(a, b)$ is the Kronecker delta. MAD and MV do not necessarily coincide when the number of classes is greater than two, as MAD measures how distant is the estimated class from the actual class, whereas MD measures the number of misclassified vectors. For all datasets, 2/3 of the input vector constituted the training set and the remaining 1/3 was the test set. All reported results are referred to test sets.

Tables 2 - 4 shows the results of applying MLPfXGAgMI over the datasets described in table 1. Different schemes are presented: backpropagation guided by MSE, genetic algorithms guided by MSE, genetic algorithms guided by mutual information (MLPGAgMI), and the studied proposal MLPfXGAgMI. Each MLP consisted of two hidden layers and one output, and three network sizes were considered: 5×5, 10×5, and 15×10. For each training scheme and network size, 20 simulations were performed and the average is reported in the tables. We included for comparison MLP trained with backpropagation guided by MSE, MLP trained with genetic algorithms guided by MSE, MLP trained with genetic

algorithms and guided by MI, and finally, MLPfXGAgMI. For backpropagation, the training stage consisted of 1000, 1500 and 2000 iterations for the three network sizes and momentum of 0.6 and learning parameter of 0.1. For the training conduced by genetic algorithms, a population of 120 individuals and 1000, 1500 and 2000 epochs were executed, for the corresponfing three network sizes. In all cases, probability of mutation was settled to 0.05, and probability of crossover was settled to 0.9. For the MI-guided schemes, several number of states (ns) for attribute discretization were considered (see fig. 3) and in the tables the average over all ns is presented.

Table 2. Avg. errors for the six data sets and for network size 5×5. Training was unfolded in 1000 iterations. (or epochs)

	Iris		Ion		Red wine		Cardio		Liver		Breast tissue	
	MV	MAD	MV	MAD	MV	MAD	MV	MAD	MV	MAD	MV	MAD
MSE Backprop	0.067	0.085	0.131	0.161	0.445	0.096	0.005	0.003	0.319	0.319	0.413	0.104
MSE GA	**0.064**	**0.079**	0.138	0.158	0.438	0.096	**0.004**	**0.003**	0.301	0.301	0.406	0.102
MLPGAgMI	0.130	0.095	0.093	0.093	**0.431**	**0.093**	0.006	0.0028	0.304	0.304	0.398	0.102
MLPfXGAgMI	0.145	0.096	**0.092**	**0.092**	0.432	0.093	0.0053	0.003	**0.292**	**0.292**	**0.384**	**0.098**

Table 3. Avg. errors for the six data sets and for network size 10×5. Training was unfolded in 1500 iterations. (or epochs)

	Iris		Ion		Red wine		Cardio		Liver		Breast tissue	
	MV	MAD	MV	MAD	MV	MAD	MV	MAD	MV	MAD	MV	MAD
MSE Backprop	0.066	0.089	0.138	0.138	0.458	0.099	0.0060	0.009	0.318	0.318	0.416	0.108
MSE GA	**0.066**	**0.083**	0.134	0.134	0.455	0.098	0.0062	0.0058	0.315	0.315	0.418	0.109
MLPGAgMI	0.102	0.093	**0.092**	**0.092**	0.443	0.0935	0.0067	**0.0056**	0.307	0.307	0.397	0.103
MLPfXGAgMI	0.113	0.097	0.168	0.168	**0.439**	0.097	**0.0059**	0.0067	**0.283**	**0.283**	**0.389**	**0.094**

Table 4. Avg. errors for the six data sets and for network size 15×10. Training was unfolded in 2000 iterations. (or epochs)

	Iris		Ion		Red wine		Cardio		Liver		Breast tissue	
	MV	MAD	MV	MAD	MV	MAD	MV	MAD	MV	MAD	MV	MAD
MSE Backprop	0.075	0.092	0.139	0.139	0.478	0.102	0.0078	0.0120	0.324	0.324	0.426	0.113
MSE GA	**0.071**	**0.089**	0.132	0.132	0.462	0.104	0.0078	0.0096	0.32	0.32	0.418	0.116
MLPGAgMI	0.089	0.093	**0.109**	**0.109**	0.448	0.098	0.0075	**0.0078**	0.312	0.312	0.405	0.103
MLPfXGAgMI	0.102	0.096	0.191	0.191	**0.437**	**0.094**	**0.0072**	0.0081	**0.307**	**0.307**	**0.382**	**0.092**

It is observed that MLPfXGAgMI presents low errors for all datasets except for the iris dataset. The low dimensionality of this space can be the cause. For the rest of the datasets, MLPfXGAgMI performs equal or better that MLP trained under MSE, either with genetic algorithms or backpropagation. The number of states in which continuous variables are discretized is an important factor. Fig. 3 shows the average error as a function of the number of states ns in which each variable is discretized.

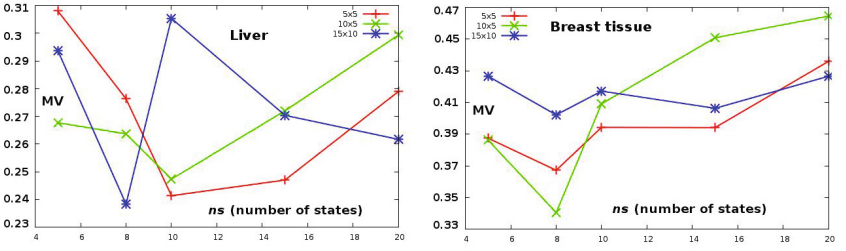


Fig. 3. Avg. MV as function of ns, the number of states in which each feature is discretized

There are two drawback for learning models guided by information-theoretic quantities. The first one is that the probability density function (pdf) for all classes may be unknown, and thus, mutual information has to be approximated by some heuristic [7]. Here, we supposed that the pdf can be computed directly from data, and as training vectors were randomly chosen, this does not affect the pdf.

A second drawback is that of non-equivalent ranges for the obtained results. Ranges do not need to be equivalent as an infinite number of ranges can offer maximal information. For the classification task, this is translated into the fact that there may not be a continuous transformation in the output. This is so because the range of the output in MLPfXGAgMI is not necessarily the same as the defined for the expected output. Mutual information does not lead the system as to be confined in a range in which average error is minimized, but to a range in which mutual information is maximized, although there can be an infinite number of those ranges. Thus, a mapping from the range for the output for MLPfXGAgMI into the range of expected output is needed. Here, we just checked all possible permutations for N classes and recorded the permutation with the lowest error.

4 Conclusions and Discussion

We tested the capabilities of multilayer perceptrons as classifiers, when trained with genetic algorithms and guided by mutual information between the obtained results and the class label. Also, attributes were substituted by implicit functions of them in order to maximize the information they present aout the class. We proposed a greedy algorithm that operates in two stages. In the first stage, each variable is substituted by a function of it such that mutual information between that function and the expected output is maximized. In the second stage, once the individual variables are substituted by functions that maximizes mutual information, a multilayer perceptron is fed with those functions as input and then it approximates a function such that the mutual information between the obtained output and the expected output is maximized.

Results are encouraging as shown in different datasets. We are still exploring several paths in this scheme. For example, it can happen that a function of one of the variables that maximizes mutual information shades other variables. We are working in the case in which the search of functions of individual variables that maximizes mutual information takes into account the already substituted variables, and thus, a new compound system is to be maximized.

Acknowledgments. A. Neme and C. Islas are in the SNI - CONACYT México. We thank colleagues at the Complex Systems Group at the Universidad Autónoma de la Ciudad de México for fruitful discussions.

References

1. Rojas, R.: Neural networks, a systematic introduction. Springer (1996)
2. Principe, J.C. (ed.): Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives. Springer (2010)
3. Xu, D., Principe, J.C.: Training MLPs layer-by-layer with the information potential. In: Intl. Joint Conf. on Neural Networks, Washington, vol. 8, pp. 1716–1720 (1999)
4. Shannon, C.E.: A Mathematical Theory of Communication. Bell System Technical Journal 27, 379–423, 623–656 (July & October 1948)
5. Cellucci, C.J., Albano, A.M., College, B., Rapp, P.E.: Statistical Validation of Mutual Information Calculations: Comparison of Alternative Numerical Algorithms. Physical Review E 71(6) (2005), doi:10.1103/PhysRevE.71.066208
6. Santos, J., Marques de Sá, J., Alexandre, L., Sereno, F.: Optimization of the error entropy minimization algorithm for neural network classification. In: ANNIE. Intelligent Engineering Systems Through Art. Neural Net., vol. 14, pp. 81–86. ASME Press, USA (2004)
7. Silva, L., Marques de Sá, J., Alexandre, L.: Neural Network Classification using Shannon's Entropy. In: Proceedings of the 13th European Symposium on Artificial Neural Networks, ESANN 2005, Bruges, Belgium, April 27-29 (2005)
8. Hild, K.E., Erdogmus, D., Torkkola, K., Principe, J.: Feature extraction using information-theoretic learning. IEEE Tran. on Pattern Analysis and Machine Intelligence 28(9), 1385–1392 (2006)
9. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998), http://www.ics.uci.edu/mlearn/MLRepository.html

Hybrid Evolutionary Algorithm with a Composite Fitness Function for Protein Structure Prediction

Camelia Chira[1] and Nima Hatami[2]

[1] Department of Computer Science
Babes-Bolyai University
Kogalniceanu 1, Cluj-Napoca 400084, Romania
cchira@cs.ubbcluj.ro
[2] BioCircuits Institute
University of California, San Diego
La Jolla, CA 92093-0328, USA
nhatami@ucsd.edu

Abstract. The problem of predicting a protein structure with minimum energy from a sequence of amino acids has a significant importance in biology. This is a computationally challenging problem being NP-hard even in simplified lattice protein models such as the hydrophobic-polar (HP) model. We investigate the performance of hybrid evolutionary algorithms in the context of the bidimensional HP model. A new fitness function to evaluate the quality of a protein conformation is proposed and engaged in a hybrid evolutionary model to address protein structure prediction. The evolutionary model relies on hill-climbing strategies integrated in the search operators and a meaningful diversification of genetic material. The proposed fitness takes into account the energy of the conformation as well as the existence of a certain conformation H core on the HP lattice. The resulting weighted fitness function is able to guide the evolutionary search in a more efficient way as emphasized by computational experiments.

Keywords: Protein Structure Prediction, Evolutionary Algorithms, Hill-Climbing, Diversification.

1 Introduction

Protein structure prediction (PSP) is one of the most important and challenging problems in molecular and computational biology. Given the primary sequence of a protein, the PSP task aims to find the protein's native configuration with minimum energy. The PSP problem has been shown to be NP-hard [5] even in simplified protein models such as the hydrophobic-polar (HP) model [6]. Many and diverse computing approaches to PSP have been investigated including evolutionary search [1,3,4,18], ant colony optimization [17], memetic algorithms [12] and tabu search [13].

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 184–191, 2012.

The HP model [6] divides amino acids in two classes emphasizing hydrophobicity as the most important difference between amino acids. The two types of residues considered are *H (hydrophobic or non-polar)* and *P (hydrophilic or polar)*. A protein is a sequence of amino acids located in a lattice and forming a self-avoiding chain. Two residues are considered topological neighbors if they are adjacent (either horizontally or vertically) in the lattice. The energy associated to a protein conformation takes into account every pair of H residues which are topological neighbors and are not connected by a bound (i.e. are not consecutive amino acids in the chain). Every such H-H contact contributes -1 to the energy function. The aim of PSP is to find the protein conformation with minimum energy.

Evolutionary approaches to PSP normally engage the energy of a conformation as the fitness function of the algorithm leading to a search process that can easily trap in local optima. In the current paper, we investigate an extension of a hybrid evolutionary model based on a new composite fitness function. The basic framework of the hybrid algorithm combines pull-move transformations and uniform crossover with hill-climbing strategies and has been originally proposed in [2] and later improved with a fingerprint diversification stage in [3]. The diversification mechanism is engaged to facilitate an efficient solution space exploration by identifying individuals corresponding to protein conformations which are similar from a fingerprint perspective and replacing redundant genetic material with newly generated individuals. The main contribution in the current hybrid model extension is the definition of a new fitness function to replace the common conformation energy previously engaged to evaluate the quality of an individual. Good conformations should have as low energy as possible (i.e. high number of H-H topological contacts). This is in fact the objective of PSP but it is very often not sufficient for a an efficient guidance in the search process due to the particular PSP characteristics such as the existence of multiple local optima and the need to avoid the search get stuck in one of these locally best solutions. It has been observed that low-energy conformations have a H core approximately surrounded by the P residues. The existence of such a core is introduced in the objective of the evolutionary search in the currently proposed model. This is achieved by using a weighted fitness function taking into account both the energy of the conformation and the placement of H residues in the HP grid.

Numerical experiments for the bidimensional HP model indicate a good performance of the hybrid evolutionary model based on the introduced fitness function. Experiments are carried out for commonly used HP protein benchmarks and results are competitive with those obtained by related population-based optimization algorithms.

The rest of the paper is structured as follows: section two describes the hybrid evolutionary algorithm used focusing on the representation, genetic operators, hill-climbing search strategy and fingerprint diversification; section three briefly

presents some related models designed for PSP particularly functions proposed to measure the quality of evolved protein conformations; section four presents the proposed weighted fitness function and section five discusses the numerical experiments and results.

2 Evolutionary Algorithm with Hill-Climbing Search Operators and Fingerprint Diversification

The hybrid evolutionary algorithm presented in [3] is used in the current paper as the framework of engaging the proposed composite fitness function. An individual represents a protein configuration for the given amino acid sequence. An internal coordinates representation is used to encode every individual (see [3] for more details). The population is initialized with randomly generated *valid* protein configurations. The direction at each position in the individual is randomly selected from the available locations so as to keep the conformation free of collisions. Hill-climbing influences the way in which crossover and mutation are applied. Recombination uses the parameter k representing the number of offspring generated from the same parents. Only the best fitted offspring (out of the k individuals created) is selected and used in replacement strategies. Uniform crossover is adapted to allow the generation of valid configurations: whenever a value from one parent can not be accepted due to collisions, a valid direction randomly selected is chosen. The mutation operator is based on the pull move transformation [13] by which a single residue is moved diagonally causing the transition of connecting residues. In the case of mutation, the integrated hill-climbing mechanism means the application of pull moves for a selected individual at every possible position resulting in the generation of several individuals from which the one having the best fitness is selected.

The offspring generated as a result of recombination or mutation replace the first parent if they clearly represent better-fitness configurations. Otherwise, the newly generated genetic material is used to replace the most redundant individuals in the population. These redundant individuals are determined based on the fingerprint distance. The fingerprint is a vector denoted by f where each component i monitors the contact status between a well-defined pair of residues. If the two residues monitored at position i are topological neighbours then $f(i) = 1$, otherwise $f(i) = 0$. Given two individuals I and J of the population, each fulfilling a different set of contacts, their degree of dissimilarity will be expressed by the Hamming distance $H(f_I, f_J)$ reported to the fingerprint length L_f. The degree of originality $\omega(I)$ of an individual I is defined as the smallest fingerprint Hamming distance between I and any other fitter individual J, reported to L_f (see [3]). A randomized approach to applying diversification is engaged by which an individual I is replaced depending on the value $\omega(I)/min_f$ (where min_f is a parameter representing the minimum allowed fingerprint distance).

3 Related Work

Genetic algorithms have been initially applied in [18] to address the PSP problem. Individuals are encoded using internal coordinates with absolute moves and a population of valid conformations is evolved by mutation and crossover. Evolutionary approaches to PSP have been extensively investigated since then focusing on the performance of the 'simplest' genetic algorithm [10], hybridization of genetic algorithms with a backtracking-based repairing procedure [4], multimeme algorithms combining genetic algorithms with a set of local search heuristics [12,9], estimation of distribution algorithms (evolutionary algorithms based on probabilistic models which replace the standard crossover and mutation) [16], combination of genetic algorithms, Taguchi method and particle swarm optimization [15], genetic tabu search [19], hybrid evolutionary algorithms based on hill-climbing search and fingerprint diversification [3] and multiobjective evolutionary optimization [8].

An essential requirement for an effective evolutionary search process refers to a good fitness function used in the evaluation of candidate protein conformations. Most existing evolutionary algorithms use the energy as defined in the HP model for this purpose. However, the energy function is not an efficient search guide as its ability to discriminate between protein conformations having the same number of H-H contacts but different topology is clearly limited. Some studies propose alternative fitness functions to address this limitation [11,14,7].

In [11], a fitness function is used to rank conformations according to the conventional energy function allowing at the same time a distinction between conformations with the same number of H-H topological contacts based on a distance-dependent function. The energy function proposed in [14] incorporates measurements of the compactness of conformations. The conventional energy function is incorporated and completed with a penalty factor to account for the constraint violation regarding the self-avoiding chain. Furthermore, the compactness of the conformation and the distance of P residues from the hydrophobic core are considered in the fitness. In [1], a global energy function is proposed as fitness in which pairs of nonconsecutive H residues contribute to the fitness (even if they are not topological neighbors) based on the distance between them. The memetic algorithm presented in [9] uses a fitness function based on H-compliance (the proximity of H residues to the center of the hypothetical H core) and P-compliance (the placement of P residues relative to the core of the conformation) measures.

A comparative study focusing on the performance of these fitness functions in evolutionary search is presented in [7]. The analysis reviews the degree of discrimination and the search performance as revealed by best found energy values, frequency of best solutions, standard deviation, overall average and best proximity and overall success counter. The findings of this study confirm the poor discrimination capabilities of the standard energy function as all alternative functions are able to provide a finer-grained discrimination. The best overall peformance is obtained by the distance-dependent function [11] and the H and P compliance modified energy [9].

4 Proposed Composite Fitness Function

A weighted fitness is proposed to evaluate the quality of a protein conformation in the HP model. The fitness function is extremely important to efficiently guide a successful evolutionary search process. The energy of a conformation is used as the first weighted term in the proposed fitness as the aim of PSP is to find minimum-energy conformations. The second term takes into account the placement of H residues in the lattice as detailed below.

Let the sequence $S = s_1...s_n$ be a protein structure with n amino acids. Each residue $s_i, \forall i$ can be either H or P and can occupy one location in the lattice. A valid protein configuration forms a self-avoiding path on a regular lattice. The energy associated to a protein conformation c takes into account every pair of H residues which are topological neighbors and are not consecutive in the chain (see Eq. 1).

$$E(c) = \sum_{\substack{s_i, s_j \in S \\ s_i = s_j = H}} e(s_i, s_j), \tag{1}$$

where

$$e(s_i, s_j) = \begin{cases} -1, \text{ if } s_i \text{ and } s_j \text{ are topological contacts} \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

The objective of PSP is to find the protein conformation with minimum energy $E^* = min\{E(c)|c \in C(S)\}$, where $C(S)$ contains all valid conformations for amino acid sequence S. The minimization of this energy function is the most straightforward approach to detect good protein conformations. However, two conformations having the same number of H-H contacts but different topological structure would be evaluated to the same fitness although a more compact structure should obviously have a better quality.

The idea behind the second weighted term in the proposed fitness is to measure how distant are the H residues from the core of the conformation and use the minimization of this distance as an additional objective of the evolutionary search. In the HP model, low-energy conformations usually tend to form a H core approximately surrounded by the P residues. Calculating the distance of each residue from the average coordinates of all H residues provides a measure to estimate how far are the H residues locations from the conformation core.

The center of the conformation denoted by s_m is calculated based on the average value of the H residues coordinates in the grid. The second term $L(c)$ takes into account the locations of all H residues relative to the center s_m as shown in Eq. 3.

$$L(c) = \frac{1}{count_H(S)} \sum_{\substack{s_i \in S \\ s_i = H}} d(s_i, s_m) \tag{3}$$

where $count_H(S)$ represents the number of H residues in S and $d(s_i, s_m)$ represents the Euclidean distance between the lattice position of residue s_i and the calculated conformation center s_m.

Table 1. Results obtained for standard bidimensional HP instances

Inst.	Length	Sequence	E^*	ELF [3]	ELF with New Fitness
S1	20	$HPHP^2H^2PHP^2HPH^2P^2HPH$	-9	-9	-9
S2	24	$H^2P^2HP^2HP^2H^2P^2HP^2HP^2HP^2H^2$	-9	-9	-9
S3	25	$P^2HP^2H^2P^4H^2P^4H^2P^4H^2$	-8	-8	-8
S4	36	$P^3H^2P^2H^2P^5H^2P^2H^2P^4H^2P^2HP^2$	-14	-14	-14
S5	48	$P^2HP^2H^2P^2H^2P^5H^{10}P^6H^2P^2H^2P^2HP^2H^5$	-23	-23	-23
S6	50	$H^2PHPHPHPH^4PHP^3HP^3HP^4HP^3HP^3$ $HPH^4PHPHPHPHH$	-21	-21	-21
S7	60	$P^2H^3PH^8P^3H^{10}PHP^3H^{12}P^4H^6PH^2PHP$	-36	-35	-35
S8	64	$H^{12}PHPHP^2H^2P^2H^2P^2HP^2H^2P^2$ $H^2P^2HP^2H^2P^2H^2P^2HPHPH^{12}$	-42	-40	-42

The fitness function of a conformation c considers a weighted contribution of the two terms $E(c)$ and $L(c)$ as follows: $\mathcal{F}(c) = w_E * E(c) + w_L * L(c)$, where w_E and w_L represent the weights given to each term (they are expressed as real numbers chosen such that $w_E + w_L = 1$). A higher weight should be normally given to the conformation energy as optimizing this term remains the major problem objective.

5 Computational Experiments and Results for HP Model

The protein benchmarks commonly used in the bidimensional HP model are used for computational experiments. The hybrid evolutionary algorithm is based on the following parameter setting: the population size and the number of generations are both set to 100; the number of offspring generated as part of hill-climbing crossover is 10; fingerprint diversification occurs every 10 generations; the minimum fingerprint distance allowed min_f is 0.01 and the weight parameters in the fitness function are $w_E = 0.75$ and $w_L = 0.25$. Table 1 reports the best results obtained over 10 runs for each protein sequence considered.

The energy values obtained by the evolutionary algorithm using the proposed fitness function are given in the last column of Table 1. The column labeled E^* shows the known optimum value for each bidiminesional HP benchmark considered (the length and sequence for each instance are also given). The obtained results are competitive with those obtained by the same hybrid evolutionary algorithm (column labeled ELF [3]) using the energy as fitness. Figure 1 presents as an example two conformations evolved for S1 and S4 both having the best known energy.

Considering that the performance of the starting point hybrid evolutionary algorithm ELF is really competitive with regard to related population-based search algorithms (see [3] for a detailed analysis and comparissons), it is clear that a composite fitness function taking into account the topological structure of the conformation in addition to its energy is further supporting a good exploration and exploitation of the solution space offering an efficient guidance to

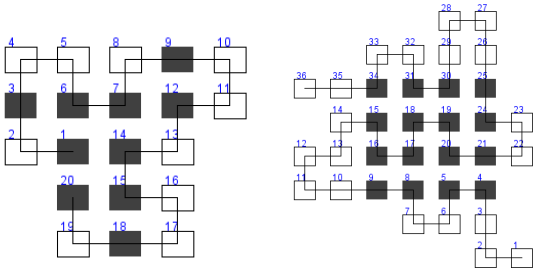


Fig. 1. Conformations detected for sequences S1 and S4 from Table 1. H residues are black and the P residues are white squares.

the search process. It is expected that the potential of the introduced weighted function to become more clear for large protein sequences from a higher number of algorithm runs.

6 Conclusions and Future Work

A weighted fitness function was presented for the evaluation of protein conformations in the bidimensional HP model. Both the energy of the conformation and its topological structure contribute to the fitness of conformations evolved by a hybrid genetic algorithm based on hill-climbing operators and fingerprint diversification.

Computational experiments indicate a competitive performance of evolutionary search guided by the proposed fitness function. The behaviour of this evaluation function needs to be further studied particularly for protein benchmarks with long amino acid sequences where the benefits of measuring the distance of the H amino acids from the center of the conformation are expected to become even more clear. Moreover, future work focuses on extending the numerical experiments to consider higher-size populations and an increased number of generations to better observe the evolution of conformations.

Acknowledgments. This research is supported by Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCS Romania.

References

1. Berenboym, I., Avigal, M.: Genetic algorithms with local search optimization for protein structure prediction problem. In: GECCO 2008, pp. 1097–1098 (2008)
2. Chira, C., Horvath, D., Dumitrescu, D.: Hill-Climbing search and diversification within an evolutionary approach to protein structure prediction. BioData Mining 4(1), 23 (2011)

3. Chira, C.: A hybrid evolutionary approach to protein structure prediction with lattice models. In: IEEE Congress on Evolutionary Computation 2011, pp. 2300–2306 (2011)
4. Cotta, C.: Protein Structure Prediction Using Evolutionary Algorithms Hybridized with Backtracking. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2687, pp. 321–328. Springer, Heidelberg (2003)
5. Crescenzi, P., Goldman, D., Papadimitriou, C.H., Piccolboni, A., Yannakakis, M.: On the Complexity of Protein Folding. Journal of Computational Biology 50, 423–466 (1998)
6. Dill, K.A.: Theory for the folding and stability of globular proteins. Biochemistry 24(6), 1501–1509 (1985)
7. Garza-Fabre, M., Rodriguez-Tello, E., Toscano-Pulido, G.: Comparing alternative energy functions for the HP model of protein structure prediction. In: IEEE Congress on Evolutionary Computation 2011, pp. 2307–2314 (2011)
8. Garza-Fabre, M., Rodriguez-Tello, E., Toscano-Pulido, G.: Multiobjectivizing the HP Model for Protein Structure Prediction. In: Hao, J.-K., Middendorf, M. (eds.) EvoCOP 2012. LNCS, vol. 7245, pp. 182–193. Springer, Heidelberg (2012)
9. Islam, K., Chetty, M.: Clustered memetic algorithm for protein structure prediction. In: IEEE Congress on Evolutionary Computation, pp. 1–8 (2010)
10. Khimasia, M.M., Coveney, P.V.: Protein structure prediction as a hard optimization problem: the genetic algorithm approach. Molecular Simulation 19, 205–226 (1997)
11. Krasnogor, N., Hart, W.E., Smith, J., Pelta, D.: Protein structure prediction with evolutionary algorithms. In: Genetic and Evolutionary Computation Conference (GECCO 1999), pp. 1569–1601. Morgan Kaufmann (1999)
12. Krasnogor, N., Blackburne, B.P., Burke, E.K., Hirst, J.D.: Multimeme Algorithms for Protein Structure Prediction. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN VII. LNCS, vol. 2439, pp. 769–778. Springer, Heidelberg (2002)
13. Lesh, N., Mitzenmacher, M., Whitesides, S.: A complete and effective move set for simplified protein folding. In: RECOMB 2003: Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology, pp. 188–195. ACM (2003)
14. Lopes, H.S., Scapin, M.P.: An Enhanced Genetic Algorithm for Protein Structure Prediction Using the 2D Hydrophobic-Polar Model. Artificial Evolution, 238–246 (2005)
15. Lin, C.-J., Hsieh, M.-H.: An efficient hybrid Taguchi-genetic algorithm for protein folding simulation. Expert Systems with Applications 36(10), 12446–12453 (2009)
16. Santana, R., Larrañaga, P., Lozano, J.A.: Protein folding in simplified models with estimation of distribution algorithms. IEEE Transactions on Evolutionary Computation 12(4), 418–438 (2008)
17. Shmygelska, A., Aguirre-Hernández, R., Hoos, H.H.: An Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) Ant Algorithms 2002. LNCS, vol. 2463, pp. 40–53. Springer, Heidelberg (2002)
18. Unger, R., Moult, J.: Genetic algorithms for protein folding simulations. J. Molec. Biol. 231, 75–81 (1993)
19. Zhang, X., Wang, T., Luo, H., Yang, J., Deng, Y., Tang, J., Yang, M.: 3D Protein structure prediction with genetic tabu search algorithm. BMC Systems Biology 4(suppl. 1), S6 (2010)

3D Fuzzy GIST to Analyze Emotional Features in Movies

Mingu Kwon and Minho Lee

School of Electronics Engineering and Computer Science, Kyungpook National University,
1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701 Korea
mgkwon@ee.knu.ac.kr, mholee@knu.ac.kr

Abstract. In this paper, we propose a 3D fuzzy GIST to effectively describe the visual dynamic features related to the emotional characteristics in a movie clip. Unlike the previous fuzzy approaches, which use images, the proposed method employs movie clips and can dynamically extract the features to classify the emotional characteristics in a movie clip. The 3D fuzzy GIST based on 3D tensor data including L*C*H color (L: Lightness, C: Chroma, H: Hue) and orientation information in a movie clip can extract the visual dynamic features related to the emotional characteristics in a movie clip. The extracted visual dynamic features obtained by the proposed 3D fuzzy GIST are used as inputs to an adaptive neuro-fuzzy inference classifier. The classifier is provided with the mean opinion scores as the teaching signals. Experimental results show that the system with the proposed 3D fuzzy GIST feature extractor not only discriminates the positive emotional features from the negative ones but also identifies the changes of emotional features in movie clips successfully.

Keywords: 3D fuzzy GIST, positive and negative emotion, adaptive neuro-fuzzy inference classifier.

1 Introduction

With the development of robotic technology and the growing interest about human computer interaction, intelligent machines are being developed to satisfy the users' requirements and provide services to improve the quality of the users' life. However, there is a huge scope for improvement as we need to develop a real human-like intelligent system that can understand user's conditions, especially emotions and provide suitable services accordingly. Emotion is a special dynamic form of cognition that is highly complicated and requires new approaches for analysis.

Recently, more and more researchers have been paying attention to the visual influences on the human emotion as in [1-4], because visual information plays a very important role in affecting a subject's emotional status [5]. Most of previous engineering researches regarding with emotion was done with the presentation of faces with different facial expressions [6]. Some evidence suggests that natural scene and movies are more reliable and strong enough to activate the human emotional responses [7]. Therefore, a mechanism to understand human emotion during the visualization of natural scenes needs to be developed so that a machine can understand the human

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 192–199, 2012.

emotion in various environments. In recent, Zhang and Lee proposed the emotion understanding models using incremental adaptive neuro-fuzzy inference systems (ANFIS) [8-9]. They proposed the fuzzy GIST concept to analyze the emotional factors in natural static scenes [8], and successfully descried the visual features as well as valence and arousal EEG features using the fuzzy GIST [8-9].

Although the fuzzy GIST can successfully extract the emotional features while viewing stationary natural images, in reality, human subjects are frequently exposed to dynamic stimuli such as movies rather than stationary images. The video stimuli comprise of successive image frames which contain various information depending on the time sequences. Therefore, there exists a need to develop a feature extraction method, which can extract the dynamic features. In this paper, we present a feature extraction method referred to as "3D fuzzy GIST", which analyzes the emotional space and can extract emotional features with dynamic characteristics. The proposed 3D fuzzy GIST uses low level visual information such as color and orientation information to extract the primitive emotional factors of each frame in movie clips. The proposed method uses the tensor data as basic feature that is useful to describe the dynamic characteristics of movies. Based on the human subject feedback emotions evoked by movie clips, the neuro-fuzzy inference system is adopted to learn the 3D fuzzy GIST and classify the two valence states as positive and negative emotions.

The remainder of this paper is organized as follows. The emotion analysis system using the proposed feature extraction is presented in section 2. In section 3, presents the experimental results and evaluate the performance of the system using the proposed feature extraction method. Conclusions and discussions are presented in section 4.

2 Methods

2.1 Overview of the Emotion Analysis System

Fig. 1 illustrates the graphic outline of the emotional analysis system using the proposed 3D fuzzy GIST to extract the emotional features in a movie clip. The system uses the visual information (H:hue, S:saturation, I:intensity and O:orientation) to get the 3D fuzzy GIST of a movie clip. The 3D tensor data are used to describe the dynamic characteristics of movies. The tensor data, which are divided into color (H, S and I) and orientation information from a movie clip, are clustered to make emotional descriptors. Then, all the descriptors are cascaded to obtain the 3D fuzzy GIST. Preprocessing for visual dynamic feature analysis in a movie clip is explained in section 2.2. The 3D fuzzy GIST, which is a feature extraction method to describe the emotional characteristics in a movie clip, is explained in section 2.3. The extracted features obtained by the 3D fuzzy GIST are used as inputs of a classifier that can classify the parts of a movie clip as positive and negative emotions. In this paper, we use the adaptive neuro-fuzzy inference network (ANFIS) as a classifier to recognize the emotional characteristics in movie clips.

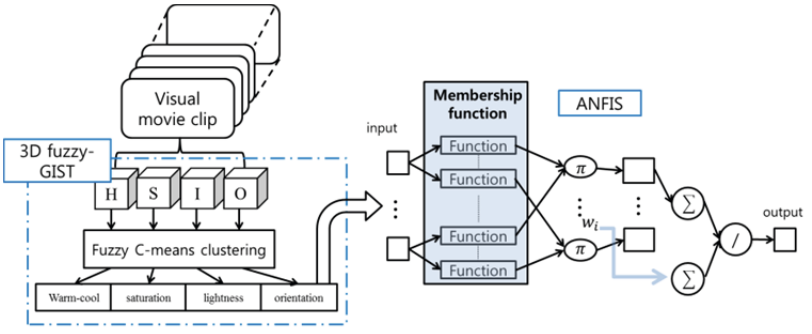


Fig. 1. Graphic outline of the emotion analysis system using the proposed 3D fuzzy GIST

2.2 Preprocessing for Visual Dynamic Feature Analysis

The procedure of feature extraction using the 3D fuzzy GIST from movie clips is depicted in Fig. 2. Human emotion can be induced by some stimulating factors of successive images [10]. According [11], there exists a relationship between the line directions and the dynamic sensation of the images. The horizontal lines in the image represent calm, stability and openness, while the vertical line represents directness and definitude. The circle and soft-edge figure induce the grace and rhythm [10]. Besides it is well known that color information contained in an image affects the human emotion status dramatically [12]. According to the relationship between scene and emotion, orientation and color information are used as basic features to construct the 3D fuzzy GIST from the movies. The operator used to extract orientation information is Gabor function [13].

The orientation feature vector is the output magnitude of a group of multi-scale oriented filters. Down sampling is used to reduce the dimensionality of each multi scale filter's output size $N \times N$. As a result, orientation of an image frame is represented by a vector with $N \times N \times K$ dimension, where K is the number of multi-scale orientation filters. L*C*H color space (Lightness, Chroma and Hue) [14], selected for color information, is an important element that affects the emotional states of human. Luminance denotes the color lightness; Chroma denotes the saturation of the color; Hue denotes the different colors. Based on the theory of color psychology [15], the high light colors invoke brilliant, light and lively emotions. The warm color always evokes happy, lively and exhilarated feelings, while the cool color always leads to sad, heavy and depressive feeling [12]. At the same time, the high saturation colors cause the showy, magnificent emotions and the low saturation colors make a modest and elegant feeling [16]. Therefore, we need to use both orientation and color features which are acquired from L*C*H color space. To reduce the computational time, color features as well as orientation feature are also down-sampled to $M \times M$, where M is greater than N.

2.3 3D Fuzzy GIST as Emotional Visual Feature Extractor

Based on the previously developed Fuzzy-GIST [8], [17], we propose the 3D fuzzy GIST to analyze visual dynamic emotional features in movie clips. To effectively analyze the movies that have a dynamic characteristic, we need to bundle up the successive image frames and make up cubic type three dimensional tensor, where the size of feature's tensor of orientation information is $N \times N \times T$ in each multi-scale orientation data (In case of color information, the size of tensor data is $M \times M \times T$). In this case, T is the length of time sequences. So, the dimension of orientation vectors is $N \times N \times T \times K$. With the help of this tensor data, we can analyze the visual dynamic characteristics related with emotional characteristics in a movie clip.

The 3D fuzzy GIST is originated from the GIST [18-19], and it is a kind of conceptual gist of a natural scene that contains semantic information. So we build the global low-level feature vector to represent a natural scene using color (H, S and I) and orientation information. Based on the fuzzy GIST [8], [17], we also use the fuzzy C-means clustering (FCM) [20]. In the proposed system the color and orientation information are clustered in three (high/middle/low) and four clusters, respectively using the FCM. By making use of the $M \times M \times T$ tensor data of each components (H, S and I) in L*C*H color information, each pixel in the tensor data is assigned a membership grade using each of the three membership functions in the FCM. Depending on the membership grade, the pixel is assigned to one of the three clusters. The high/middle/low clusters of the color information indicate warm, neutral and cool emotions in the movies.

The clustering of the orientation information in to four clusters is done in two stages, since the down sampled images have multi-scale orientations. In the first stage, the pixels in the tensor data corresponding to a single scale are given membership grades using the four membership functions. Depending on the membership grades the pixels are clustered. The procedure is repeated for each of the K scales. Then, the size of orientation vector becomes $4 \times K$. In the second stage, the clusters obtained for each of the scales are further clustered using a similar procedure to obtain four clusters. Therefore, finally we can get the 4×1 size vectors. Finally, we construct the visual dynamic features by concatenating each feature, and resultantly we can obtain the 3D fuzzy GIST as a feature extractor to describe the visual dynamic characteristics related to the emotional characteristics in a movie clip.

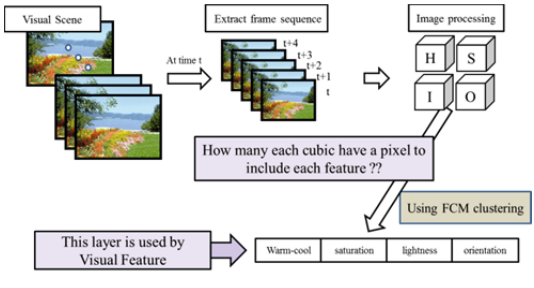


Fig. 2. Configuration of 3D fuzzy GIST; H (Hue image), S (Saturation image), I (Intensity image), O (Orientation image)

2.4 Adaptive Neuro-Fuzzy Inference System (ANFIS)

Human emotion recognition is a task that is frequently permeated with uncertainty. However, previous research on modeling a subject's emotion has already been able to reduce this uncertainty using probability theory to model the human emotional state by monitoring the presence or absence of a specific emotion [21]. Nonetheless, system modeling based on conventional mathematical tools is not well suited for dealing with ill-defined and uncertain systems. By contrast, a fuzzy inference system employing fuzzy "if-then" rules can model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analyses [22] such as measurement of degree of emotions in human beings. The neuro-fuzzy system used in this paper is the ANFIS. For a first order Sugeno fuzzy model [22], the rule set is as follows:

$$\text{IF } x_1 \text{ is } A_1^k \text{ AND } \cdots x_n \text{ is } A_n^k \text{ THEN } y_k = f^k(x_1, x_2, \cdots, x_n) \tag{1}$$

where, A_j^k represents the *j*-th variable defined by the membership function μ_j^k. The consequent part of the rule is the first-order polynomial of the input vector $[x_1, x_2, \cdots, x_n]$ and the mode predicts the values of the variable $[y_1, y_2, \cdots, y_n]$ given the input. Different types of membership functions can be used for the antecedent fuzzy set. In this paper, Gaussian membership functions are used to their ideal properties [23] for representing fuzzy membership function such as smooth, unimodal, and normal.

The clustering of the emotional features using FCM forms the primitive knowledge about emotion. Based on this knowledge, the neuro-fuzzy system develops its understanding of emotion under the control of an ANFIS network. By interacting with humans, the learning procedure searches for the optimal membership functions and optimal parameters of the consequent models [22]. Mean opinion scores are used for teaching signals of the neuro-fuzzy system.

3 Experimental Results

3.1 Environment of Experiments

The experimental visual stimuli employed in the present study consist of videos selected from movies and documentaries. To classify the emotion as positive and negative, we use a total of 39 videos as the training data, which are divided into two groups of 17 positive and 22 negative emotional. The test data consists of two groups. One group consists of 4 videos in which the emotional characteristics change from positive to negative, and the second group consists of 4 videos in which the emotional characteristics change from negative to positive. Most of the positive videos consist of documentary videos which have some beautiful natural scenes. On the other hand, most of the negative videos belong to some genre such as action movie, horror movie

and documentary which deals with a natural disaster. Some sample frames of each video are shown in Fig. 3.

To obtain the target data of the ANFIS network related with the emotional states, the human subjects are asked to answer the following questionnaire after each visual stimulus: 'what was your emotion state after seeing the movie ?', 'when did your emotion change from positive to negative or from negative to positive ?'.

Fig. 3. Examples of captures from videos that is used as visual stimuli

3.2 3D fuzzy-GIST and Visual Information Extraction

The orientation information was obtained by a group of multi-scale oriented filters. We obtain 4 orientation information such as 0°, 45°, 90° and 135° in three different successive image scales. Then they were down-sampled from a size of 256×256 to a size of 4×4 to form the feature vector with dimension of $4 \times 4 \times 12$. To analyze the emotional features in the video clips, we bundled up 7 successive image frames and construct the cubic type three dimensional tensors. So, the dimension of orientation vectors is $4 \times 4 \times 7 \times 12$. The FCM is used to partition the orientation information into four clusters in terms of orientation distribution. So, the size of orientation vectors becomes 4×12. Then, the 48×1 vectors are applied by the FCM that is used to partition the vectors into four clusters again. Finally, we can get the 4×1 size vectors. The L*C*H color space is used for color information, and they are down-sampled to a size of 10×10. The L*C*H color space features also bundle up 7 continuous frames and make up the cubic type three dimensional tensor data, which is similar to the procedure for the orientation features. So dimension of color features in each color components is $10 \times 10 \times 7$. Each component (Lightness, Chroma and Hue) is applied to FCM that is used to partition the information into three clusters. From this procedure, we can get the features which have a 13 dimensional feature vector.

3.3 Emotion Analysis

After the training, the emotional analysis system is tested using the test data which is randomly selected. To analyze the subject's individual emotion, we obtain the

features using the 3D fuzzy GIST, and then use the ANFIS which is the classifier to classify emotions such as positive and negative. The ANFIS is initialized by 5 membership functions to categorize emotional characteristics as positive and negative.

As shown in Table 1, the proposed system successfully categorizes the emotional characteristics as positive and negative in the movie clips. The average performance of the system on the test data is almost 72.88 %. These results show us that the proposed 3D fuzzy GIST is appropriate as a feature extraction method to analyze and find the emotion states in movies

Table 1. The result of the proposed emotion analysis system toward the task of recognizing the 2 emotional characteristics in the movie clilps

	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Subject 7	Subject 8	Subject 9
Train(%)	96.62	89.83	98.75	98.20	94.71	94.82	95.13	97.24	96.58
Test(%)	72.26	81.30	79.13	77.39	68.61	72.61	64.99	69.83	69.87

4 Conclusion

In order to develop an autonomous emotion development system, we proposed a suitable new emotional feature extraction method, so called the 3D fuzzy GIST, to analyze dynamic video stimuli.

According to the relationship between emotional factors and the characteristics of image, L*C*H color and orientation information are adopted to analyze the visual features at semantic level, by incorporating the fuzzy concept to extract features with semantic meanings. Furthermore, we suggest the 3-dimensional tensor as the basic feature type to analyze video stimuli. It means that proposed feature extraction concept is more suitable to apply in real space.

As a future work, we would like to implement an emotion understanding system that can autonomously analyze the emotional features using the 3D fuzzy GIST together with EEG emotional features of human subjects. Furthermore, we would like to combine the visual dynamic features and EEG features, and apply the developed "recurrent neuro-fuzzy system" to analyze the human emotion mechanism and to predict the emotion of human.

Acknowledgment. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by Ministry of Education, Science and Technology(2012R1A1B6004217).

References

1. Colombo, C., Bimbo, A., Pala, P.: Semantics in visual information retrieval. IEEE Multimedia 6(3), 38–53 (1999)
2. Iwadatae, Y., Inoue, M., Suzuki, R., Hikawa, N., Makino, M., Kanemoto, Y.: MIC Interatcive Dance System. In: Proceedings of the International Conference on Knowledge-based Intelligent Engineering Systems and Allied Technologies, vol. 1, pp. 95–98 (2000)

3. Assfalg, J., Bertini, M., Colombo, C., Bimbo, A.: Semantic annotation of sports videos. IEEE Multimedia 9(2), 52–60 (2002)
4. Yu, C., Xu, L.: An emotion-based approach to decision making and self learning in autonomous robot control. In: The Fifth World Congress on Intelligent Control and Automation, vol. 3, pp. 2386–2390 (2004)
5. Oatley, K., Jenkins, J.: Understanding Emotions. Blackwell Publishers, Cambridge (1996)
6. Günenkin, B., Basar, E.: Emotion face expressions are differentiated with brain oscillations. International Journal of Psychophysiology 64, 91–100 (2007)
7. Hasson, U., Malach, R., Heeger, D.J.: Reliablility of cortical activity during natural stimulation. Trends in Cognitive Sciences 14(1), 40–48 (2010)
8. Zhang, Q., Lee, M.: Emotion development system by interacting with human EEG and natural scene understanding. Cognitive System Research 14, 37–49 (2012)
9. Zhang, Q., Jeong, S., Lee, M.: Autonomous emotion development using incremental modified adaptive neuro-fuzzy inference system. Neurocomputing 86, 33–44 (2012)
10. Picard, R.: Affective computing. MIT Press, London (1997)
11. Weining, W., Yinglin, Y., Jianchao, Z.: A new SVM based emotional classification of image. Journal of Electronics 22(1), 98–104 (2005)
12. Wang, W.N., Yu, Y.L., Jiang, S.M.: Image retrieval by emotional semantics: A study of emotional space and feature extraction. In: IEEE International Conference on Proceedings Systems, Man and Cybernetics, vol. 4, pp. 3534–3539 (2006)
13. Feichtinger, H.G., Strohmer, T.: Gabor Analysis and Algorithms. Birkhäuser (1998)
14. CIE, Commission Internationale de TEclairageColorimetry. Publication No. 15. Bureau central de la, Vienna, Austria (1976)
15. Choi, J.-H., Sohn, J.-I., Yang, H.-S., Chung, Y.-R., Lee, M., Koh, S.-C.: A study of colour emotion and colour preference. part I: Colour emotions for single colours. Color Research and Application 29(3), 232–240 (2004)
16. Itten, J.: Art of Color. John Wiley & Sons, Inc., New York (1961)
17. Zhang, Q., Lee, M.: Fuzzy-GIST for 4-Emotion Recognition in Natural Scene Images. In: DEVLRN 2009: Proceedings of the 2009 IEEE 8th International Conference on Development and Learning, pp. 1–7. IEEE Computer Society, Washingon, DC (2009)
18. Zhang, Q., Lee, M.: Analysis of positive and negative emotions in natural scene using brain activity and gist. Neurocomputing 72(4-6), 1302–1306 (2009)
19. Oliva, A., Torralba, A.: Building the gist of a scene: the role of global image features in recognition. Progress in Brain Research 155, 23–36 (2006)
20. Bezdek, J.: Fuzzy mathematics in pattern classification. Ph.D. thesis, Cornell University Ithaca (1973)
21. Dubois, D., Prade, H.: Possibility theory is not fully compositional!: a comment on a short note by Greenberg, H.J. Fuzzy Sets Syst. 95(1), 131–134 (1998)
22. Jang, J.-S.R., Sun, C.-T., Mizutani, E.: Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. Prentice Hall, Inc., Upper Saddle River (1997)
23. Castellano, G.: A neurofuzzy methodology for predictive modeling. Ph.D. dissertation, Univ. of Bari (2000)

Interactive Information Retrieval Algorithm for Wikipedia Articles

Julian Szymański

Department of Computer Systems Architecture,
Gdańsk University of Technology, Poland
julian.szymanski@eti.pg.gda.pl

Abstract. The article presents an algorithm for retrieving textual information in documents collection. The algorithm employs a category system that organizes the repository and using interaction with the user improves search precision. The algorithm was implemented for simple English Wikipedia and the first evaluation results indicates the proposed method can help to retrieve information from large document repositories.

Keywords: information retrieval, documents clustering, text processing.

1 Introduction

The amount of data stored in documents in natural language requires advanced methods for information retrieval [1]. The size of textual repositories and unambiguity of natural language cause that the typical keyword-based search is insufficient [2]. Beside the fact that the keywords do not index all relevant information in the repository the user often even doesn't know precisely what he or she is searching for until see the retrieved results. The typical interaction with the search engine is a scenario in which the user gives poorly defined search criteria (given as search phrases) and expects the retrieval of relevant information or suggestions of content that will be interesting for her or him.

In this paper we present the interactive algorithm for documents retrieval based on categories that organize the documents repository. The algorithm has been applied for simple Wikipedia articles where it extends a keyword-based search.

2 Interactive Retrieval Algorithm

To improve organization of large repositories of documents a system of categories is often introduced. We can find this approach in traditional libraries, as well as in digital repositories (eg. MEDLINE) [3]. One of the biggest repositories of human knowledge is Wikipedia. It contains over 3 million articles and also provides categories that organize them on higher abstraction level than single articles. The Wikipedia articles have similar structure to web pages on the Internet, thus this repository is a good testbed for algorithms that may be used for web-scale information retrieval.

Our algorithm works in two phases. In the first one for a given search phrase the *recall* measure [4] is improved. In the second phase through the user interaction with the

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 200–207, 2012.

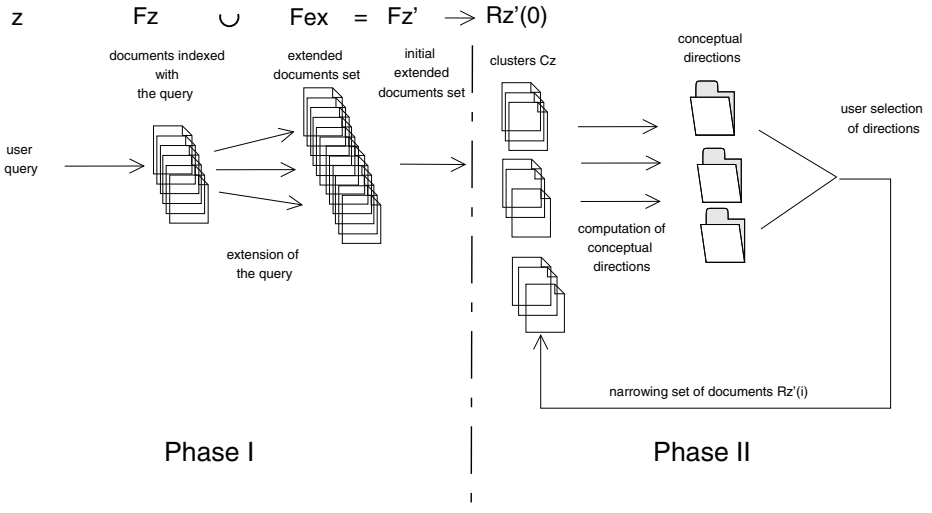


Fig. 1. A scheme of interactive algorithm for information retrieval

system the *precision* is improved. In figure 1 the scheme of the algorithm for interactive information retrieval is presented, where we can select the following steps:

Phase I

1. For a given user query z the set of articles is retrieved (F_z). These articles are the ones that contain the words given by the user.
2. Because of the unambiguity of natural language the words from query z will not index all documents relevant to the user. For example the synonyms *road* and *street* will index different sets of articles, but both of them are potentially interesting for the user who is searching for a word *way*. Thus the recall measure will not achieve its maximum because some documents that are relevant will not be retrieved. Improvement of the recall can be made by extending the F_z set with similar documents F_{ex}. There are many techniques that perform such a task:
   - Application of synonyms dictionary that contains groups of words of similar meanings, for such a task eg. WordNet [5] may be used. Also here the algorithms that compute semantic similarity between words may be useful [6],
   - *Latent Semantic Indexing* [7],
   - Query Expansion [8], that may be performed in many different ways eg. with diffusion of words on semantic network [9],
   - In our application we used modified approach of Contextual Network Graphs [10].

   The above techniques allow to extend F_z set (indexed only with search phrase z) and add documents that do not contain words given in the query, but they are somehow related with them (it depends on algorithm used for extending the F_z set). In Figure 1 we denote as F_{ex} set of the documents related to F_z. $F_{z'} = F_z \cup F_{ex}$ denotes the joined set of documents – indexed with z phrase and the related ones.

Thus in the set $F_{z'}$ recall measure will be greater than in F_z. The negative effect of such extension is a decrease of precision measure in $F_{z'}$. The $F_{z'}$ is the entrance point for the second phase of the algorithm which has been denoted as $R_{z'}(0)$. $R_{z'}$ is narrowed in succeeding steps of the algorithm by removing irrelevant groups of documents and strengthening significant ones. In step i this set has been denoted as $R_{z'}(i)$.

Phase II

3. The documents in Wikipedia are related with links. The strongest associations between articles form groups of documents (C_z) that are thematically similar to each other. Using this observation the $R_{z'}(i)$ set may be divided into groups that complete specified similarity (here established with links). In our implementation the thematic groups have been constructed using DBSCAN clustering that is known to provide well quality clusters that may have different shapes (not only convex as in typical clustering) [11]. The application of clustering allows us to treat very similar documents as one entity, which during calculation of their conceptual differences (described further) has a positive influence on the quality of the results (noise reduction) as well as warrants the effectiveness of computations (we do not have to operate on whole dataset of articles but we can work on their prototypes).

   Presenting the clusters to the user is a widely used method for improving effectiveness of information retrieval. Despite its advantages it causes some problems, especially if there are large numbers of clusters as well as there occur issues with their labeling.

   In our approach instead of presenting clusters C_z we present conceptual directions4 in which the user may continue the search. The conceptual directions are computed calculating Information Gain (IG) for each of the category k_i that represents clusters C_z of articles from $R_{z'}(i)$ set. The highest IG (calculated using formula 1) indicates the most important conceptual direction which describes how continuing the search in that direction will separate the articles thematically similar (because they belong to the same C_z cluster) from the others.

   The categories that represent the clusters are provided by the articles categorical associations. Each article in C_z belongs directly to some categories – we call it explicit association between the article and the category. What is more – the categories are related to one another which provides additional, implicit associations between the articles and the categories. If one article belongs explicitly to the category it gets association weight $w = 1$. If association to the category k_s results from relation of this category with other category k_d the w weight is calculated as e^{-dist}, where $dist$ is the number of categories between k_s and k_d. The aggregation of all article categories in C_z creates representation of that cluster. Computing IG (1) for each of the category k_i and selecting the highest ones allows to find the conceptual directions that separate articles within clusters.

$$IG(k_i) = -\sum_{j=1}^{n} p(w_j) \cdot \log\ p(w_j) \tag{1}$$

$$p(w_j) = \frac{|w_j|}{n} \tag{2}$$

where:

n - number of unique values of w weight in category k_i,
$|w_j|$ - number of occurrence of particular w_j weight values in category k_i.

4. Information gain is used for selecting the categories that are the most distinctive for articles in the set $R_{z'}(i)$. The categories with highest IG show directions where the user may continue the search. They are presented to the user and he or she may to point the categories that are interesting, or the irrelevant ones, and some of them may be left unjudged.

   Removing the documents that belong to categories that are negatively indicated and strengthening the documents that are pointed positively change the information gain distribution for categories within $R_{z'}(i+1)$ set. Selecting conceptual directions allows to narrow the relevant documents set and allow to guide search process in the direction described by the user clues. In next step algorithm calculates information gain in new set of articles $R_{z'}(i+1)$ and the user continues the search as it was described in step 3. Repeating this process allows to improve precision measure in next $R_{z'}$ set.

3 System Prototype

The algorithm for interactive information retrieval we implemented in the form of web page available at http://bettersearch.eti.pg.gda.pl.

User can here enter search phrase and refine the results selecting which clusters are the most related to his or her search. Pressing *redo search* button system narrows the result set according to user selection and forms new clusters that again organize, limited set of articles. If the organization of this set is not suitable for user requirements he can press *back* button and reselect his clusters. The area in right panel is reserved for displaying content of the article. The system allows user to enter the search phrase and organize the results in the form of the clusters that can be skimmed with buttons '$<$' and '$>$'. Additionally system presents the abstract concept that user can answer yes / no / don't know. This information is used to reorganize and narrow the results set.

4 Evaluation Results

Objective evaluation of the search results is challenging task, because every person can considers relevance of the results differently. The methods of retrieval evaluation can be divided into two main groups [12] based on below criterias:

1. Internal metrics.
   This evaluation is performed without external knowledge. Cohesion and distance of results is validated. Often this kind of metrics are similar to objective function of retrieval algorithm. One of the most well known is described with the formula 3.

$$\Phi(C) = \frac{1}{|C|} \sum_{c_i \in C} |c_i| \times sim(c_i, c_i), \tag{3}$$

where $sim()$ is some similarity function between retrieved elements (c_i). Internal metrics are best to compare results within the same data collection. However one have to keep in mind that whole environment of a test must not change (e.g. constant similarity measures, retrieval algorithm, etc.). This kind of metrics allows to evaluate results according to the defined similarity measure. As we are operate on natural language resources such a measure can also describe some aspect of the data so for the proper evaluation of the algorithm quality in terms of user satisfaction is not really useful. The quantitive measures of such a parameters as clusters density, dispersion do not provide evaluation of the algorithm usability that mus be measured qualititevely.

2. External metrics.
This metrics allow to evaluate usefulness of the retrieval algorithm in terms of satisfaction of user expectations. One of informal metrics is to collect feedback from users in the form of questionnaires. Another type of external metrics are formal metrics based on a relevance set. The most popular are *Precision* (P), *Recall* (R) combined into *F-measure* and *Purity*.

   Precision is a percentage of retrieved documents that are relevant [13] to the user query:

$$P = \frac{\text{Number of relevant documents}}{\text{Total amount of documents}} \tag{4}$$

   Recall is defined as percentage of relevant documents that were retrieved [13]:

$$R = \frac{\text{Number of relevant documents}}{\text{Total amount of relevant documents}} \tag{5}$$

   F-measure is a composition of Precision and Recall (weighted harmonic mean) and keeps a balance between them [13]:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \tag{6}$$

   where $\beta \in [0, \infty)$ is a weight coefficient. For $\beta = 1$ F-measure balances P and R. By increasing β we emphasis on Precision. Most common values for β are 1, 3 and 5.

   Purity is measured by dividing number of correctly retrieved documents4 by N – total amount of retrieved documents, what describes the formula 7.

$$\text{Purity}(Omega, C) = \frac{1}{n} \sum_k \max_j |\omega_k \cap c_j|, \tag{7}$$

For the seven test phrases: *mathematics*, *deer*, *microsoft*, *politics*, *tiger*, *siberian* we evaluate how the process of interaction with the user allows to narrow the number of elements in results set. In figure 2 we presented for each of test query histograms that shows how successive interactions with the user narrows the number of the results. Note in the first steps some small increases of the articles amounts are caused by extension of the query phase of the algorithm.

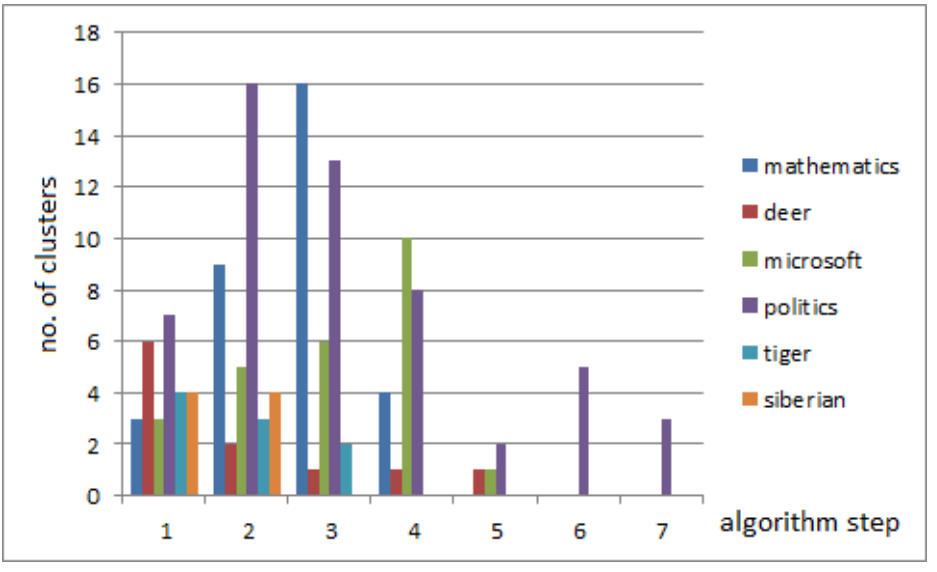


Fig. 2. Articles amount

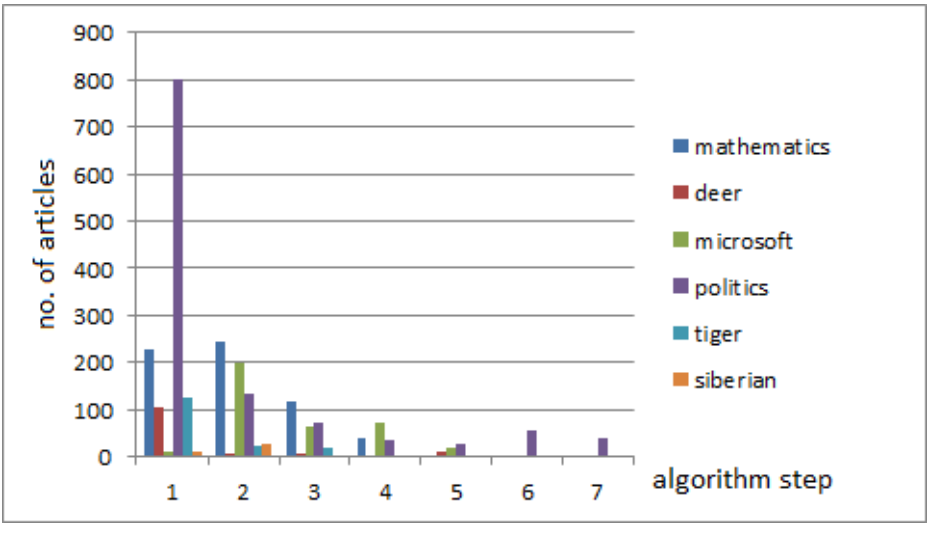


Fig. 3. Clusters amount

Also the number of the clusters in the function of algorithm steps (interactions with the user) shown in Figure 3 has decreasing trend what indicates the topic of groups are narrowed.

The qualitative evaluation has been performed using hand crafted relevance set for the above mentioned test queries for the last algorithm step, where interaction with the

Table 1. Results of evaluation WikiClusterSearch

Query	F_1	F_5	Purity
mathematics	0.72	0.85	0.87
deer	0.79	0.91	0.91
microsoft	0.63	0.74	0.78
politics	0.64	0.76	0.76
tiger	0.75	0.87	0.88
siberian	0.61	0.70	0.89

user the most refine search results. The values of F-measure (F_1 and F_5) and Purity are shown in the Table 1.

What can be seen from the results the Purity is kept on a high level for almost all queries. It indicates the results in final sets are highly related to the user expectations and they are relevant. Evaluation of the results with F-measure is similar as it was with Purity. High level of this metrics indicates the final results are quite good quality. The 10% difference between between (F_1) and (F_5) indicates the Precision of the results is not achieved with nudge loss of the Recall what is typically problem in information retrieval algorithms.

5 Future Directions

The idea of the algorithm is similar to the word game where one thinks about a concept and the second one tries to guess it asking questions he or she may answer: yes, no, or don't know. We successfully implement this game in restricted domain of animals [14] where computer asking human questions guesses what object he or she has in mind[1]. Now this approach has been adopted for information retrieval in Simple Wikipedia where instead of questions, distinctive categories are presented to the user. The first implementation and evaluation results are promising.

In future we plan to extend our implementation to English Wikipedia. For now we successfully scale up the proposed method for the Polish Wikipedia that is one times bigger than presented here Simple Wikipedia (in terms of number of processed documents). Achieving the another order of magnitude requires to take into consideration some issues related to the effectiveness of processing such a large repository. This implies to implement some additional mechanisms for improving the speed of computations.

We also plan to improve document representation that is fundamental to obtain good results of natural language processing. Our idea is to introduce more background knowledge and capture some semantics. The approach is to map articles on activations of a semantic network and then calculate distances between them. WordNet dictionary [5] may be used for this purpose with word disambiguation techniques [15] that allow to map words to their proper synsets. We have made some research in this direction and the first results are promising [16]. Representation methods based on neurolinguistic inspirations [17] that use natural concept semantics will also be investigated.

[1] http://diodor.eti.pg.gda.pl

Acknowledgements. This work has been supported by the National Center for Research and Development (NCBiR) under research Grant No. SP/I/1/77065/1 SYNAT: "Establishment of the universal, open, hosting and communication, repository platform for network resources of knowledge to be used by science, education and open knowledge society".

References

1. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval, vol. 463. ACM press, New York (1999)
2. Castells, P., Fernandez, M., Vallet, D.: An adaptation of the vector-space model for ontology-based information retrieval. IEEE Transactions on Knowledge and Data Engineering 19, 261–272 (2007)
3. Kilicoglu, H., Fiszman, M., Rodriguez, A., Shin, D., Ripple, A., Rindflesch, T.: Semantic medline: A web application for managing the results of pubmed searches. In: Proceedings of the Third International Symposium for Semantic Mining in Biomedicine, pp. 69–76 (2008)
4. Raghavan, V., Bollmann, P., Jung, G.: A critical investigation of recall and precision as measures of retrieval system performance. ACM Transactions on Information Systems (TOIS) 7, 205–229 (1989)
5. Miller, G.A., Beckitch, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: An On-line Lexical Database. Cognitive Science Laboratory, Princeton University Press (1993)
6. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. Arxiv preprint cmp-lg/9709008 (1997)
7. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)
8. Xu, J., Croft, W.: Query expansion using local and global document analysis. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 4–11. ACM (1996)
9. Majewski, P., Szymański, J.: Text Categorization with Semantic Commonsense Knowledge: First Results. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part II. LNCS, vol. 4985, pp. 769–778. Springer, Heidelberg (2008)
10. Ceglowski, M., Coburn, A., Cuadrado, J.: Semantic search of unstructured data using contextual network graphs. National Institute for Technology and Liberal Education 10 (2003)
11. Cui-ru, W., Chun-hong, D.: An Improved Density-based DBSCAN Clustering Algorithm. Journal of Guangxi Normal University (Natural Science Edition) 4 (2007)
12. Rosell, M.: Introduction to text clustering (2008)
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
14. Szymanski, J., Duch, W.: Information Retrieval with Semantic Memory Model. Cognitive Systems Research (2011)
15. 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)
16. Szymański, J., Mizgier, A., Szopiński, M., Lubomski, P.: Ujednoznacznianie słow przy użyciu słownika WordNet. Wydawnictwo Naukowe PG TI 2008 18, 89–195 (2008)
17. Duch, W., Matykiewicz, P., Pestian, J.: Neurolinguistic approach to natural language processing with applications to medical text analysis. Neural Networks 21(10), 1500–1510 (2008)

Prototype Based Modelling for Ordinal Classification

Shereen Fouad and Peter Tino*

The University of Birmingham, Birmingham B15 2TT, United Kingdom
{saf942,P.Tino}@cs.bham.ac.uk
http://www.birmingham.ac.uk

Abstract. Many pattern analysis problems require classification of examples into naturally ordered classes. In such cases nominal classification schemes will ignore the class order relationships, which can have detrimental effect on classification accuracy. This paper introduces a novel ordinal Learning Vector Quantization (LVQ) scheme, with metric learning, specifically designed for classifying data items into ordered classes. Unlike in nominal LVQ, in ordinal LVQ the class order information is utilized during training in selection of the class prototypes to be adapted, as well as in determining the exact manner in which the prototypes get updated. Prototype based models are in general more amenable to interpretations and can often be constructed at a smaller computational cost than alternative non-linear classification models. Experiments demonstrate that the proposed ordinal LVQ formulation compares favorably with its nominal counterpart. Moreover, our method achieves competitive performance against existing benchmark ordinal regression models.

Keywords: Matrix Learning Vector Quantization (MLVQ), Ordinal Classification.

1 Introduction

Most classification algorithms focus on predicting data labels from nominal (non-ordered) classes. However, many pattern recognition problems involve classifying data into classes which have a natural ordering. This type of problem, known as ordinal classification or ordinal regression, is commonly seen in real life applications, such as information retrieval [1] and medical analysis [2]. In such problems, although it is still possible to use the conventional (nominal) methods, the order relation among the classes will be ignored, which may affect the stability of learning and the overall prediction accuracy. Lot of effort has already been devoted to the problem of ordinal classification in the machine learning literature. For instance, one stream of ordinal regression research assumes that ordinal labels originate from coarse measurements of a continuous variable. The labels are thus associated with intervals on the real line. A group of algorithms,

* Shereen Fouad and Peter Tino are with the School of Computer Science, The University of Birmingham.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 208–215, 2012.

known as threshold models, focuses on two main issues: **(i)** How to find the 'optimal' projection line, representing the assumed linear order of classes; **(ii)** How to optimally position thresholds defining the label intervals so that the margin of separation between neighbouring classes is maximized. For instance, in the SVM context, Shashua and Levin [3] proposed two large-margin principles: the fixed-margin principle and the sum of margins principle to handle the direction and multiple thresholds. This work was further extended by Chu and Keerthi [1] who proposed two Support Vector Ordinal Regression (SVOR) methods by optimizing multiple thresholds to define parallel discriminant hyperplanes. The first method (SVOR-EXC) explicitly imposes the ordering of the thresholds, whereas this is done implicitly in the second method (SVOR-IMC). Li and Lin [4] presented a reduction framework from ordinal regression to binary classification based on 'extended' examples, referred here to (SVM_EBC). Note that, the SVM based algorithms all suffer from high computational complexity (in the number of training points) [6]. Sun et al. [6] introduced a (non-SVM-based) Kernel Discriminant Learning for Ordinal Regression model (KDLOR) with a lower computational complexity .

In this paper, we propose a novel Learning Vector Quantization (LVQ) based model specifically designed for classifying data into ordered classes. LVQ, originally introduced by Kohonen in [7,8], constitutes a family of supervised learning multi-class classification algorithms. Compared to SVM type methods, prototype based models are in general more amenable to interpretations and can be constructed at a smaller computational cost. The function of such classifiers can be more directly understood because of the intuitive classification of data points to the class of their closet prototype (under a given metric). In particular, we extend the recently proposed modifications of LVQ, termed Matrix LVQ (MLVQ) [9,10], to the case of ordinal classification. As in nominal MLVQ, in Ordinal MLVQ (OMLVQ) the prototype positions, as well as the (global) metric in the data space can be modified. However, unlike in nominal MLVQ, in the proposed OMLVQ the class order information is utilized during training in selection of the class prototypes to be adapted, as well as in determining the exact manner in which the prototypes get updated.

This paper is organized as follows: Section 2 gives a brief introduction to the LVQ based methods related to this study. In section 3 we introduce a novel ordinal LVQ approach for classifying data with ordered labels. Experimental results are presented in section 4. Section 5 concludes the study by summarizing the key contributions.

2 Learning Vector Quantization (LVQ) and Its Extension Matrix LVQ (MLVQ)

Learning Vector Quantization (LVQ) constitutes a family of supervised learning algorithms that adapt class prototypes to the training data in an on-line manner [8]. Assume training data $(x_i, y_i) \in \mathbb{R}^m \times \{1, ..., K\}$, $i = 1, 2, ..., n$ is given, K is number of different classes. A typical LVQ network consists of a number of prototypes

$w_i \in \mathbb{R}^m$, $i = 1, 2, 3, ..., L$, which are characterized by their location in input space and their class label $c(w_i) \in \{1, ..., K\}$. Given a distance measure $d(x_i, w)$ in $\mathbb{R}^m$, classification is based on a winner-takes-all scheme: a data point $x_i \in \mathbb{R}^m$ is assigned to the label $c(w_j)$ of prototype w_j with $d(x, w_j) < d(x, w_i), \forall j \neq i$. During training, for each data point x_i with class label $c(x_i)$, the closest prototype with the same label is rewarded by pushing it closer to the training input; the closest prototype with different label is penalized by moving it away of the pattern x_i. The goal of learning is to adapt prototypes automatically such that $c(x_i)$ of x_i in the receptive fields[1] coincide with $c(w)$, that is the label of the corresponding prototype. Several modifications of this basic learning scheme have been proposed, aiming to achieve better approximation of decision boundaries and/or faster and more robust convergence. Recently, special attention was paid to schemes for manipulating the input space metric used to quantify the similarity between prototypes and feature vectors [9].

This paper focuses on a recently proposed prototype based algorithm, namely Matrix LVQ (MLVQ) [10], which allows the integration of a full adaptive matrix in the metric [9,10]. MLVQ is a new heuristic extension of the basic LVQ1 [7] with a full (e.g. not only diagonal elements) matrix tensor based distance measure. Given an $(m \times m)$ positive definite matrix $\mathbf{\Lambda}$, the algorithm uses a generalized form of the Euclidean distance

$$d^{\mathbf{\Lambda}}(x_i, w) = \sqrt{(x_i - w)^T \mathbf{\Lambda} (x_i - w)}. \tag{1}$$

Positive definiteness of $\mathbf{\Lambda}$ can be achieved by substituting $\mathbf{\Lambda} = \mathbf{\Omega}^T \mathbf{\Omega}$, where $\mathbf{\Omega} \in \mathbb{R}^{m \times m}$, $1 \leq l \leq m$ is a full-rank matrix. Furthermore, $\mathbf{\Lambda}$ needs to be normalized after each learning step to prevent the algorithm from degeneration. For each training pattern x_i, the algorithm implements Hebbian updates for the closest prototype w and for the metric parameter $\boldsymbol{\Omega}$. If $c(x_i) = c(w)$, then w is attracted towards x_i, otherwise w is repelled away (for more details, please consult [10]).

3 Ordinal MLVQ (OMLVQ) Classifier

This section presents a novel methodology based on MLVQ for classifying data with ordinal classes. We assume that we are given training data $(x_i, y_i) \in \mathbb{R}^m \times \{1, ..., K\}$, where $i = 1, 2, .., , n$, and K is the number of classes. In ordinal classification problem, it is assumed that classes are ordered $y_K > y_{K-1} > ... > y_1$, where $>$ denotes the order relation on labels. As in LVQ models, the proposed classifier is parameterized with L prototype-label pairs:

$$W = \{(w_q, k) \mid w_q \in \mathbb{R}^m, q \in \{1, ..., L\}, k \in \{1, ..., K\}\}. \tag{2}$$

[1] The receptive field of prototype w is defined as the set of points which pick this prototype as their winner.

We assume that each class $k \in \{1, 2, ..., K\}$, may be represented by P prototypes[2] collected in the set $W(k) = \{w \in W |\ c(w) = k\}$, leading to total number of $L = K \cdot P$ prototypes. The prototypes define a classifier by means of a winner-takes-all rule: a pattern $x_i \in \mathbb{R}^m$ is classified with the label of the closest prototype, $c(x_i) = c(w_j)$, $j = \arg\min_l d^{\boldsymbol{\Lambda}}(x_i, w_l)$, where $d^{\boldsymbol{\Lambda}}$ denotes the metric (1).

Whereas nominal versions of LVQ aim to position the class prototypes in the input space so that the overall miss-classification error is minimized, the proposed ordinal LVQ models adapt the class prototypes so that the average absolute error of class miss-labellings is minimized. In the next section we describe identification of prototypes to be modified, given each training input x_i.

3.1 Identification of Class Prototypes to Be Adapted

The initial step in adaptive response to each training instance x_i, $i = 1, 2, ..., n$, focuses on detecting the 'correct' and 'incorrect' prototype classes (with respect to $c(x_i)$). Subsequently, the correct prototypes will be pushed towards x_i, whereas the incorrect ones will be pushed away from x_i.

Due to the ordinal nature of labels, for each training instance x_i and prototype w_q, $q = 1, 2, ..., L$, the 'correctness' of the prototype's label $c(w_q)$ is measured through the absolute error loss function $H\,(c(w_q), c(x_i)) = |\ c(w_q) - c(x_i)\ |$. Given a rank loss threshold L_{min}, defined on the range of the loss function[3], the sets of 'correct' (or 'tolerable') prototype classes (denoted here as $N(c(x_i))^+$) and incorrect prototype classes (denoted here as $N(c(x_i))^-$) for input x_i read:

$$N\,(c(x_i))^+ = \{c(w_q) \in \{1, 2, 3.., , K\} \mid |c(w_q) - c(x_i)| \leq L_{min}\} \tag{3}$$

$$N\,(c(x_i))^- = \{c(w_q) \in \{1, 2, 3.., , K\} \mid |c(w_q) - c(x_i)| > L_{min}\}\,, \tag{4}$$

Given a training pattern x_i,

- for *correct prototypes* it makes sense to push towards x_i only the closest prototype from each class in $N(c(x_i))^+$. The set of correct prototypes to be modified given input x_i reads:

$$\begin{aligned} W(x_i)^+ = \{w_{z(k)} |\ \ & c(w_{z(k)}) = k \in N^+(c(x_i)), \\ & z(k) \quad = \arg\min_{l \in W_k} [d^{\boldsymbol{\Lambda}}(w_l, x_i)]\} \end{aligned} \tag{5}$$

  where W_k is the set of prototypes of class k.
- for *incorrect prototypes* from in $N(c(x_i))^-$ it is desirable to push away from x_i all incorrect prototypes lying in the 'neighbourhood' of x_i. In our case the neighbourhood will be defined as a sphere of radius D under the metric $d^{\boldsymbol{\Lambda}}$.

$$W(x_i)^- = \{w_z |\ c(w_z) \in N^-(c(x_i)),\ d^{\boldsymbol{\Lambda}}(w_z, x_i) < D\}. \tag{6}$$

[2] This imposition can be relaxed to a variable number of prototypes per class.
[3] In our case $[0, K-1]$.

3.2 Incremental Learning Algorithm

This section presents a new LVQ adaptation rule taking into account the order relationship among the classes. In particular, we generalize the MLVQ algorithm (see section 2) to the case of linearly ordered classes. We will refer to this new learning scheme as Ordinal MLVQ (OMLVQ). Unlike in nominal MLVQ, in OMLVQ each training iteration adapts multiple prototypes in $W(x_i)^+$ and $W(x_i)^-$, albeit to a different degree.

Given a training input x_i, the attractive and repulsive force applied to correct and incorrect prototypes w will decrease and increase, respectively, with growing $H(c(w), c(x_i))$. In addition, for incorrect prototypes w, the repulsive force will diminish with increasing distance form x_i, as one should worry less about correcting prototypes further away from x_i than about those lying in its neighbourhood. This is expressed in the following weighting schemes:

- For *correct prototypes* $w \in W(x_i)^+$ we propose a Gaussian weighting scheme,

$$\alpha^+ = \exp\left\{-\frac{(H(c(w), c(x_i)))^2}{2\sigma^2}\right\}, \tag{7}$$

  where, σ is the Gaussian kernel width.
- The *incorrect prototypes* $w \in W(x_i)^-$ are weighted as follows: Denote by ϵ_{max} the maximum rank loss error within the set $W(x_i)^-$, $\epsilon_{max} = \max_{w \in W(x_i)^-} H(c(w), c(x_i))$. The weight factor α^- for incorrect prototype $w \in W(x_i)^-$ is then calculated as

$$\alpha^- = \exp\left\{-\frac{(\epsilon_{max} - H(c(w), c(x_i)))^2}{2\sigma^2}\right\} \cdot \exp\left\{-\frac{(d^{\Lambda}(x_i, w))^2}{2\sigma'^2}\right\}, \tag{8}$$

  where σ' is the Gaussian kernel width for the distance factor in α^-.

Summary of Ordinal MLVQ (OMLVQ) Training Algorithm:

1. Initialize the prototype positions w[4] and the matrix tensor parameter[5] $\boldsymbol{\Omega}$.
2. **While** a stopping criterion is not reached do:
   (a) Select a training pattern x_i, $i \in \{1, 2, ..., n\}$, with class label $c(x_i)$.
   (b) Determine $N(c(x_i))^+$ and $N(c(x_i))^-$ based on (3) and (4), respectively.
   (c) Find $W(x_i)^+$ and $W(x_i)^-$ using (5) and (6).
   (d) Assign weight factors $\alpha^{\pm}$ to the selected prototypes (Eqs. (7) and (8)).
   (e) Update prototypes and the distance metric as follows:
      i. $\forall\ w \in W(x)^+$ do:
      $w = w + \eta_w \cdot \alpha^+ \cdot \boldsymbol{\Lambda} \cdot (x_i - w)$ (w dragged towards x_i)
      $\boldsymbol{\Omega} = \boldsymbol{\Omega} - \eta_{\boldsymbol{\Omega}} \cdot \alpha^+ \cdot \boldsymbol{\Omega} \cdot (x_i - w)(x_i - w)^T$ ($d^{\boldsymbol{\Lambda}}(x_i, w)$ is shrinked)

[4] Following [9,10], the means of P random subsets of training samples selected from each class k, where $k \in \{1, 2, ..., K\}$, are chosen as initial states of the prototypes. Alternatively, one could run a vector quantization with P centers on each class.

[5] By setting it equal to the identity matrix (Euclidean distance).

ii. $\forall\ w \in W(x)^-$ do:
$w = w - \eta_w \cdot \alpha^- \cdot \boldsymbol{\Lambda} \cdot (x_i - w)$ (w pushed away from x_i)
$\boldsymbol{\Omega} = \boldsymbol{\Omega} + \eta_{\boldsymbol{\Omega}} \cdot \alpha^- \cdot \boldsymbol{\Omega} \cdot (x_i - w)(x_i - w)^T$ ($d^{\boldsymbol{\Lambda}}(x_i, w)$ is increased).

where η_w, $\eta_{\boldsymbol{\Omega}}$ are positive learning rates for prototypes and metric[6], respectively. They decrease monotonically with time as [10]: $\eta_u \leftarrow \frac{\eta_u}{1+\tau(t-1)}$, where $u \in \{\boldsymbol{\Omega}, w\}$, $\tau > 0$ determines the speed of annealing[7], and t indexes the number of training epochs done. As in the original MLVQ (section 2), to prevent the algorithm from degeneration, $\boldsymbol{\Omega}$ is normalized after each learning step so that $\sum_i \boldsymbol{\Lambda}_{ii} = 1$ [9,10].

3. **End While**

4 Experiments

We evaluated performance of the proposed ordinal prototype-based model (OMLVQ) through a set of experiments conducted on five benchmark ordinal regression datasets[8] used in [1,6,4,5]. The OMLVQ framework was assessed against its nominal counterpart MLVQ and compared with four benchmark ordinal regression methods: two threshold SVM based models (SVOR-IMC and SVOR-EXC [1]), one reduction framework (SVM-EBC [4]) and a Kernel Discriminant Learning for Ordinal Regression method (KDLOR [6]). For comparison purposes, on each dataset we conducted the same pre-processing and experimental settings as described in [1,4,6,5]. The input vectors were normalized to have zero mean and unit variance. Data labels were discretized into ten ordinal quantities using equal-frequency binning. Each dataset was randomly partitioned (20 times) into training/test splits. On each data set, the algorithm parameters were tunned through 5-fold cross-validation on the training set[9]. Our experiments utilize two evaluation metrics to measure accuracy of predicted class $\hat{y}$ with respect to true class y on a test set :**(i) Mean Zero-one Error (MZE)**: the fraction of incorrect predictions, i.e. $MZE = \frac{\sum_{i=1}^{v} I(y_i \neq \hat{y}_i)}{v}$, where v is the number of test examples and $I(y_i \neq \hat{y}_i)$ denotes the indicator function returning 1 if the predicate holds and 0 otherwise. **(ii) Mean Absolute Error (MAE)**: the average deviation of the prediction from the true rank, i.e. $MAE = \frac{\sum_{i=1}^{v} |y_i - \hat{y}_i|}{v}$.

A. Comparison between MLVQ and OMLVQ: The averages MZE and MAE results (over 20 trials), along with standard deviations (represented by error bars), are shown in Fig. 1. On average, across the 5 datasets the OMLVQ algorithm outperforms the baseline MLVQ by relative improvement of 7% and 18% on MZE and MAE, repectively. As expected, OMLVQ method demonstrate stronger improvement over MLVQ in terms of MAE, rather than MZE.

[6] The initial learning rates are chosen individually for every application through cross-validations.

[7] In our experiments τ was set to 0.0001.

[8] Available at `http://www.gatsby.ucl.ac.uk/~chuwei/ordinalregression.html`

[9] Except for the *Triazines* dataset in which we implemented 10-fold cross-validations for fair comparisons with (SVOR-IMC) and (SVOR-EXC) results reported in [5].

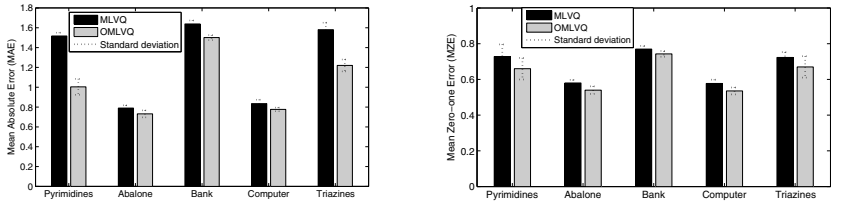


Fig. 1. MAE and MZE test results (left and right plots, respectively) of the nominal LVQ model (MLVQ) and its ordinal counterpart (OMLVQ)

Table 1. MZE test results along with standard deviations, (±) across 20 training/test re-sampling, for four algorithms. Best results are marked with bold font

Dataset	Dimension	training/testing	KDLOR	SVOR-IMC	SVOR-EXC	OMLVQ
Pyrimidines	27	50/24	0.739±(0.050)	0.719±(0.066)	0.752±(0.063)	**0.660 ±(0.060)**
Abalone	8	1000/3177	0.740±(0.020)	0.732±(0.007)	0.736±(0.011)	**0.545±(0.021)**
Bank	32	3000/5182	0.745±(0.0025)	0.751±(0.005)	**0.744±(0.005)**	0.756 ±(0.016)
Computer	21	4000/4182	0.472±(0.020)	0.473±(0.005)	**0.462±(0.050)**	0.535± (0.019)
Triazines	60	100/86	N/A	0.710±(0.020)	0.720 ±(0.000)	**0.670±(0.050)**

Table 2. MAE test results, along with standard deviations (±) across 20 training/test re-sampling, for five algorithms. Best results are marked with bold font

Dataset	SVM_EBC	KDLOR	SVOR-IMC	SVOR-EXC	OMLVQ
Pyrimidines	1.304±(0.040)	1.100±(0.100)	1.294±(0.204)	1.331±(0.193)	**1.004±(0.080)**
Abalone	1.383±(0.004)	1.400±(0.050)	1.361±(0.013)	1.391±(0.021)	**0.732±(0.035)**
Bank	1.404±(0.002)	1.450± (0.020)	**1.393±(0.011)**	1.512±(0.017)	1.501±(0.025)
Computer	**0.565±(0.002)**	0.601±(0.025)	0.596±(0.008)	0.602±(0.009)	0.776±(0.018)
Triazines	N/A	N/A	1.270± (0.070)	1.340±(0.000)	**1.220±(0.060)**

B. Comparison with Benchmark Ordinal Regression Approaches: MZE and MAE test results, along with standard deviations over 20 training /test re-samplings, are listed in Tables 1 and 2, respectively[10]. In comparison with other methods, OMLVQ algorithm achieves the lowest MZE and MAE results on three datasets (*Pyrimidines*, *Abalone* and *Triazines*) and competitive performance on the *Bank* dataset. Note that on the *Computer* data set, where the OMLVQ method was beaten by the competitors, the original MLVQ method performed poorly as well (see Fig. 1). We hypothesize that the class distribution structure of this data set may not be naturally captured by the prototype based methods.

[10] MZE results of the (SVM_EBC) model is not listed because only MAE of this algorithm was recorded in [4]. In addition, the *Triazines* dataset results of (SVM_EBC) and (KDLOR) algorithms are not listed (denoted in tables as N/A) because they were not reported by their authors in [4,6].

5 Conclusion

This paper introduced a novel prototype-based learning methodology, especially tailored for classifying data with ordered classes. Based on the existing nominal Matrix LVQ (MLVQ) [9,10] we proposed a new Ordinal MLVQ (OMLVQ). Unlike in nominal MLVQ, in OMLVQ the class order information is utilized during training in selection of the class prototypes to be adapted, as well as in determining the exact manner in which the prototypes get updated. In particular, the prototypes are adapted so that the ordinal relations amongst the prototype classes are preserved, reflected in reduction of the overall mean absolute error. Experimental results on five benchmark datasets empirically verify the effectiveness of our OMLVQ framework when compared with its standard nominal MLVQ version. The mean zero-one error (MZE) and mean absolute error (MAE) rates of the proposed methods were considerably lower, with more pronounced improvements on the MAE rates. In addition, in comparison with existing benchmark ordinal regression methods, our ordinal MLVQ framework attained a competitive performance in terms of MZE and MAE measurements.

References

1. Chu, W., Keerthi, S.S.: Support Vector Ordinal Regression. Neural Computation 19(3), 792–815 (2007)
2. Cardoso, J.S., Pinto da Costa, J.F., Cardoso, M.J.: Modelling Ordinal Relations with SVMs: An Application to Objective Aesthetic Evaluation of Breast Cancer Conservative Treatment. Neural Networks 18(5-6), 808–817 (2005)
3. Shashua, A., Levin, A.: Ranking with Large Margin Principle: Two Approaches. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems 15, pp. 937–944. MIT Press, Vancouver (2002)
4. Li, L., Lin, H.: Ordinal Regression by Extended Binary Classification. In: Schölkopf, B., Platt, J.C., Hofmann, T. (eds.) Advances in Neural Information Processing Systems 19, pp. 865–872. MIT Press, Vancouver (2007)
5. Van Belle, V., Pelckmans, K., Suykens, J.A.K., Van Huffel, S.: MINLIP: Efficient Learning of Transformation Models. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009, Part I. LNCS, vol. 5768, pp. 60–69. Springer, Heidelberg (2009)
6. Sun, B.Y., Li, J., Wu, D.D., Zhang, X.M., Li, W.B.: Kernel Discriminant Learning for Ordinal Regression. IEEE Transactions on Knowledge and Data Engineering 22(6), 906–910 (2010)
7. Kohonen, T.: Learning Vector Quantization for Pattern Recognition. Technical report, TKKF-A601, Laboratory of Computer and Information Science, Department of Technical Physics, Helsinki University of Technology. Espoo, Finland (1986)
8. Kohonen, T.: Learning Vector Quantization. The handbook of brain theory and neural networks. MIT Press, Cambridge (1998)
9. Schneider, P., Biehl, M., Hammer, B.: Adaptive Relevance Matrices in Learning Vector Quantization. Neural Computation 21(12), 3532–3561 (2009)
10. Schneider, P.: Advanced Methods for Prototype-based Classification. Doctoral Dissertation. University of Groningen (2010), http://irs.ub.rug.nl/ppn/327245379

A Genetic Graph-Based Clustering Algorithm

Héctor Menéndez and David Camacho

Departamento de Ingeniería Informática, Escuela Politécnica Superior,
Universidad Autónoma de Madrid,
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain
{hector.menendez,david.camacho}@uam.es
http://aida.ii.uam.es

Abstract. The interest in the analysis and study of clustering techniques have grown since the introduction of new algorithms based on the continuity of the data, where problems related to image segmentation and tracking, amongst others, makes difficult the correct classification of data into their appropriate groups, or clusters. Some new techniques, such as Spectral Clustering (SC), uses graph theory to generate the clusters through the spectrum of the graph created by a similarity function applied to the elements of the database. The approach taken by SC allows to handle the problem of data continuity though the graph representation. Based on this idea, this study uses genetic algorithms to select the groups using the same similarity graph built by the Spectral Clustering method. The main contribution is to create a new algorithm which improves the robustness of the Spectral Clustering algorithm reducing the dependency of the similarity metric parameters that currently affects to the performance of SC approaches. This algorithm, named Genetic Graph-based Clustering (GGC), has been tested with different synthetic and real-world datasets, the experimental results have been compared against classical clustering algorithms like K-Means, EM and SC.

Keywords: Machine Learning, Clustering, Spectral Clustering, Genetic Algorithms.

1 Introduction

The unsupervised learning methods are mainly based on clustering techniques [3]. These techniques were designed to find hidden information or features in a dataset grouping the data with similar properties in clusters. The different methods are divided in three main categories[7]: *partitional* (consists in a disjoint division of the data where each element belongs only to a single cluster); *overlapping* or non-exclusive (allows each element to belong to multiple clusters) and *hierarchical* (nests the clusters formed through a partitional clustering method creating bigger partitions and grouping the clusters by hierarchical levels).

This work is focused on the first category: partitional clustering which also has three main approximations[3]: *Parametric or Model-based clustering* (consists on an estimator based on a mixture of probabilities whose parameters are

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 216–225, 2012.

estimated, it fixes the model to the dataset); *Non-Parametric clustering* (there is not an initial probability model or estimator) and *Semiparametric clustering* (a combination of both methods).

This work is based on a well-known technique of Non-Parametric Partitional Clustering: Spectral Clustering (SC). It was introduced by Ng et al. in [10]. The algorithm is divided in three main steps:

1. A Similarity Function is applied to all the pairs of data elements to generate a Similarity Graph. There are three different kind of similarity graph: **the ϵ-neighbourhood graph** (all the components whose pairwise distance is smaller than ϵ are connected), **the k-nearest neighbour graph** (the vertex v_i is connected with vertex v_j if v_j is among the k-nearest neighbours of v_i) and **the fully connected graph** (all points with positive similarity are connected with each other).
2. The Laplacian Matrix (or Spectrum) of the Similarity Graph is extracted to study its eigenvectors. There are three different Laplacian matrices[12]. They define different versions of the SC algorithm: **Unnormalized SC** (the Laplacian matrix is: $L = D - W$), **Normalized SC** (the Laplacian matrix is: $L_{sym} = D^{-1/2}LD^{-1/2}$) and **Normalized SC related to Random Walks** (the Laplacian matrix is: $L_{rw} = D^{-1}L$).
3. Kmeans (or other partitional clustering technique) is applied to the matrix formed by the k-first eigenvectors to discriminate the information and assign the final clusters.

The main problem of the SC algorithm is the computation of the eigenvectors and eigenvalues of the Laplacian Matrix and the effect that they produce on the convergence of the algorithm. is how to compute the eigenvector and the eigenvalues of the Laplacian matrix of this similarity graph. The theoretical analysis of the convergence is justified using the perturbation theory [12], random walks and graph cut theory [12]. Some of the main problems of Spectral Clustering are related to the consistency of the two typical methods used in the analysis: normalized and un-normalized spectral clustering. A deep analysis about the theoretical effectiveness of normalized clustering over un-normalized was carry out by von Luxburg in [13].

Other problem of the Spectral Clustering algorithm is its sensitivity to the definition of the similarity function. It produces several problems when there is noisy information as Chang and Yeung exposed in [2]. Some solutions to this problem were based on the improvements of the parameters selection for the similarity function [2]. Other solutions are focused on the selection of the partitional clustering algorithm for the third step of SC [14]. This work develops a new algorithm based on Genetic Algorithms (GA) to improve the robustness of the clusters selection taking the similarity graph as a starting point.

Genetic Algorithms have been traditionally used in optimization problems. The complexity of the algorithm depends on the codification and the operations that are used to reproduce, cross, mutate and select the different individuals (chromosomes) of the population [4]. The algorithm applies a fitness function which guides the search to find the best individual of the population.

Different approximation of genetic codifications to the clustering problem were profound studied by Hruschka et al. in [7]. They show the different codifications, operations and fitness functions applied in several genetic algorithms to solved the clustering problem. Our previous work was also focused on resolve this problem using GA, but it was centred on overlapping clustering [1].

This work presents a Genetic Graph-based Clustering (GGC) algorithm which is inspired on the Spectral Clustering algorithm (it takes the same Similarity Graph as a starting point) and improves the robustness of the solution. The algorithm is experimentally compared with Spectral Clustering, Kmeans [9] and Expectation Maximization (EM) [9] to test its accuracy. The experimental study is also focused in a comparison between the robustness of the SC and GGC algorithms.

The rest of the work is structured as follows: Section 2 presents the Genetic Graph-based Clustering Algorithm; Section 3 shows the experimental results. Finally, Section 4 gives the conclusions and future work.

2 Genetic Graph-Based Clustering Algorithm (GGC)

This section explains the algorithm which has been implemented. It is mainly based on a simple Genetic Algorithm (GA). It is necessary to give a number of clusters initially. The algorithm begins with a Similarity Graph in the same manner that the Spectral Clustering algorithm. The population of the GGC algorithm is a set of possible solutions (partitions) which evolves until the best solution is found or the number of generations is ended. The fitness function is a quality measure for the solutions.

2.1 Codification and Genetic Operators

The codification is a simple vector-based numerical representation. Each individual is a n-dimensional vector (where n is the number of data instances) which has integer values between 1 and the number of clusters. They represent a cluster selection for the dataset. During the evolution process, the operators can create invalid individuals. These individuals represent solutions where one or more clusters have no elements. In this problem of partitional clustering these solutions are not valid because the number of clusters is initially given. To avoid the invalid individuals generation problem, they receive a 0 fitness value. The operators used in the GGC algorithm are the traditional ones extracted from the GA literature, they can be briefly summarized as follows:

- **Selection:** The selection process selects a subset of the best individuals. These chromosomes are reproduced and also pass to the next generation. It is called a $(\mu + \lambda)$ selection [4], where μ represents those chromosomes which are chosen, and λ the new chromosomes generated.
- **Reproduction:** The reproduction randomly selects two individuals (using the classical wheel algorithm [4]), and applies the crossover operation to the chosen chromosomes creating two new individuals.

Algorithm 1. Pseudo-code of the Fitness Function

Require: A n-vector of elements with values between 0 and k where k is the number of clusters and a variable $neighbours$ which represents the number of neighbours for the KNN measure.

Ensure: A value between 0 and 1 which corresponds with the fitness achieved.

```
1: TotalKNN = 0.;
2: TotalMC = 0.;
3: Generate the set of k Clusters: C.
4: for all C_a ∈ C do
5:     if C_a = ∅ then
6:         return 0
7:     end if
8:     SumKNN = 0; SumMC = 0.
9:     for all ind ∈ C_a do
10:        SumKNN += PofKNN(neighbours, ind) {It calculates the percentage
           of neighbours for the individual ind which are assigned to the same
           cluster.}
11:        SumMC += AvEdWCut(ind) {It calculates the average value of the edge
           weights which have been cut from ind.}
12:    end for
13:    TotalKNN += SumKNN / |C_a|; {|C_a| represents the number of elements of
       C_a.}
14:    TotalMC += SumMC / |C_a|;
15: end for
16: return TotalKNN/|C| × (1 − TotalMC/|C|)
```

- **Crossover:** The main problem of the crossover operation is those individuals which have different numerical values but represents the same solution. These individuals need to be relabelled before the application of the operation. For this reason, a measure which compares the number of commons elements between the clusters is used to find the similarity degree of the chromosomes. After, one of the two chromosomes is relabelled trying to maximize the similarity. Finally, the crossover exchanges strings of numbers between the two chromosomes (both string have the same length).
- **Mutation:** The mutation randomly choose different chromosomes to change the values of some of their alleles. The new value is a random number between 1 and the number of clusters.

2.2 The GGC Fitness Function

The fitness function is a combination of the classical K-Nearest Neighbourhood (KNN) [9] algorithm and the Minimal Cut measure [11]. KNN assigns an element to a cluster if its neighbours are in the same cluster. It is useful to ensure the continuity condition that is common in the Spectral Clustering solutions. To control the separation between the elements of the clusters, the Minimal Cut measure is used. It guarantees that those elements which clearly belongs to

different clusters are not assigned to the same cluster. The K value for KNN is initially given.

Algortihm 1 shows the pseudo-code of the fitness: KNN covers all the nodes and check if the K-closest elements are in the same cluster (lines 9 to 12). The fitness value of this metric is the mean of the percentage of well-classified neighbours of all the individuals in a cluster (lines 10 and 13). The Minimal Cut measure calculates the average value edge weights which have been removed (lines 11 and 14). The final value of the fitness if the product of the KNN metric and the subtraction between one and the Minimal Cut metric (line 16), both metrics have the same range: [0,1]. Therefore, the algorithm maximizes the value of $\frac{TotalKNN}{|C|} \times \left(1 - \frac{TotalMC}{|C|}\right)$ (line 16) where:

$$TotalMC = \sum_{x \in C} \frac{\sum_{y \notin C_x} w_{xy}}{|\{y | y \notin C_x\}|}$$

$$TotalKNN = \sum_{x \in C} \frac{|\{y | y \in \Gamma(x) \wedge y \in C_x\}|}{|\Gamma(x)|}$$

In these formulas, C represents the set of clusters and $\Gamma(x)$ represents the neighbourhood of the element x. It reduces the weight values of the edges which are cut and improve the proximity of the neighbours.

3 Experimental Results

This section shows the different experiments carried out to evaluate the behaviour of our approach. These experiments are both synthetic and real-world experiments. First, the experiments analyse the distance dependency problem of the Spectral Clustering algorithm compared with the GGC algorithm. This initial analysis shows the robustness of the algorithm to the metric parameters. Second, the GGC algorithm results are compared with other algorithms (Kmeans, EM and Spectral Clustering) using synthetic datasets. Finally, these algorithms are applied to real-world datasets, which are classified, to test their results.

The parameters of the GGC algorithm have been experimentally fixed. To find these values, 100 experiments have been executed using the synthetic data (see Section 3.2).the selected parameters are:

- Population: 200
- Generations: 2000
- Crossover probability: 0.3
- Mutation probability: 0.5
- Selection $(\mu + \lambda)$: The 50^{th} best individual are selected from the previous generation.
- K value of the KNN metric: 2

3.1 The Robustness of the GGC Algorithm

An important problem of the Spectral Clustering algorithm is its dependency to the parameters of the similarity function. The GGC algorithm has been designed to avoid this problem. The KNN metric which is applied in the fitness calculation provides a higher robustness to the algorithm compared to the Spectral Clustering algorithm, it does not depend of the order of magnitude of the distances calculated by the metric. Figure 1 shows a clear example. In this case, the Spectral Clustering algorithm (implemented in the "kernlab" package of CRAN [8]) is compared with the GGC algorithm. In the "kernlab" package, Karatzoglou et al. implements the Random Walks Normalized Spectral Clustering algorithm. They use the Gaussian RBF Kernel to set the similarity graph. It is defined by: $K_{ij} = e^{-\sigma||x_i - x_j||^2}$, where K is the similarity matrix, x_i, x_j are data instances, and σ is the parameter which changes the order of magnitude. The experimental results show that the clustering technique clearly depends of the σ parameter. Figure 1 shows the different clustering results of the Spectral Clustering and the GGC algorithm modifying the σ parameter.

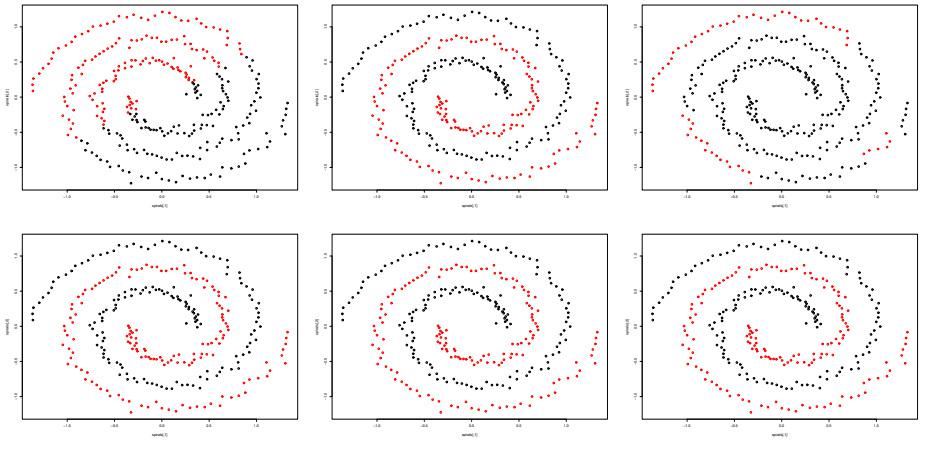


Fig. 1. Spectral Clustering and Genetic Algorithm results for the spirals [8] dataset with $\sigma = 2$, $\sigma = 500$, $\sigma = 2000$, respectively

These results show that the parameters used in the definition of the kernel are very important because these parameters defined the degree of the similarity. Ng et al. introduced a method to calculate the optimal σ in [10], however, as Figure 2 shows, this technique is not always enough. GGC obtains always the same results because it consider metrics which do not depend of the value of the distances, they depend of the order relation between the distances.

3.2 Experiments on Synthetic Data

In this section the different datasets which are used for the experimentation are explained and analysed. These datasets have been extracted from different clustering works which study the behaviour of the algorithms in difficult conditions.

Data Description. The GGC algorithm has been tested with different datasets. These datasets are 2-Dimensional data which can be separated by human intuition but are problematic for the classical clustering algorithms. The following datasets have been analysed:

- **Aggregation**[6]: This dataset is composed by 7 clusters, some of them can be separated by parametric clustering.
- **Compound**[15]: There are 6 clusters with are only separable by non-parametric methods (or special kernels if parametric clustering is applied).
- **Spiral**[2]: In this case, there are 3 spirals close to each other.

Results of Data Test. Figure 2 shows the classification results of the different datasets. Table 1 shows the best fitness values achieved by the GGC algorithm. In these cases the σ parameter to generate the similarity matrix of the Spectral Clustering and the GGC algorithms is 100 (it has been approximated using the method described by Ng et al. [10]). All the algorithms have been run 50 times and their best results have been selected. GGC and SC use the RBF kernel. EM and Kmeans use the Euclidean distance metric.

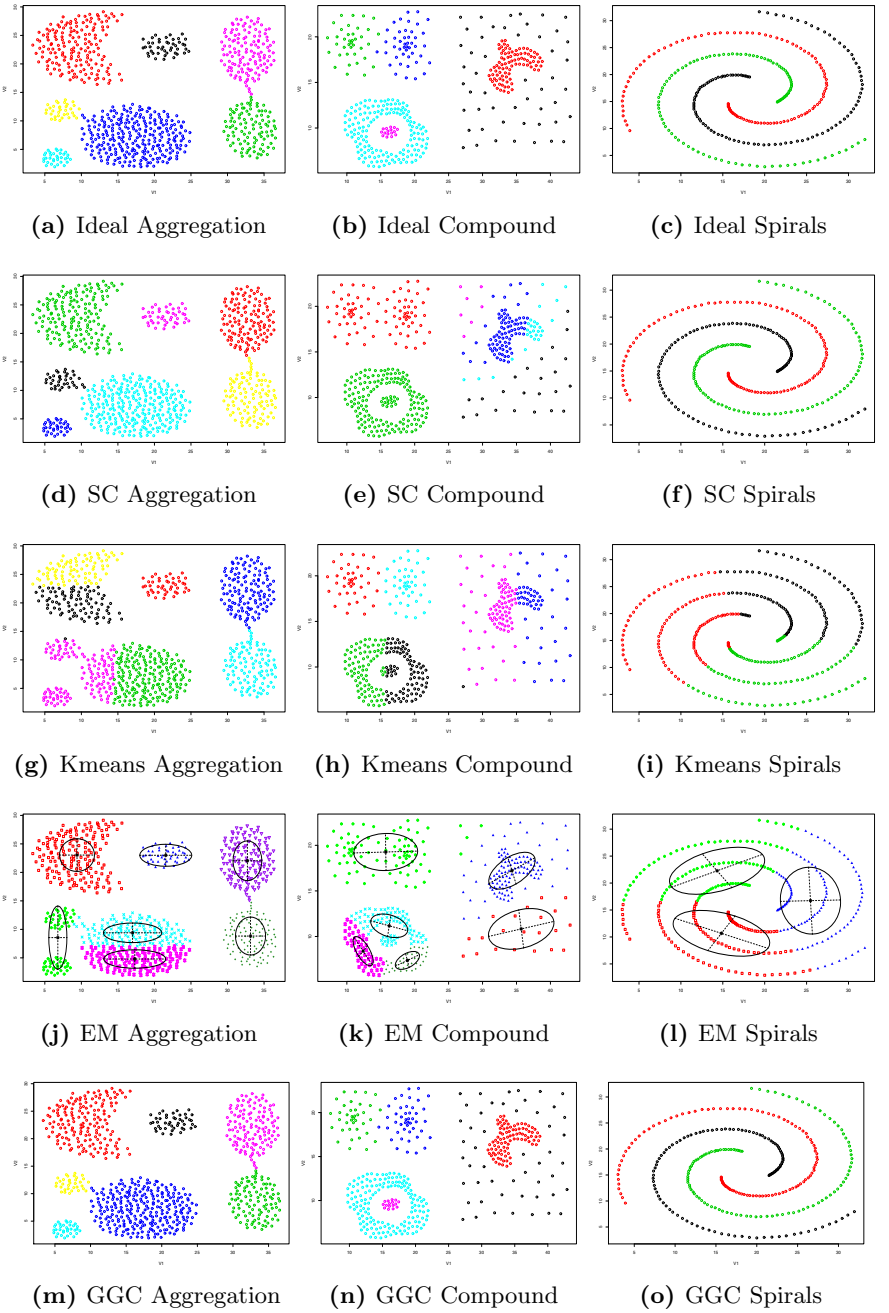
GGC and SC correctly classify Aggregation (GGC achieves a fitness value of 0.9928 which is the maximum value of fitness achieved by the algorithm; it is a consequence of those elements which could belong to two clusters) while EM and Kmeans have problems related to the form of the data. These problems could be a consequence of local minimum convergence for the centroids. Compound is impossible to classify with these parametric algorithms and the Euclidean distance. Also SC has problems related to the different distributions of the data. The GGC algorithm correctly classifies the Compound problem with a fitness value of 0.9552 (this value is also the maximum fitness achieved by the algorithm; in this case, there are elements assigned to different clusters which are closed to other clusters). Finally, Spirals classification is also impossible for the parametric methods while GGC and SC classify it correctly (in this case GGC achieves the maximum fitness).

Table 1. Fitness values achieved by GGC (see Figure 2)

Dataset	Fitness achieved
Aggregation	0.9928
Compound	0.9552
Spiral	1.0

3.3 Experiments on UCI Datasets

In this section the experiments are focused on real-world datasets which have been previously classified. Here, the accuracy of the algorithm is tested. In these databases the correct number of clusters is known. The measures used are the Euclidean Distance and the RBF kernel because they are the best known of the dissimilarity measures for these databases and they have been employed in previous works for all the methods used here. The GGC algorithm has been



(a) Ideal Aggregation **(b)** Ideal Compound **(c)** Ideal Spirals

(d) SC Aggregation **(e)** SC Compound **(f)** SC Spirals

(g) Kmeans Aggregation **(h)** Kmeans Compound **(i)** Kmeans Spirals

(j) EM Aggregation **(k)** EM Compound **(l)** EM Spirals

(m) GGC Aggregation **(n)** GGC Compound **(o)** GGC Spirals

Fig. 2. Each row represents the cluster selection of each algorithm for the synthetic datasets from left to right: "aggregation", "compound", "spiral". The rows represents from top to bottom: the ideal results, the Spectral Clustering results, the Kmeans results, the EM results and the GA results.

applied on 2 real and classified datasets (without missing values) extracted from the UCI Machine Learning Repository [5]:

- **Iris**: This dataset is a well-know dataset. It has 150 instance of 3 different classes (50 in each class). Each class refers to a type of iris plant. Each instance has 4 attributes.
- **Wine**: This dataset has 178 instance of 3 different classes (not balanced). Each class refers to a type of wine. Each instance has 13 attributes.

Results of the Data Test. The experiments have followed the same procedure that they followed in the synthetic datasets experiments. Also the value of σ has been approximated to 100. The results for the Iris show that EM is the best classifier (with an accuracy of the 96,67 %) and the GGC algorithm is the second (92%). The results for the Wine datasets shows that all the algorithm obtain high accuracy values (higher than the 95 %), and the GGC algorithm obtains a perfect classification with the maximum fitness value. These results are a consequence of the data distribution. Iris dataset has instances of different classes which are closed to each other, the GGC algorithm has problems to discriminate the boundary of the clusters specially when there are intersections between the clusters. The fitness value of the Iris is the higher that the algorithm has achieved, it shows that there are instance which belongs to different cluster but are closed to each other. In the case of the Wine dataset, the classes are clearer separated (as the different clustering techniques show). It improves the results of the GGC algorithm, because the boundary is clearer.

Table 2. Experimental results obtained using the UCI datasets

	Iris dataset	**Wine dataset**
Kmeans best classification	89.33%	95.50 %
EM best classification	96.67%	97.19%
Spectral Clustering best classification	89.33%	95.50%
GGC best classification	92% (Fitness=0.9946)	100% (Fitness=1)

4 Conclusions and Future Work

This work presents a new clustering method inspired by the Spectral Clustering algorithm and based on Genetic Algorithms. The GGC algorithm is defined using simple codification and operations. The main contribution of the algorithm is the fitness selection. GGC uses KNN and Minimum Cut measures. It is applied to the similarity graph which is generated in the first step of the Spectral Clustering algorithm. The combination of these measures improves the robustness of the algorithm giving a higher independence of the parameters of the similarity function. The results of Section 3 show that the new algorithm obtains good results for both synthetic and real-world datasets.

The future work will be focused on several improvements that could be made to the GGC algorithm. The effects of noisy information could be deeply analysed. The number of clusters could be automatically selected using strategies

such as cross-validation. Finally, other fitness functions which could improve the convergence, and the clusters quality, of the GGC algorithm will be studied.

Acknowledgements. This work has been partly supported by: Spanish Ministry of Science and Education under project TIN2010-19872.

References

1. Bello, G., Menéndez, H., Camacho, D.: Using the Clustering Coefficient to Guide a Genetic-Based Communities Finding Algorithm. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 160–169. Springer, Heidelberg (2011)
2. Chang, H., Yeung, D.-Y.: Robust path-based spectral clustering. Pattern Recogn. 41(1), 191–203 (2008)
3. Cios, K.J., Swiniarski, R.W., Pedrycz, W., Kurgan, L.A.: Unsupervised learning: Clustering. In: Data Mining, pp. 257–288. Springer, US (2007)
4. Coley. An Introduction to Genetic Algorithms for scientists and engineers. World Scientific Publishing (1999)
5. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
6. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. ACM Trans. Knowl. Discov. Data 1(1) (March 2007)
7. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., de Carvalho, A.C.P.L.F.: A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39(2), 133–155 (2009)
8. Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A.: kernlab – an S4 package for kernel methods in R. Journal of Statistical Software 11(9), 1–20 (2004)
9. Larose, D.T.: Discovering Knowledge in Data. John Wiley & Sons (2005)
10. Ng, A., Jordan, M., Weiss, Y.: On Spectral Clustering: Analysis and an algorithm. In: Dietterich, T., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems, pp. 849–856. MIT Press (2001)
11. Schaeffer, S.E.: Graph clustering. Computer Science Review 1(1), 27–64 (2007)
12. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
13. von Luxburg, U., Belkin, M., Bousquet, O.: Consistency of spectral clustering. The Annals of Statistics 36(2), 555–586 (2008)
14. Wang, H., Chen, J., Guo, K.: A genetic spectral clustering algorithm. Journal of Computational Information Systems 7(9), 3245–3252 (2011)
15. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers C-20(1), 68–86 (1971)

Echo State Networks for Seasonal Streamflow Series Forecasting

Hugo Siqueira[1], Levy Boccato[2], Romis Attux[2], and Christiano Lyra Filho[1]

[1] Systems Engineering Department (DENSIS)
[2] Department of Computer Engineering and Industrial Automation (DCA),
School of Electrical and Computer Engineering, University of Campinas – UNICAMP,
Campinas-SP, Brazil
{hugo,chrlyra}@densis.fee.unicamp.br,
{lboccato,attux}@dca.fee.unicamp.br

Abstract. The prediction of seasonal streamflow series is very important in countries where power generation is predominantly done by hydroelectric plants. Echo state networks can be safely regarded as promising tools in forecasting because they are recurrent networks that have a simple and efficient training process based on linear regression. Recently, Boccato et al. proposed a new architecture in which the output layer is built using a principal component analysis and a Volterra filter. This work performs a comparative investigation between the performances of different ESNs in the context of the forecasting of seasonal streamflow series associated with Brazilian hydroelectric plants. Two possible reservoir design approaches were tested with the classical and the Volterra-based output layer structures, and a multilayer perceptron was also included to establish bases for comparison. The obtained results show the relevance of these networks and also contribute to a better understanding of their applicability to forecasting problems.

Keywords: Echo State Networks, Volterra Filtering, PCA, Forecasting, Monthly Seasonal Streamflow Series.

1 Introduction

The forecasting of monthly streamflow series is a very important problem for countries that generate electric power using hydroelectric plants. This is the case of Brazil, a country for which this source is responsible for more than 80% of all generated electric power. Under these circumstances, important decisions about energy planning and generation, as well as pricing strategies, are highly dependent on accurate predictions of these series [10][13]. However, their non-stationary and seasonal character represents a significant obstacle that any prediction strategy must cope with. Besides, it is crucial that the chosen prediction structure be capable of exploring the relationship existing between the samples of the series [3].

In this context, recurrent neural networks (RNNs) emerge as a promising alternative as the presence of feedback connections can engender a memory element potentially

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 226–236, 2012.

beneficial in terms of prediction accuracy. Unfortunately, well-known difficulties associated with the training process of these networks, such as fading gradient and the possibility of instability, hamper, to a certain extent, their practical application [6].

An interesting alternative to circumvent these difficulties was proposed by Jaeger [8]: a novel architecture characterized by the existence of a dynamic reservoir with fixed parameters – which acts as an intermediate recurrent layer – and a linear output layer adjusted with the aid of supervised strategies. The proposal received the name of echo state network (ESN), and became a pillar of the research field known as reservoir computing (RC) [9]. The attractiveness of ESNs led to alternative design strategies, like the proposal of Ozturk et al. [11], which uses the project of linear systems based on Kautz principle to diversify the signals generated by the reservoir.

The possibility of allying the processing capability of a recurrent structure to a simple training process encourages the application of ESNs in the problem of streamflow series prediction. This perspective has been initially addressed by Sacchi et al. [12]. However, the question as to whether ESNs can be interesting options to this problem is far from being exhausted. For instance, the potential advantages of using a data pre-processing stage still have not been fully explored. Moreover, different reservoir design methods, such as that of Ozturk et al. [11], and other ESN architectures should be tested.

Recently, Boccato et al. [1] [2] proposed a new ESN architecture characterized by: (*i*) the use of Volterra filters at the output layer, which allows a more extensive exploitation of the higher-order statistics of the reservoir signals while preserving the simplicity of the training process; (*ii*) the application of the data compression technique known as Principal Component Analysis (PCA) as a pre-processing stage before the reservoir states are transmitted to the output layer. This proposal was capable of leading to significant performance improvements in different signal processing tasks, such as channel equalization and source separation [1] [2], but still has not been analyzed in depth in the context of stochastic time series prediction.

In this work, we present an up-to-date analysis of the applicability of ESNs to the problem in question by making a comparison between different reservoir design strategies – those proposed by Jaeger [8] and Ozturk et al. [11] – and between the standard linear readout and the aforementioned proposal of Boccato et al. [2] [13].

It is worth mentioning that these structures have been recently applied to the forecasting of the seasonal streamflow series from Furnas hydroelectric plant, in Brazil, and the preliminary results have indicated potential performance improvements with the use of the architecture of Boccato et al. [2]. Here, additional elements for comparison are brought by a feedforward multilayer perceptron (MLP) trained with the aid of the improved Scaled Conjugate Gradient Method (SCGM) [5]. Additionally, the repertoire of investigative scenarios is expanded and includes the streamflow series from three important Brazilian hydroelectric plants: Emborcação, Furnas and Sobradinho.

The continuation of this work is organized as follows: Section 2 presents the Echo State Network paradigm, while Section 3 exposes the new structure proposed by Boccato et al.; experimental results can be found in the Section 4 and, finally, Section 5 presents the conclusions and some possibilities for future work.

2 Echo State Networks

2.1 Classical Echo State Networks

We shall consider the standard ESN architecture depicted in Figure 1.

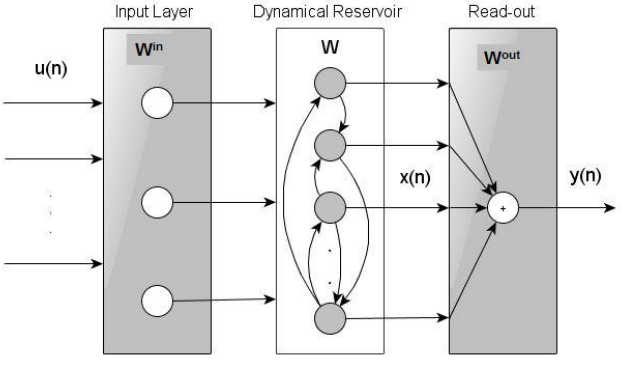


Fig. 1. Classical Echo State Network

The hidden layer, known as dynamical reservoir, is composed by fully interconnected nonlinear processing units, and is responsible for creating and maintaining in its state vector $\mathbf{x}(n)$ a kind of memory of the received input stimuli, which are represented by the input vector $\mathbf{u}(n) = [u(n), u(n-1), ...u(n-K+1)]^{\mathrm{T}}$. The network states are updated according to the following expression:

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W}^{\mathbf{in}}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n)) \tag{1}$$

where $\mathbf{W}^{\mathbf{in}} \in \Re^{NxK}$ and $\mathbf{W} \in \Re^{NxN}$ denote the weight matrices of the input and recurrent connections, respectively, and $\mathbf{f}(.) = (f_1(.), f_2(.), .., f_N(.))$ specifies the activation functions of the neurons within the reservoir. Finally, the network output vector $\mathbf{y}(n) = [y_1(n), y_2(n), ..., y_L(n)]^{\mathrm{T}}$ is given by:

$$\mathbf{y}(n+1) = \mathbf{f}^{\mathbf{out}}(\mathbf{W}^{\mathbf{out}}\mathbf{x}(n+1)) \tag{2}$$

where $\mathbf{f}^{\mathbf{out}}(.) = (f_1^{out}(.), f_2^{out}(.), ..., f_L^{out}(.))$ stands for the activation functions of all neurons in the output layer $\mathbf{W}^{\mathbf{out}} \in \Re^{LxN}$. Throughout this work, we will assume that these activation functions are simply identity functions.

In his pioneering work [8], Jaeger observed that, under certain hypotheses closely related to specific properties of the reservoir weight matrix **W**, the state vector $\mathbf{x}(n)$ tends to become asymptotically independent of the initial condition. In this case, the effect of the input history is patent, in the long run, within the dynamical reservoir and, as a consequence, the network is said to have echo states [8][9]. This property

allows the process of designing the reservoir of an echo state network to be performed separately and prior to the network training, so that the weights of the recurrent connections can remain fixed during this process. Hence, only the output weights are effectively trained with the aid of an error signal, and, due to the linear nature of the readout, this essentially amounts to employing a least-squares methodology.

With respect to the reservoir W design, the most common strategies are that proposed by Jaeger – to create a random sparse weight matrix $\mathbf{W^{Ja}} \in \Re^{NxN}$ according to a predefined distribution – and the more recent idea advocated by Ozturk et al. $\mathbf{W^{Oz}} \in \Re^{NxN}$ – to seek a uniform spread of the eigenvalues of over a circle with radius $R < 1$ [11]. These reservoir models are exemplified in (3) and (4), respectively:

$$w_{ij}^{Ja} = \begin{cases} +0.4 - .probability. = 0.025 \\ -0.4 - .probability. = 0.025 \\ 0 - .probability. = 0.95 \end{cases} \tag{3}$$

where w_{ij}^{Ja} represents each element of $\mathbf{W^{Ja}}$, $i,j = 1,\ldots,N$, and

$$\mathbf{W^{Oz}} = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & -R^N \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \tag{4}$$

Regardless of the adopted reservoir design method, the training process of the corresponding ESN consists of the following three main steps [9]: (*i*) create a reservoir weight matrix with spectral radius (i.e., the largest absolute eigenvalue) smaller than or equal to the unity; (*ii*) randomly define the input weight matrix $\mathbf{W^{in}}$ and (*iii*) determine the optimal coefficients of the linear combined at the output in the least-squares sense. This procedure established a simple and fast paradigm for recurrent neural network training, avoiding the original difficulties in the training process faced in the conventional RNN approach. These advantages are obtained at the expense of a "rigid" reservoir, which, however, does not necessarily leads to significant performance degradation when compared to a fully optimized RNN.

After this brief exposition of the foundations of ESNs, we present, in the following, the extension proposed by Boccato et al. [1] [2].

2.2 Echo State Networks Based on Volterra Filtering and Principal Component Analysis

The linear character of the ESN readout offers an advantage in terms of the training simplicity. Nevertheless, this structure cannot make use of the higher-order statistics

of the information coming from the reservoir dynamics. In order to overcome this limitation, Boccato et al. [1] [2] proposed the use of a nonlinear readout based on Volterra filters. Hence, each network output is determined according to the following expression:

$$y_i(n) = h_0 + \sum_{p=1}^{N} h_1(p)x_p(n) + \sum_{p=1}^{N}\sum_{q=1}^{N} h_2(p,q)x_p(n)x_q(n) + \sum_{p=1}^{N}\sum_{q=1}^{N}\sum_{r=1}^{N} h_3(p,q,r)x_p(n)x_q(n)x_r(n) + \ldots \quad (5)$$

where $x_k(n)$ is the k-th echo state in the instant n and $y_i(n)$ is the i-th ESN output, $h_i(.)$ are the coefficients of the filter (linearly related to the output) and N is the number of echo states. The proposal of Boccato et al. was also motivated by the fact that this structure is linear with respect to the free parameters, as we can observe in (5), which means that the training process still can be formulated in terms of a least-squares solution.

It is important to remark that a new problem arises in this proposal: an increase in the number of echo states causes the number of $h_i(.)$ coefficients to be adapted to grow rapidly. This computational burden has been mitigated through the application of a data compression methodology, more specifically, principal component analysis (PCA) [7], which reduces the number of effective signals that are transmitted to the readout.

The joint application of Volterra filtering and PCA is capable of providing the network with additional nonlinear mapping potential without violating the ESN main premises. This architecture has been analyzed in the context of channel equalization and blind separation of convolutive mixtures [1][2], and promising results were obtained. Recently, this proposal was preliminarily applied to the prediction of an important monthly seasonal streamflow series from the Furnas hydroelectric plant, and the obtained results were encouraging [13]. These efforts motivated this work, in which the ESNs presented in this section shall be applied to streamflow series with different hydrological behaviors, leading to an effective analysis of the performance of these proposals. Additionally, as a reference for performance comparison, we include in this repertoire feedforward multilayer perceptron (MLP) networks [6].

The next section discusses the series focused in this work, as well as their preprocessing, and presents the results obtained for their forecasting.

3 Seasonal Streamflow Series Forecasting

3.1 Monthly Seasonal Streamflow Series and Preprocessing

Monthly seasonal streamflow series are non-stationary in view of their seasonal components, which are determined by the rainfall periods in the vicinity of Brazilian rivers. These components can cause difficulties to the generation of an effective (either

linear or nonlinear) predictor [10][13]. In view of this fact, it is useful to apply a statistical technique to deseasonalize the series, removing these components and reinserting then at the end of the forecasting process. Equation (6) describes a procedure of this kind [10], which will be employed in this work:

$$z_{i,m} = \frac{s_{i,m} - \mu_m}{\sigma_m} \tag{6}$$

where the samples $s_{i,m}$ that form the original series $\mathbf{s}(n)$ are transformed into a new deseasonalized series $\mathbf{z}(n)$, with zero mean and standard deviation equal to one. The average μ_m and the standard deviation σ_m of each month m are estimated by:

$$\mu_m = \frac{1}{N_y} \sum_{i=1}^{N_y} s_{i,m} \tag{7}$$

$$\sigma_m = \sqrt{\frac{1}{N_y} \sum_{i=1}^{N_y} (s_{i,m} - \mu_m)^2} \tag{8}$$

where $s_{i,m}$ denotes the streamflow in the year i=1,2,…,N_y in the month m = 1,2,…,12.

Hence, this work will perform the forecasting of the series $\mathbf{z}(n)$, separating the training and test samples; afterwards, the seasonal component is reinserted to propitiate a comparative performance analysis.

3.2 Computational Results

The case studies presented in this section were built from three different representative examples taken from the historical monthly seasonal streamflow series from Brazilian plants: Emborcação, Furnas and Sobradinho. The corresponding data are available at the website of the Brazilian Electric System National Operator (ONS) [14]. The considered series range from 1931 to 1990. The test sets are defined to cover three periods with different degrees of precipitation and water volume: Emborcação from 1952 to 1956, Furnas from 1952 to 1956, Sobradinho from 1952 to 1956 and Sobradinho from 1972 to 1976.

All periods comprise 5 years and 60 samples, and are often used in this kind of investigation [13]. These scenarios can be considered as representative choices to test the performance and robustness of the predictors, because of the variability of their characteristics. The training set was composed of all the available samples except those associated with the test periods.

In addition to the ESNs, an MLP trained with the Scaled Conjugated Gradient Method (SCGM) [5] was used in the forecasting process. Cross validation was used to avoid overtraining. The choice for a feedforward structure trained with an effective optimization method [6] allows the obtained results to be placed in a clearer light [4].

The first step in the forecasting process was the application of the deseasonalization process described in the previous section. Afterwards, the sets had their average subtracted from them, and the MLP and the ESNs were trained. All networks had only two input delays ($K = 2$), a choice that can be considered parsimonious and that poses a challenging problem to the predictors. The forecasting was always carried out aiming at one step ahead and the networks yield a single output ($L = 1$). Another important remark is that, following the guidelines found in [1] and [2], only the first- and third-order terms of the Volterra filter were utilized. The choice of the reservoir parameters followed the canonical approaches of Jaeger [8] and Ozturk at al. [11], being, in the latter case, the spectral radius set to 0.8. After a number of preliminary tests, the number of principal components in the proposal of Boccato et al. [2] was set to $N_{PC} = 2$, and the number of neurons (N) in each case was also chosen. For the MLP, a cross validation technique was applied to define the training-stopping criterion.

Tables 1 to 4 display the performance of the employed neural networks in terms of mean squared error (MSE) and mean absolute error (MAE), described on Equations (9) and (10). The presented results represent an average over 20 simulations. The following labels were used: "Boc.+J." for the ESN proposed by Boccato et al. with Jaeger's reservoir and "Boc.+O." for the ESN proposed by Boccato et al. with the Ozturk et al. reservoir.

$$MSE = \frac{1}{N_s}\sum_{t=1}^{N_s}(d(n) - y(n))^2 \tag{9}$$

$$MAE = \frac{1}{N_s}\sum_{t=1}^{Ns}\left|d(n) - y(n)\right| \tag{10}$$

where $d(n)$ is the observed data, $y(n)$ is the output of the network and N_s the number of samples.

Table 1. Mean square error and mean absolute error for series EMBORCAÇÃO

	N	MSE(e+04)	MAE	MSE desea	MAE desea
MLP	10	7.3182	163.95	0.9910	0.7554
Jaeger	80	7.3587	159.62	1.0120	0.7400
Ozturk	120	7.4540	169.78	1.1233	0.8120
Boc.+J.	60	7.2512	162.01	0.9762	0.7455
Boc.+O.	110	7.0952	162.90	0.9841	0.7545

Table 2. Mean square error and mean absolute error for series FURNAS

	N	MSE(e+04)	MAE	MSE desea	MAE desea
MLP	10	6.7171	178.05	0.2882	0.4495
Jaeger	15	7.5076	191.06	0.3155	0.4600
Ozturk	20	7.2448	176.75	0.2736	0.4062
Boc.+J.	10	5.3837	165.54	0.2515	0.4235
Boc.+O.	120	5.7795	166.94	0.2538	0.4197

Table 3. Mean square error and mean absolute error for series SOBRADINHO 1952/1956

	N	MSE(e+04)	MAE	MSE desea	MAE desea
MLP	12	24.03	620.24	0.4931	10.2578
Jaeger	30	17.32	624.02	0.4611	10.4370
Ozturk	3	17.78	632.63	0.4715	10.9520
Boc.+J.	20	17.83	606.88	0.4730	10.2202
Boc.+O.	40	17.02	598.36	0.4479	10.0466

Table 4. Mean square error and mean absolute error for series SOBRADINHO 1972/1976

	N	MSE(e+04)	MAE	MSE desea	MAE desea
MLP	9	19.69	509.20	0.4551	6.1088
Jaeger	30	15.88	483.08	0.4317	5.7588
Ozturk	20	16.57	513.56	0.4694	6.1163
Boc.+J.	120	16.30	490.98	0.4438	5.8471
Boc.+O.	120	15.52	477.48	0.4420	5.6350

Finally, the ANOVA Friedman's test [10] was used to check whether the approaches actually provided different results. The *p-value* achieved was 1.71588e-10 for the Emborcação series and it was equal to zero in the other cases. This shows that the prediction performances are, indeed, different or, in other words, that the approach applied in the prediction process directly affected the overall results.

Analyzing the results summarized in Tables 1 to 4, it is possible to notice that, for both the real and the deseasonalized domain, the Volterra-based ESN achieved, in most cases, the best results, showing that a nonlinear output layer allows a more thorough use of the information coming from the reservoir. If we consider that the neural networks aim at minimizing the mean-squared error in the training process, the proposal by Boccato et al. achieved the best results in all but one case. The two methods for designing the dynamical reservoir led to similar results, both being applicable to the task at hand.

Another observation refers to the number of artificial neurons used by each architecture. The dynamical reservoir in the ESNs needs to be as variable as possible, and a large number of neurons do not compromise the fast convergence of the training process, which is based on linear combination. On the other hand, the MLP can become over trained with a large number of neurons - losing its ability to generalize – and, moreover, its training process tends to be slower.

It is possible to state that this work reveals the benefits obtained with a more flexible output layer, although ESNs in general can be considered as promising alternatives to cope with monthly seasonal streamflow series forecasting. Moreover, the computational complexity and the processing time were quite attractive as there are no derivatives to be calculated.

To conclude the analysis, in Figures 2 to 5, we present the test sets, in real and deseasonalized domains, associated with the best results.

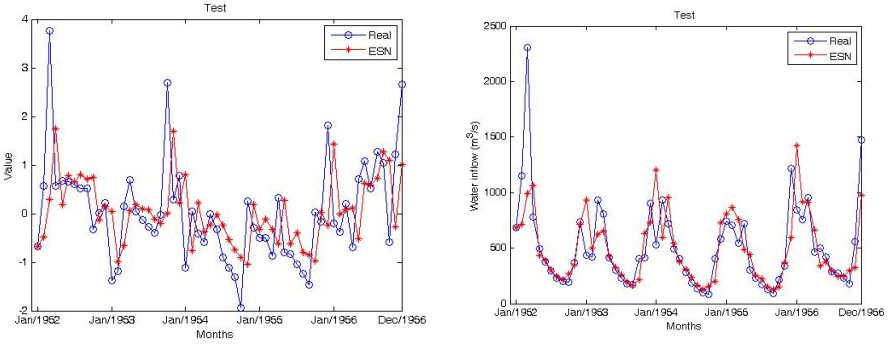


Fig. 2. Best performance for series EMBORCAÇÃO in deseasonalized and real spaces

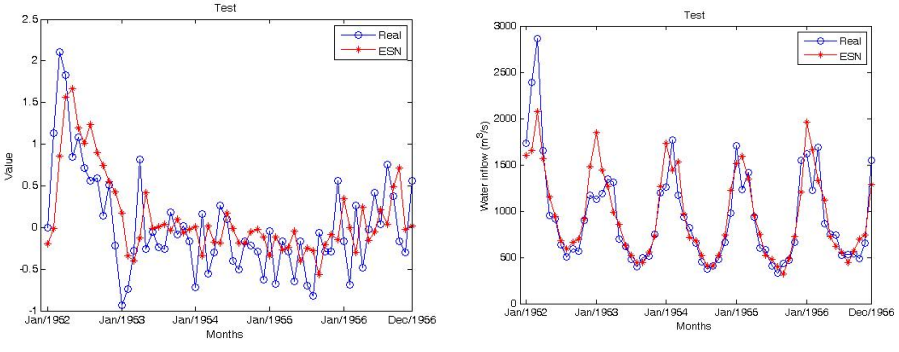


Fig. 3. Best performance for series FURNAS in deseasonalized and real spaces

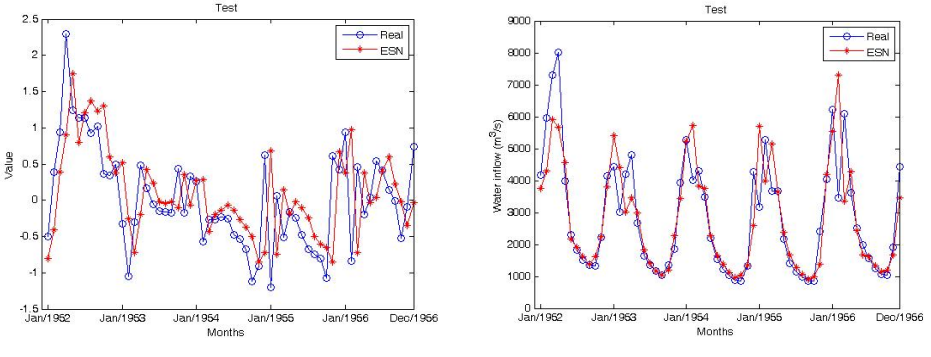


Fig. 4. Best performance for series SOBRADINHO 1952/56 in deseasonalized and real spaces

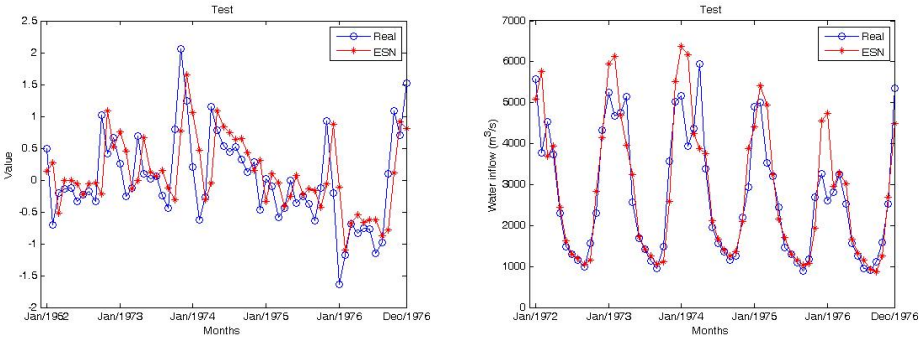


Fig. 5. Best performance for series SOBRADINHO 1972/76 in deseasonalized and real spaces

4 Conclusions

This work presented a comparison of the performances of different architectures of echo state networks (ESNs) – together with an MLP - in the monthly seasonal streamflow forecasting of series from important Brazilian hydroelectric plants. This problem is a very significant one from the standpoint of electric planning and power generation. The studied ESNs were based on reservoir proposals by Jaeger and Ozturk et al. and on the approach introduced by Boccato et al., which is characterized by the use of a Volterra filter in the output layer combined with a stage based on principal component analysis to compress information coming from the dynamical reservoir.

The motivation to use the ESN comes from their intrinsic attractive features, such as the simplicity on implementation and operation allied to a fast and efficient training process, based on linear regression. Consequently, the computational complexity and memory requirements of the ESNs, compared with those of classical recurrent neural approaches, are significantly smaller. The new proposal of Boccato et al. [1] [2] maintains the most important characteristic of this network, i.e., the simplicity of the training process.

Within the framework of the forecasting of three different series from distinct hydroelectric plants – Emborcação, Furnas and Sobradinho – aiming at one step ahead, it was possible to verify that the proposal of Boccato et al. achieved the best performance in both the deseasonalized and real domains, in most cases. Besides, the combination of Volterra filtering and PCA allowed a further exploration of the variability of the echo states, as a higher number of artificial neurons can be used without meeting the "curse of dimensionality".

These observations show that ESNs, especially the proposal with a nonlinear output layer, are good alternatives to this forecasting problem, deserving to be considered viable alternatives to deal with problems related to hydrological series.

Among the perspectives for future work, it is possible to highlight the use of different monthly seasonal streamflow series, tests employing forecasting with multiple steps ahead and a more detailed analysis of the impact of the pre-processing methodology.

Acknowledgment. This work was supported by grants from CAPES, CNPq and FAPESP.

References

1. Boccato, L., Lopes, A., Attux, R., Von Zuben, F.J.: An Echo State Network Architecture Based on Volterra Filtering and PCA With Application to the Channel Equalization Problem. In: IEEE Proceedings of International Joint Conference on Neural Networks, San Jose - CA, USA (2011)
2. Boccato, L., Lopes, A., Attux, R., Von Zuben, F.J.: An Extended Echo State Network Using Volterra Filtering and Principal Component Analysis. Neural Networks (2012) (available online February 16, 2012), doi: 10.1016/j.neunet.2012.02.028
3. Box, G., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis, Forecasting and Control, 3rd edn., Holden Day, Oakland, California, EUA (1994)
4. Crone, S.F., Hibon, M., Nikolopoulos, K.: Advances in Forecasting With Neural Networks? Empirical Evidence from the NN3 Competition on Time Series Prediction. International Journal of Forecasting 27(3), 635–660 (2011)
5. Dos Santos, E.P., Von Zuben, F.J.: Improved Second-Order Training Algorithms for Globally and Partially Recurrent Neural Networks. In: IEEE International Joint Conference on Neural Networks (IJCNN 1999), Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 1999), Washington-USA, vol. 3, pp. 1501–1506 (1999)
6. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice-Hall (1999)
7. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons, New York (2001)
8. Jaeger, H.: The Echo State Approach to Analyzing and Training Recurrent Neural Networks, Bremem: German National Research Center for Information Technology. Tech. Rep. GMD Report 148 (2001)
9. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review 3(3), 127–149 (2009)
10. Luna, I., Ballini, R.: Top-Down Strategies Based on Adaptive Fuzzy Rule-Based Systems for Daily Time Series Forecasting. International Journal of Forecasting, 1–17 (2011)
11. Ozturk, M.C., Xu, D., Principe, J.C.: Analysis and Design of Echo State Networks. Neural Computation 19, 111–138 (2007)
12. Sacchi, R., Ozturk, M.C., Príncipe, J.C., Carneiro, A.A.F.M., da Silva, I.N.: Water Inflow Forecasting Using the Echo State Network: a Brazilian Case Study. In: IEEE Proceedings of International Joint Conference on Neural Network, Orlando FL, USA (2007)
13. Siqueira, H.V., Boccato, L., Attux, R., Lyra Filho, C.: Seasonal Streamflow Series Forecasting Using Echo State Networks. In: 10th Brazilian Congress on Computational Intelligence, Fortaleza-CE, Brazil (2011) (in Portuguese) (Previsão de séries de vazões com redes neurais de estados de eco)
14. ONS - Electric System National Operator - Brazil (online), `http://www.ons.org.br/operacao/vazoes_naturais.aspx`

Face Segmentation Using Projection Pursuit for Texture Classification

Victor R.S. Laboreiro, J.E.B. Maia, and Thelmo P. de Araujo

Universidade Estadual do Ceará – UECE
Av. Paranjana, 1700 – Fortaleza – CE – Brazil – 60740-903
{victor.laboreiro,thelmo.dearaujo}@gmail.com, jose.maia@uece.br

Abstract. Frontal face images are segmented into 7 regions using only sum and difference histograms as pixel information, without any a priori knowledge. In the training phase, a decision tree is created using a projection pursuit algorithm: in each step, the optimal one-dimensional projection is chosen by a simulated annealing process according to a projection index, and classes are isolated by a decision boundary that maximizes class separability, until the end nodes contain only one class each. Satisfactory qualitative and quantitative results were obtained and presented.

Keywords: face segmentation, exploratory projection pursuit, decision trees, simulated annealing.

1 Introduction

Image segmentation is one of the most studied problems in computer vision [22], and consists of dividing an image into its component regions or objects [7]. Being the image as complex as a human face, segmentation becomes a very difficult task [7], specially when gray-scale images are considered.

In face segmentation one usually looks for the extraction of facial features such as eyes, mouth, nose, etc. We adopt a bottom-up approach, but image histograms are analyzed instead of pixel intensities. Introduced by Unser [25], sum and difference histograms count the frequency of the sum (or the difference) of pixel intensities in a window for each pair of pixels separated by d_1 rows and d_2 columns.

Sum and difference histograms shall be used to train a supervised learning process in order to identify regions of interest in the facial image for posterior classification of test images. Learning methods that use images for training deal with huge dimensions. A way to reduce the number of dimensions involved is projecting the data into lower dimension spaces. PCA, LDA, and ICA are widely used methods for dimensionality reduction [28].

Together with dimensionality reduction, classification may be achieved by means of projections: if a projection helps to maximize class separability, the classification process will be easier. LDA is again a widely used method of classification via projections [22,28,13].

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 237–244, 2012.

However, PCA and LDA are computationally expensive. On the contrary, random projections are shown to be much less expensive. Bingham and Mannila [3] use Johnson-Lindenstrauss theorem [10] to support the effectiveness of random projections in dimensionality reduction. This method generates random projections and chooses the best one(s) by some criterium. In this paper a *projection index* evaluates each random projection by measuring the ratio of between-class and within-class variances [13].

Our work aims to segment facial images into seven classes, viz., eyes, nose, mouth, eyebrows, hair, facial skin, and background. To achieve such segmentation via exploratory projection pursuit, a decision tree is constructed using simulated annealing [12,13], which searches for optimal one-dimensional projections. Once the projection is chosen, the algorithm computes a decision boundary to separate classes in some optimal sense. New nodes are created in the decision tree until each end node contains only one class.

The remainder of the paper is organized as follows: Section 2 presents some related works. Methodology is described in Section 3. Data, results, and discussion are in Section 4. Section 5 concludes the paper.

2 Related Works

Methods for segmentation of facial images may be roughly divided into two types: the ones that partition the image into two classes—e.g., face and non-face—and are common in real-time applications [14]; and the ones that aim for multi-class segmentation. However, multi-class segmentation methods usually make use of a priori knowledge of facial features, e.g., Shylaja et al. [21] extract eyes and lips from face images using a marker based watershed method. Tao et al. [23] use skin color and facial geometry information to extract features such as eyes, mouth, cheeks, and chin.

In this work we propose not to use a priori knowledge of facial features to extract the seven above mentioned features, although post-processing is necessary for accuracy sake.

Liu and Fieguth [15] propose texture classification based on random projections and claim significant improvements over four state-of-the-art texture classification methods. Velloso et al. [26] use and compare two neural PCA and two neural ICA methods for dimensionality reduction in order to apply local pattern spectra to gray-scale images. Local spectra histograms are used by Liu and Wang [16] to provide statistics for texture and non texture regions in real and in synthetic images.

Since feature extraction is the most time consuming part of most segmentation methods [20], we consider exploratory data analysis, introduced by Tukey [24], because it maximizes data insights, extracts important variables, and detects outliers [17].

As for dimensionality reduction, projection pursuit is shown to perform well when a priori knowledge of data structure is few [5,6]. Bingham and Mannila [3] have shown that random projections is much less computationally expensive

than, for example, PCA methods, on the order of $O(dkN)$, where d is the original dimension, k the desired dimension, and N the number of samples.

In order to optimize projections, a variety of projection indexes are used, among them, moment index [11], entropy index [9], Legendre index [5], Hermite index [8], and a modified LDA index [13]. We adopted the latter in our algorithm.

3 Methodology

Our methodology for facial image segmentation, to sum up, uses a supervised classification of image pixels in which sets of sample images are used to train the classifier. For each pixel, sum and difference histograms [25] are the features computed in a small window around it.

The classifier is a decision tree whose end nodes indicate the classes. Decision tree nodes are created based on one-dimensional projections of the feature vectors: simulated annealing finds the optimal projection guided by a projection index. Once the optimal projection is found, a decision boundary splits the projected vector into two parts, and each part spans a new tree node. The process continues until the end nodes contain only one class each.

A post-processing algorithm is applied to reduce noise. The steps are described in details in the following subsections.

3.1 Feature Extraction

Figure 1 summarizes the preprocessing and pixel feature extraction phase. The size of the original images is reduced in two steps: in the first one, background is reduced using Viola-Jones algorithm [27] available on the OpenCV library [1]. In the second step, image size is further reduced to 25% of its original size with nearest-neighbor interpolation [7].

Next, the reduced color images are converted to gray scale, and then the number of gray levels are reduced from 256 to 64 using histogram equalization [7]. This reduces data dimensions and lighting variations.

Sum and difference histograms are computed for each pixel in the preprocessed images as described by Unser [25]:

$$h_s(i, d_1, d_2) = |\{(k, l) \in D,\, s_{k,l} = i\}|$$
$$h_d(i, d_1, d_2) = |\{(k, l) \in D,\, d_{k,l} = i\}|\,,$$

where

$$s_{k,l} = y_{k,l} + y_{k+d_1,l+d_2} \quad \text{and} \quad d_{k,l} = y_{k,l} - y_{k+d_1,l+d_2}\,,$$

(k, l) is the pixel coordinates inside window D, d_1 and d_2 are distances in pixels, and i and $y_{\cdot,\cdot}$ are pixel intensities. Histograms are further normalized: h_s ranges from 0 to 126 and h_d from -63 to 63.

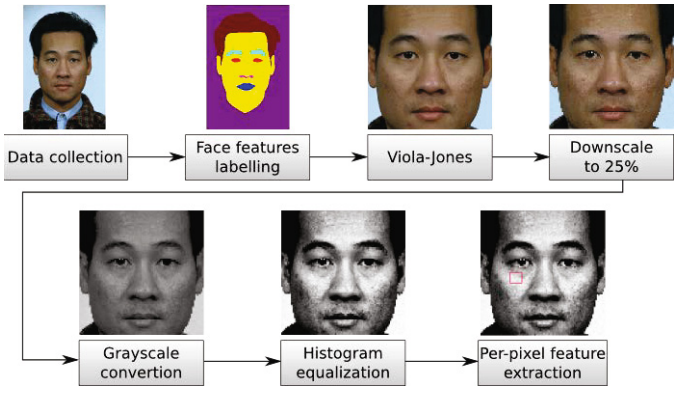


Fig. 1. Overview of the proposed methodology. Stages included in the diagram are data gathering, image preprocessing and pixel feature extraction.

3.2 Decision Trees

Friedman and Tukey [6] suggest an approach of class isolation for the training phase of the classification algorithm. In [13] class isolation is performed by constructing a decision tree using projection pursuit.

In order to project M-dimensional feature data into a (lower) m-dimensional feature space, an $m \times M$ matrix is used. The classifier is a binary decision tree, which has a projection matrix (in our case, $m = 1$, a row vector) and a decision boundary value in each node, making this process different from feature selection.

Using as inputs the histograms data, the labels, and the simulated annealing parameters, a tree node is created. If there is only one class, this is a leaf node. If not, the simulated annealing is ran to produce an optimal projection, guided by a projection index, defined below. This optimal projection is store in the node and used to project the data.

The projected vector is used to compute a vector with the ordered values of the means for each class. The decision boundary is defined as the middle point of the greatest difference of two sequential elements in this ordered vector, thus maximizing class separability. All projected data is scanned and the points lesser than the decision boundary form a (new) left node in the tree. The ones greater than the decision boundary form a right node. The algorithm runs until there is only one class in each end node.

Following Lee et al. [13], an L_p-like norm is used as projection index for optimality decisions in the simulated annealing:

$$L_r(A) = \left[\frac{\sum_{i=1}^{g} |\overline{x}_i - \overline{\overline{x}}|^r}{\sum_{i=1}^{g} \sum_{j=1}^{n_i} |x_{ij} - \overline{x}_i|^r} \right]^{\frac{1}{r}},$$

where A is the projected data; x_{ij} is the j-th sample of class i; $\overline{x}_i$ is the average for class i; $\overline{\overline{x}}$ the average for all class; g is the number of classes; and n_i is the number of samples in class i.

To avoid the outliers increase the tree depth unreasonably and to prevent over-fitting, a small modification is introduced in the classification algorithm: instead of checking the position of each projected data point relatively to the decision boundary, we consider all projected points in a class whose mean is lesser than the decision boundary value to belong to the new left node. And similarly to the right node. Algorithm 1 summarizes this training phase.

Algorithm 1. Decision tree training

Input: Features and labels data sets
Output: The constructed decision tree

1. A node is created.
2. If the input data has only one class, then mark the current node class as this unique class value. Else, do steps 3-5.
3. Projection pursuit is performed using simulated annealing to look for the best one-dimensional projection evaluated by the L_r projection index.
4. Data is transformed using the obtained projection and a decision boundary is defined separating projected data into two clusters.
5. Projected data is used to compute the mean per class and, therefore, all high-dimensional samples are separated as input to recursively construct left and right children of the current node, depending, respectively, whether the sample class mean is lesser than or greater than the decision boundary.

3.3 Post-processing

Once the training phase is finished, all images are classified using the decision tree previously constructed, and a region-growing algorithm [18] is used to perform post-processing on the images classified by the decision tree.

A loop iterates for each pixel on the input image by first checking whether the current position have been already visited and, this not being the case, the four-connected regions flood-fill algorithm is executed using as inputs the image to be processed, the current pixel position, and the current pixel label.

The returned list of pixels belonging to the flooded region is used to track which pixels have been already visited and the number of elements of this returned list is used to check if the number of pixels in the region is lesser than an input threshold value (in this work, 35 pixels). If this happens, the list of labels from neighboring pixels, which is also returned by the flood-fill algorithm, is used to find the most frequent label. Post-processing is summarized in Algorithm 2.

4 Data, Results, and Discussion

Experiments were performed using randomly selected frontal face color images from the FERET database [19]. The selected images were then manually labeled (second column images on Figure 2) using the open-source software GIMP [2] into seven classes: eyes, eyebrows, hair, skin, nose, mouth and background. Since

Algorithm 2. Flood-fill based post-processing

Input: A segmented image and a threshold value
Output: The post-processed image
for *each pixel* (i, j) *on image* **do**
  if $visited\_pixels[i, j] = false$ **then**
    empty $pixels\_list$ and $neighbors\_list$
    $flood\_fill(image, i, j, image[i, j], list\_pixels, neighbors\_list)$
    Mark all positions on $list\_pixels$ as visited on the $visited\_pixels$ matrix
    if *number of elements of* $list\_pixels < threshold$ **then**
      paint all pixels in $list\_pixels$ by the mode of $neighbors\_list$
    end if
  end if
end for

the manual labeling process (which is part of the training phase) is very time consuming, only 40 images were selected to perform the tests.

In this work we adopted the following parameters: For the simulated annealing, initial temperature $T_0 = 0.1$ and cooling factor $c = 0.999$. At the feature extraction step, an 8×8 window is used to compute sum and difference histograms. Each feature vector is composed by 4 sum and 4 difference histograms computed for angles $0\,°, 45\,°, 90\,°$, and $135\,°$, and pixel distance $d = 2$.

A total of 34 experiments were performed, each one consisting of training the decision tree using 10 randomly selected images with the projection pursuit algorithm, classifying all the 40 images and counting the amount of correctly classified pixels for each image.

Results in Table 1 show the mean and standard deviation percent classification accuracy per image. The total accuracy average and standard deviation obtained is, respectively, 77.24% and 6.27%. The results are, therefore, superior compared with the ones presented in [4] which obtained 63.24% as total mean and 15.18% as total deviation.

Figure 2 shows typical qualitative results: all regions of interest are accurately located, although with noisy boundaries, due to the low quantity of segmentation features of gray-scale facial images (a curse not shared by some medical images).

Table 1. Mean and std. deviation percent ratio of correctly classified pixels by image

	Img_01	Img_02	Img_03	Img_04	Img_05	Img_06	Img_07	Img_08	Img_09	Img_10
Mean	81.70	77.31	74.09	79.74	80.20	78.43	76.84	81.72	83.58	83.93
Standard Deviation	2.43	3.46	3.18	2.89	3.81	3.76	4.38	2.85	4.06	2.68
	Img_11	Img_12	Img_13	Img_14	Img_15	Img_16	Img_17	Img_18	Img_19	Img_20
Mean	83.36	77.22	80.94	67.42	72.71	79.56	79.72	79.75	83.25	83.07
Standard Deviation	3.23	3.93	2.77	3.25	2.89	3.60	2.94	2.83	3.04	2.61
	Img_21	Img_22	Img_23	Img_24	Img_25	Img_26	Img_27	Img_28	Img_29	Img_30
Mean	82.88	73.62	82.00	83.91	76.03	70.85	81.57	77.24	73.43	76
Standard Deviation	3.62	2.40	2.80	2.55	3.55	3.51	2.86	3.23	2.27	3.83
	Img_31	Img_32	Img_33	Img_34	Img_35	Img_36	Img_37	Img_38	Img_39	Img_40
Mean	78.91	73.04	77.47	71.47	76.27	73.13	58.26	72.75	76.99	69.25
Standard Deviation	3.08	4.90	2.71	4.26	4.62	3.29	4.22	4.97	3.37	3.79

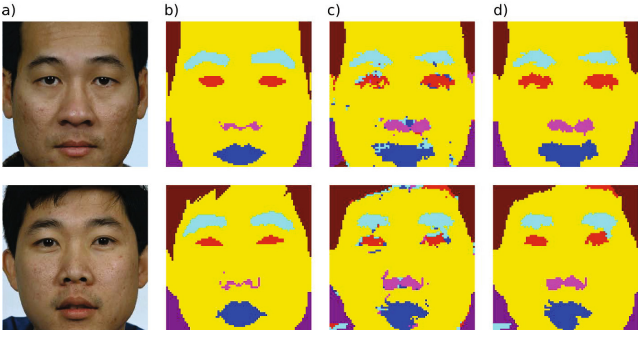


Fig. 2. (a): Original images after being cropped by Viola-Jones face detector. **(b)**: Target labeled images. Classification results of the proposed method before **(c)** and after **(d)** post-processing.

5 Conclusions

This work presented the results of segmentation experiments on frontal facial images based on pixel classification. Texture features (sum and difference histograms) were used to segment seven interest regions in the human face. Qualitative and quantitative performance results for the used algorithm were presented and considered satisfactory, encouraging us to proceed further on this line of reasoning.

We are currently working on improving the algorithm, mainly in three directions: first, by adding new features (such as Gabor wavelets) and re-selecting them. Second, by testing other classifiers, such as RBFNN. And, finally, by improving the post-processing algorithm using morphological processing techniques.

References

1. Open Computer Vision Library (April 2012), http://sourceforge.net/projects/opencvlibrary/
2. The GNU Image Manipulation Program (April 2012), http://www.gimp.org/
3. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 245–250. ACM (2001)
4. Chrisóstomo, H.B., Laboreiro, V.R.S., Araujo, T.P.d., Maia, J.E.B.: Face segmentation based on texture classification and heuristic. In: X Congresso Brasileiro de Inteligência Computacional (CBIC) (November 2011)
5. Friedman, J.H.: Exploratory projection pursuit. Journal of the American Statistical Association, 249–266 (1987)
6. Friedman, J.H., Tukey, J.W.: A projection pursuit algorithm for exploratory data analysis. IEEE Transactions on Computers 100(9), 881–890 (1974)

7. Gonzalez, R.C., Woods, R.E.: Processamento Digital de Imagens, 3rd edn. Prentice Hall - Pearson, São Paulo (2009); Trad. C. Yamagami e L. Piamonte
8. Hall, P.: On polynomial-based projection indices for exploratory projection pursuit. The Annals of Statistics 17(2), 589–605 (1989)
9. Huber, P.J.: Projection pursuit. The Annals of Statistics, 435–475 (1985)
10. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. Contemporary Mathematics 26(189-206), 1 (1984)
11. Jones, M.C., Sibson, R.: What is projection pursuit? Journal of the Royal Statistical Society. Series A (General), pp. 1–37 (1987)
12. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220(4598), 671 (1983)
13. Lee, E.K., Cook, D., Klinke, S., Lumley, T.: Projection pursuit for exploratory supervised classification. Journal of Computational and Graphical Statistics 14(4), 831–846 (2005)
14. Li, H., Ngan, K.N., Liu, Q.: FaceSeg: automatic face segmentation for real-time video. IEEE Transactions on Multimedia 11(1), 77–88 (2009)
15. Liu, L., Fieguth, P.: Texture classification from random features. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(3), 574–586 (2012)
16. Liu, X., Wang, D.L.: Image and texture segmentation using local spectral histograms. IEEE Transactions on Image Processing 15(10), 3066–3077 (2006)
17. Natrella, M.: NIST/SEMATECH e-Handbook of Statistical Methods (2010), http://www.itl.nist.gov/div898/handbook/ (accessed: March 23, 2012)
18. Pakhira, M.K.: Computer Graphics, Multimedia, and Animation. PHI Learning Pvt. Ltd., New Delhi (2008)
19. Phillips, P.J.: The facial recognition technology FERET database. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2004)
20. Randen, T., Husoy, J.H.: Filtering for texture classification: A comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(4), 291–310 (1999)
21. Shylaja, S.S., Balasubramanya Murthy, K.N., Natarajan, S., Prasad, A., Modi, A., Harlalka, S.: Feature extraction using marker based watershed segmentation on the human face. In: 2012 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–5 (January 2012) (to appear)
22. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer, London (2011)
23. Tao, C., Shanxua, D., Fangrui, L., Ting, R.: Face and facial feature localization based on color segmentation and symmetry transform. In: International Conference on Multimedia Information Networking and Security, MINES 2009, vol. 2, pp. 185–189. IEEE (2009)
24. Tukey, J.W.: Exploratory Data Analysis. Addison Wesley, Reading (1977)
25. Unser, M.: Sum and difference histograms for texture classification. IEEE Transactions on Pattern Analysis and Machine Intelligence (1), 118–125 (1986)
26. Velloso, M.L.F., Carneiro, T.A.A., Souza, F.J.D.: Pattern spectra for texture segmentation of gray-scale images. In: 7th International Conference on Intelligent Systems Design and Applications, ISDA 2007, pp. 347–352. IEEE (2007)
27. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, p. I–511. IEEE (2001)
28. Zhao, W., Chellappa, R. (eds.): Face Processing: Advanced Modeling and Methods. Academic Press, San Diego (2006)

Visual Data Mining for Identification of Patterns and Outliers in Weather Stations' Data

José Roberto M. Garcia, Antônio Miguel V. Monteiro, and Rafael D.C. Santos

Brazilian National Institute for Space Research,
Av dos Astronautas, 1.758, Jd. Granja - CEP 12227-010,
São José dos Campos – São Paulo – Brasil

Abstract. Quality control of climate data obtained from weather stations is essential to ensure reliability of research and services based on this data. One way to perform this control is to compare data received from one station with data from other stations which somehow are expected to show similar behavior. The purpose of this work is to evaluate some visual data mining techniques to identify groupings (and outliers of these groupings) of weather stations using historical precipitation data in a specific time interval. We present and discuss the techniques' details, variants, results and applicability on this type of problem.

Keywords: Visual data mining, clustering, self-organizing map, fuzzy C-means.

1 Introduction

Observational data obtained from weather stations is important due to its use on generating weather and climate numeric predictions, evaluating models results and making climatic research [1], so having reliable data is an essential issue to make reliable research and applications. However, the data is not completely reliable: some weather stations are still human operated, which often are subject to reporting errors; and even the automatic ones depend on hardware and network communication which can pollute the data [2]. A quality control system is clearly required to verify the data's quality.

At the Brazilian National Institute for Space Research's (INPE) CPTEC (Brazilian National Center for Weather Prediction and Climate Studies) there is a 3-level quality control system for weather stations data. The first approach verifies whether the data is inside upper and lower limits to the variable; the second uses arbitrary geographic rectangular regions and limits on the variables on these regions, and the third one uses limits for each variable and specific weather station. These controls aims to reject spurious data and classify suspicious data [3].

The problem with these approaches is that the bounds used (for the variable and geographic regions) are not natural and can filter important data from datasets for analysis. Moreover, the number of rejections increases the work of the data administrator that need to analyze, one by one, all the rejected data.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 245–252, 2012.

Clearly some other methods that help meteorologists interpret the vast amount of data in search for potential suspicious data are required. This work presents some possible algorithms and implementations based on visual data mining to accomplish this task.

This paper is divided in the following sections: Section 2 presents some visual data mining concepts relevant to this work. Section 3 presents the data used in this study and its relevant features. Sections 4 and 5 presents two of the algorithms that were selected for implementation of the visual data mining tasks. Section 6 presents some conclusions and directions for future work.

2 Visual Data Mining

Visual data mining can be defined as the set of techniques and approaches used to extract and understand the information encoded in data sets using the human visual perception system as part of the data processing task [4,5]. Visual data mining may help uncover or highlight data features, may make the understanding of the features easier and speedier and may be used to cross-validate conclusions obtained through other methods [6,7].

The difference between visual data mining and traditional data visualization is somehow blurred: for our purposes we consider that visual data mining tasks involve the processing of the data with one or more data mining algorithms and that visualization is done over the original data together with information obtained from those algorithms. Visual data mining may also be considered one of the components of exploratory data analysis (EDA) and strongly related to visual analytics [8,9].

Visual data mining tools and approaches are used in several knowledge domains, including analysis of environmental, geophysical and atmospheric data [6,10,11], which are relevant to our interests.

3 Data

The data we want to mine and visualize is inherently spatio-temporal: time series collected from sensors with geographical coordinates associated. The data was selected from a database containing daily precipitation data for all Brazil, with a total of 115 million records and with some weather stations having more than 100 years of recorded data.

In order to evaluate the visual data mining techniques we've selected only data from stations on the state of São Paulo, and created for each station a time series containing the monthly accumulated precipitation. Only stations that could yield at least 25 readings in a month were considered. The final database contained data from 1.341 weather stations (including geographic coordinates and altitude) with a time series with 84 entries corresponding to monthly accumulated precipitation. The data for each weather station covers the same period in time.

Several visualization techniques can be used to get some basic information about the behavior of this type of data. The most frequently used are time series plots [10] or parallel coordinates plots [12] (with each horizontal axis mapped to a time coordinate). Figure 1 shows a plot of all data from the 1.341 weather stations. From Figure 1 we can see monthly and global extrema and get a feeling

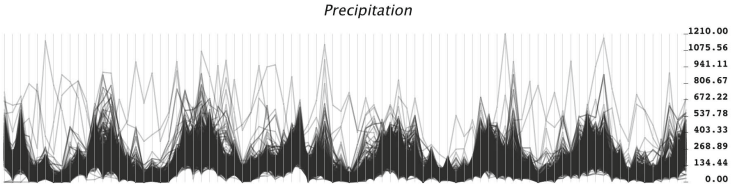


Fig. 1. Time series plot of the data used in this work

of the behavior of the whole set of time series: there are periods with more and less accumulated precipitation which are more or less correspondent to the wet and dry seasons. At the same time we can observe that there isn't a clear global maxima or minima, and visual identification of which station is providing data outside of a range is not trivial. Although this kind of plot provides some general information about the series it does not conveys any information of behavior similarity (or anomalies) between stations – in other words we cannot infer whether two or more stations have similar or different behaviors nor identify geographically close weather stations – this is important because we expect that weather patterns have a geographic extent and that would influence data from stations that are close to this pattern.

Since the data is inherently geographic (stations' coordinates are points in space), it is natural to visualize them overlaid in a map. The problem is that the data associated to each point in the map is a multidimensional time series – in order to visually identify behavior similarity we would need to reduce the time dimension or extract fewer features from it so it can also be plotted in the map and used for visual comparison with other points in the map.

In this work we investigate two approaches to extract features from the time series that can be used in for visual data mining: the first is based on a common clustering algorithm and the other on a well-known dimensionality reduction algorithm. These techniques and results are presented in Sections 4 and 5.

4 Fuzzy C-Means Clustering

Fuzzy C-Means [13,14] is an iterative algorithm that attempts to minimize an objective function J defined as:

$$J = \sum_{k=1}^{n} \sum_{i=1}^{c} \mu_{ik}^{m} |x_k - v_i|^2 \ , \quad m > 1 \tag{1}$$

Where x_k is the k-th data vector, v_i is the i-th cluster center vector, c is the number of clusters, n is the number of points in the data, μ is the membership values' table or matrix which contains the membership values for all points in all clusters; which indicates to which degree or extent the data vector x_k belongs to the cluster v_i, and m is a fuzziness value. The membership values are subject to the conditions $0 \leq u_{ik} \leq 1$ and $\sum_{i=1}^{c} u_{ik} = 1$ for all k.

One problem with the Fuzzy C-Means algorithm is the definition of its parameters. In particular two parameters are often defined experimentally: the number of clusters C and the fuzziness factor m. Of those, determination of a suitable m is relatively easy: when m is close to 1 the algorithm behaves like the non-fuzzy K-Means; while when m is large enough all data may have equal membership in all clusters. For some applications empirical values of m between 1.5 and 2.5 are suggested [15,16].

C, the number of clusters, is often empirically determined, although some metrics of cluster validity may be employed to find a suitable value for C. Three of those metrics are the partition coefficient, the partition entropy and the compactness and separation metrics [14]. These metrics are calculated after the data is clustered with several values of C and the best metric for a particular C can be used (maximum value for partition coefficient; minimum value for the others).

In order to use the Fuzzy C-Means algorithm to map the time series into a fixed number of clusters we've executed experiments with some values of m and several values of C to determine the best values for clustering. Five arbitrary values were used for m (1.01, 1.125, 1.25, 2.5, 5), while values from 2 to 25 were used for C. Other parameters for the algorithm that control the number of interactions were left large enough to ensure convergence. From this experiment we've concluded that for large values of m ($\geq$ 2.5) the results were practically indistinguishable, which led us to use $m = 1.25$. Determination of C was harder since there wasn't a single value of C that was a clear minima or maxima for the validity measures. We've used $C = 13$ as it seemed slightly better than other possible values accordingly to the compactness and separation metric.

The Fuzzy C-Means algorithm will yield two results we want to use to reduce the time dimension in our data for visual mining: a discrete cluster number that will be used to select distinct colors for plotting and the maximum membership value for each data vector. This value is obtained from the rows in the matrix μ and ranges from $1/C$ to 1, where higher values indicate stronger membership in a given cluster, and can be considered an indicator of quality of clustering for a particular data vector. The maximum membership value for each data vector was used to determine the size of the point to be plotted over the map.

Figure 2 shows the map with the data points plotted over it. Colors are arbitrary, points assigned (through defuzzification) to the same cluster have the same color. Sizes of the plotted points are relative to the maximum membership value for the point: smaller points have smaller maximum membership for all clusters. It must be pointed that the coordinates for the stations were not used in the clustering process itself, but only for the map generation.

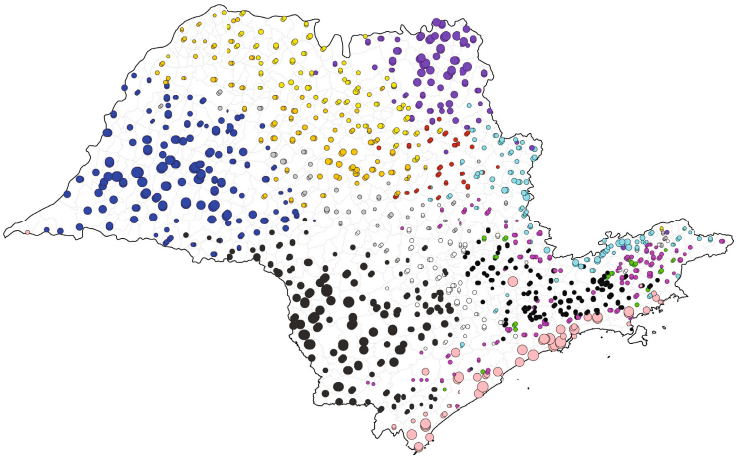


Fig. 2. Visualization using the Fuzzy C-Means results

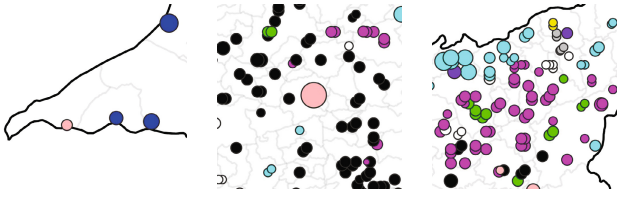


Fig. 3. Details of Figure 2

The map shown in Figure 2 and its details (Figure 3) presents clusters that are mostly contiguous in space, confirming our expectatives that weather phenomena (in this case, precipitation) have a moderate spatial correlation. Outliers (points that were assigned to clusters different from points nearby) are also easily identified due to the use of different colors, and the point size can also be used to identify data vectors that were strongly or weakly assigned to their clusters (e.g. central large point in the middle figure, small sub-clusters on the right).

5 Self-Organizing Maps

Another way to reduce the 84-dimension time vector to some few variables that can be plotted in a map is using the Self-Organizing Map [17] or SOM. This well-known neural network-based algorithm is able to reduce the dimension of a data set while preserving its topology: in other words, data vectors which were close in the original feature space will appear close in the new topology. The SOM uses as input the data and some parameters, the most important being the number of neurons that will be organized in a regular grid, and gives as output the best matching neuron for a particular data vector.

The topology-preserving feature of the SOM is particularly interesting for us: if we use an adequate representation for the neurons we can plot points that will appear similar if the clusters are similar. One natural choice for graphical representation of the neurons is to use a hue-based color system and map the hue and saturation values to the neurons in such a way that neurons that are topologically close have visually similar colors associated with them. Some of those mappings for a 2-dimensional, squared-lattice SOM are shown in Figure 4 (from the left to the right: mappings on a 3×3, 9×9 and 15×15 SOM).

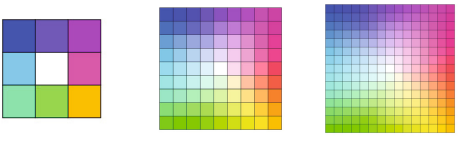


Fig. 4. Mapping of colors to neurons in SOMs with different sizes

Some authors (e.g. [18]) suggest that the number of neurons in a SOM should be a function of the number of input vectors. For our purposes we tried to visualize the data points with colors chosen from the color tables similar to those shown in Figure 4 and several numbers of neurons. The best results for visualization were achieved with SOMs of 9×9 and 15×15 neurons, but surprisingly, even SOMs of size 3×3 yielded easily interpretable results: data corresponding to weather stations with behavior different from the geographic neighbors were plotted in different colors. The map created with the processing of the data with a 15×15 SOM is shown in Figure 5, with some details shown in Figure 6.

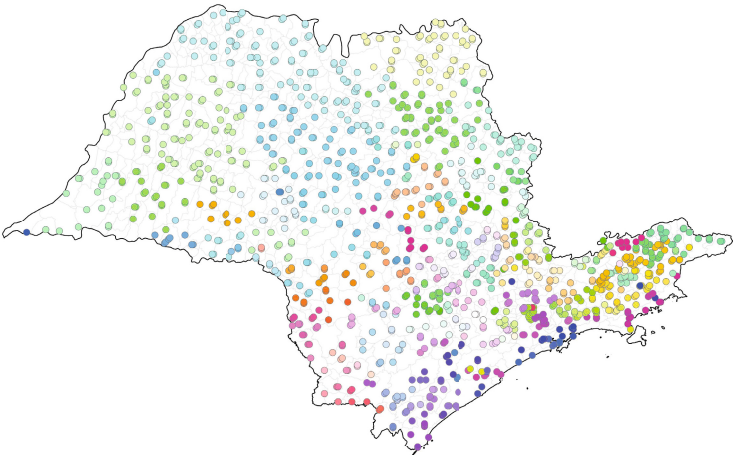


Fig. 5. Visualization using the SOM results

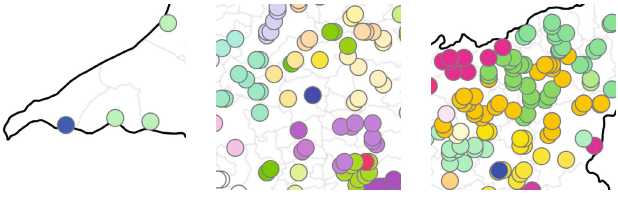


Fig. 6. Details of Figure 5

Figures 5 and 6 shows the general visual clustering structure for the data (data from weather stations geographically close have similar colors) and also some outliers (e.g. left part of Figure 6). One advantage of using a topology-preserving algorithm is that we can perceptually evaluate how much a data point is different from the others.

6 Conclusions and Future Work

In this paper we evaluated two techniques for dimension reduction or mapping in order to create visual representation of time series over geographic coordinates. Ultimately these techniques may be incorporated on the data collection systems at INPE's CPTEC to help identify potentially problematic data subsets. We must point that we did not cluster time series for classification or characterization, which, depending on the metric used, can be considered meaningless [19,20]. In this research we used the metrics extracted from different clustering algorithms to visualize spatio-temporal data.

Of the two techniques the one based on the SOM was considered to be more easily interpretable since it is possible to identify data points that are different from a cluster and at the same time perceptually evaluate how much it is different. Both techniques has been used by researchers in several domains, but due to its features SOM-based techniques are more prevalent, particularly for analysis of data with spatial components (e.g. [21]).

References

1. Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability, 1st edn. Cambridge Press (2003)
2. Expert Team on Requirements of Data from Automatic Weather Stations: Final report (2002), http://www.wmo.int/pages/prog/www/OSY/Meetings/ET-AWS1-2002/Final-Report.pdf
3. Garcia, J.R.M., Carvalho, L.S.M., Júnior, H.C., Sanches, M.B.: BDC - banco de dados climatológico. In: Proceedings do XIV Congresso Brasileiro de Meteorologia (2006)
4. Simoff, S.J., Böhlen, M.H., Mazeika, A.: Visual Data Mining: An Introduction and Overview. In: Simoff, S.J., Böhlen, M.H., Mazeika, A. (eds.) Visual Data Mining. LNCS, vol. 4404, pp. 1–12. Springer, Heidelberg (2008)

5. Keim, D., Panse, C., Sips, M.: Visual Data Mining of Large Spatial Data Sets. In: Bianchi-Berthouze, N. (ed.) DNIS 2003. LNCS, vol. 2822, pp. 201–215. Springer, Heidelberg (2003)
6. Macêdo, M., Cook, D., Brown, T.: Visual data mining in atmospheric science data. Data Mining and Knowledge Discovery 4, 69–80 (2000)
7. Kopanakis, I., Pelekis, N., Karanikas, H., Mavroudkis, T.: Visual Techniques for the Interpretation of Data Mining Outcomes. In: Bozanis, P., Houstis, E.N. (eds.) PCI 2005. LNCS, vol. 3746, pp. 25–35. Springer, Heidelberg (2005)
8. Andrienko, N., Andrienko, G.: Exploratory Analysis of Spatial And Temporal Data: A Systematic Approach. Springer (2006)
9. Huang, M.L., Nguyen, Q.V.: Context Visualization for Visual Data Mining. In: Simoff, S.J., Böhlen, M.H., Mazeika, A. (eds.) Visual Data Mining. LNCS, vol. 4404, pp. 248–263. Springer, Heidelberg (2008)
10. Andrienko, G., Andrienko, N., Gatalsky, P.: Visual Mining of Spatial Time Series Data. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 524–527. Springer, Heidelberg (2004)
11. Watanabe, C., Touma, E., Yamauchi, K., Noguchi, K., Hayashida, S., Joe, K.: Development of an Interactive Visual Data Mining System for Atmospheric Science. In: Labarta, J., Joe, K., Sato, T. (eds.) ISHPC 2006 and ALPS 2006. LNCS, vol. 4759, pp. 279–286. Springer, Heidelberg (2008)
12. Inselberg, A.: Parallel Coordinates – Visual Multidimensional Geometry and Its Applications. Springer (2009)
13. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms, 1st edn. Plenum Press (1987)
14. Chi, Z., Yan, H., Pham, T.: Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition. World Scientific Publishing (1996)
15. Yang, M.S., Wu, K.L.: Unsupervised possibilistic clustering. Pattern Recogn. 39(1), 5–21 (2006)
16. Wu, K.L.: Analysis of parameter selections for fuzzy c-means. Pattern Recogn. 45(1), 407–415 (2012)
17. Kohonen, T.: Self-Organizing Maps, 2nd edn. Springer (1997)
18. Barreto, G.: Time Series Prediction with the Self-Organizing Map: A Review. In: Hammer, B., Hitzler, P. (eds.) Perspectives of Neural-Symbolic Integration. SCI, vol. 77, pp. 135–158. Springer, Heidelberg (2007)
19. Keogh, E., Lin, J.: Clustering of time-series subsequences is meaningless: implications for previous and future research. Knowledge and Information Systems 8, 154–177 (2005)
20. Chen, J.: Making subsequence time series clustering meaningful. In: Fifth IEEE International Conference on Data Mining, 8 p. (November 2005)
21. Koua, E., Kraak, M.J.: Geovisualization to support the exploration of large health and demographic survey data. International Journal of Health Geographics 3, 1–13 (2004)

RSGALS-SVM: Random Subspace Method Applied to a LS-SVM Ensemble Optimized by Genetic Algorithm

Carlos Padilha, Adrião D. Dória Neto, and Jorge D. Melo

Federal University of Rio Grande do Norte, Department of Computer Engineering and Automation, 59078-900 Natal, Brazil
{carlosalberto,adriao,jdmelo}@dca.ufrn.br

Abstract. The Support Vector Machines (SVMs) have received great emphasis in the pattern classification due its good ability to generalize. The Least Squares formulation of SVM (LS-SVM) finds the solution by solving a set of linear equations instead of quadratic programming. Both the SVMs and the LS-SVMs provide some free parameters that have to be tuned to reflect the requirements of the given task. Despite their high performance, lots of tools have been developed to improve them, mainly the development of new classifying methods and the employment of ensembles. So, in this paper, our proposal is to use both the theory of ensembles and a genetic algorithm to enhance the LS-SVM classification. First, we randomly divide the problem into subspaces to generate diversity among the classifiers of the ensemble. So, we apply a genetic algorithm to optimize the classification of this ensemble of LS-SVM, testing with some benchmark data sets.

Keywords: Pattern Classification, LS-SVM; Ensembles, Genetic Algorithm, Random Subspace Method.

1 Introduction

The study of the Support Vector Machines (SVMs) is one of the leading research areas in pattern classification. SVMs are linear statistical learning machines whose training is based on the Structural Risk Minimization principle [1]. Through the quadratic programming, an optimal decision hyperplane is found, thus maximizing the margin of separation of the classes. The Least Squares formulation of SVM, called LS-SVM, was introduced by Suykens [2] and it finds the solution by solving a linear system instead of quadratic programming. It uses equality constraints instead of inequality constraints in the problem formulation. Both the SVMs and the LS-SVMs provide some parameters that have to be tuned to reflect the requirements of the given task. Despite their high performance, several techniques have been employed in order to improve them, either by developing new training methods [3] or by creating ensembles [4].

The most popular ensemble learning methods are Bagging [5], Boosting [6] and the Random Subspace Method (RSM) [7]. In Bagging, one samples the training set, generating random independent bootstrap replicates [8], constructs the classifier on each of these, and aggregates them by a simple majority vote in the final decision

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 253–260, 2012.

rule. In Boosting, classifiers are constructed on weighted versions of the training set, which are dependent on previous classification results. Initially, all objects have equal weights, and the first classifier is constructed on this data set. Then, weights are changed according to the performance of the classifier. Erroneously classified objects get larger weights, and the next classifier is boosted on the reweighted training set. In this way, a sequence of training sets and classifiers is obtained, which is then combined by single majority voting or by weighted majority voting in the final decision. In the RSM, classifiers are constructed in random subspaces of the data feature space. These classifiers are usually combined by single majority voting in the final decision rule.

In our previous work [9], we used a GA to analyze the importance of each SVM in the ensemble by means of a weight vector. The diversity in ensemble was generated providing different parameter values for each model. In this paper, we propose another way to generate diversity in ensemble and extend the use of GA, called RSGALS-SVM. We use the combination of the RSM and GA to enhance the classification of a LS-SVM ensemble. First, we use the RSM, constructing models in random subspaces of an *n*-dimensional problem, so that each LS-SVM will be responsible for the classification of a subproblem. Then, the GA is used to minimize an error function and therefore it will act finding effective values for the parameters of each model in the ensemble and a weight vector, measuring the importance of each one in the final classification. That way, if, for example, there is one LS-SVM whose decision surface works better than the others, the GA will find the weight vector so that the final classifying is the best possible. Then we compare our results to proposed method with some other algorithms.

This paper is organized as follows: Section 2 introduces the LS-SVM and some of its characteristics, and also has a brief explanation on genetic algorithms and its mechanism. Section 3 describes our proposed method, while Section 4 has the experimental results and its analysis. Section V presents the conclusion of the paper.

2 Theoretical Background

2.1 Least Squares Support Vector Machines

In this Section we consider first the case of two classes. Given a training set of N data points$\{x_k, d_k\}_{k=1}^{N}$, where $x_k \in \mathbb{R}^n$ denotes the *k*-th input pattern and $d_k \in \mathbb{R}$ the *k*-th output pattern, the support vector method approach aims at constructing a classifier of the form:

$$y(x) = sign\left[\sum_{k=1}^{N} \alpha_k d_k K(x, x_k) + b\right] \tag{1}$$

where α_k are support values and b is a real constant. For $K(\cdot,\cdot)$ one typically has the following choices: $K(x, x_k) = x^T x_k$ (linear SVM); $K(x, x_k) = (x^T x_k + 1)^p$ (polynomial SVM of degree p); $K(x, x_k) = \exp\left(-\frac{1}{2\sigma^2}\|x - x_k\|^2\right)$ (RBF SVM); $K(x, x_k) = \tanh\left(\beta_o x^T x_k + \beta_1\right)$ (MLP SVM), where σ, β_o and β_1 are constants.

For the case of two classes, one assumes

$$\begin{cases} w^T\varphi(x_k) + b \geq +1, se\ y_k = +1 \\ w^T\varphi(x_k) + b \leq -1, se\ y_k = -1 \end{cases} \tag{2}$$

which is equivalent to

$$d_k[w^T\varphi(x_k) + b] \geq +1, k = 1, \dots, N \tag{3}$$

where $\varphi(\cdot)$ is a nonlinear function which maps the input space into a higher dimensional space. LS-SVM classifiers as introduced in [2] are obtained as solution to the following optimization problem:

$$\min_{w,b,e} Jp(w,b,e) = \frac{1}{2}w^Tw + C\frac{1}{2}\sum_{k=1}^{N} e_k^2 \tag{4}$$

subject to the equality constraints

$$d_k[w^T\varphi(x_k) + b] = 1 - e_k, k = 1, \dots, N \tag{5}$$

One defines the Lagrangian

$$L(w,b,e;\alpha) = Jp - \sum_{k=1}^{N} \alpha_k\{d_k[w^T\varphi(x_k) + b] - 1 + e_k\} \tag{6}$$

where α_k are Lagrange multipliers, which can be either positive or negative due to equality constraints as follows from Karush-Kuhn-Tucker (KKT) conditions [10].

The conditions for optimality

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{k=1}^{N} \alpha_k d_k \varphi(x_k) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^{N} \alpha_k d_k = 0 \\ \frac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = Ce_k, k = 1, \dots, N \\ \frac{\partial L}{\partial \alpha_k} = 0 \rightarrow d_k[w^T\varphi(x_k) + b] = 1 - e_k = 0, k = 1, \dots, N \end{cases} \tag{7}$$

can be written after elimination of *w* and *e* as the linear system [2]:

$$\begin{bmatrix} 0 & D^T \\ D & ZZ^T + C^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{1} \end{bmatrix} \tag{8}$$

Where $Z = [\varphi(x_1)^T d_1; \dots; \varphi(x_N)^T d_N], D = [d_1; \dots; d_N], \vec{1} = [1; \dots; 1], \alpha = [\alpha_1; \dots; \alpha_N]$, $e = [e_1; \dots; e_N]$. The Mercer's condition is applied to the matrix $\Omega = ZZ^T$ with

$$\Omega_{kl} = d_k d_l \phi(x_k)^T \phi(x_l) = d_k d_l K(x_k, x_l) \tag{9}$$

2.2 Genetic Algorithms

The Genetic Algorithm (GA) [11] is a search-based metaheuristic inspired by the Darwinian principle of the evolution and by the theory of genetics. The GA was

initially proposed by John Holland in 1975, as part of his attempts to explain processes occurring in natural systems and to build artificial systems based on such processes. It operates on a population of solutions (chromosomes) and then proceeds according to the "survival of the fittest" where a fitness function will analyze how fit is that individual and so evaluate its survivability.

Conceptually, it adopts two separated spaces well defined called search space and solution space. The first one is formed by encoded solutions (genotype) for the problem and the second one, the space of the real solutions (phenotype), where they will be evaluated by the fitness function.

The canonic genetic algorithm (simple GA) can be seen like a union of the three components: (a) the chromosome representation, commonly either a string of bits or a set of floating-point values; (b) the synthesis of new individuals, using biologically-inspired operators of crossover and mutation; (c) selection of the parents of the next generation, using the fitness function, through a roulette wheel, a tournament, or another selection method.

The crossover is an exchange of information between parent chromosomes, creating new chromosomes. In bit strings, it usually happens through the exchange of determined bit indexes, while in floating-point values, it is an arithmetical operation between the chromosomes. It also occurs with a certain probability, and the choice of which individuals will participate in the crossover might also be made according to the fitness function.

The mutation is a change in the value of a gene on the chromosome. Like the crossover, it also has a probability of occurrence. The mutation might be either a simple inversion of a gene in the bit string, or a small addition or subtraction from the chromosome, in the floating-point case.

The selection is responsible for the perpetuation of the good features of the population. Chromosomes are selected from the population to be parents to crossover. The problem is how to select these chromosomes. According to Darwin's evolution theory the best ones should survive and create new offspring. There are many methods how to select the best chromosomes, for example roulette wheel, tournament, rank selection, steady state selection and some others.

3 RSGALS-SVM

Our main objective is to improve the LS-SVM ensemble performance through the combination of RSM and GA. In order to create this set of LS-SVMs, we delved a little into the ensemble theory.

In [12,13,14,15] we see that an effective ensemble should consist of a set of models that are not only highly accurate, but ones that make their errors on different parts of the input space as well. Thus, varying the feature subsets used by each member of the ensemble should help promote this necessary diversity. From [4] we see that the SVM kernel that allows for higher diversity from the most popular ones is the Radial Basis Function kernel, because its Gaussian width parameter, σ, allows detailed tuning.

Therefore, the combination of RSM and GA is used to generate highly accurate models and promote disagreement among them. Given an n-dimensional problem, we

use RSM to divide it randomly into M subspaces of the data feature space, thus each LS-SVM will be responsible for the classification of the problem based on the information that your subspace provides.

Once defined the division into subspaces of the original problem, we define how the GA will be used in this work. The GA will act on two different levels of the ensemble, on the parameters and output of each model. At the first level, the GA will find effective values of σ and C, the regularization term that controls the tradeoff between allowing training errors and forcing rigid margins, for M LS-SVMs. At the second level, the GA will find a weight vector w, measuring the importance of each LS-SVM in the final classifying. The final classification is obtained by a simple linear combination of the decision values of the LS-SVMs with the weight vector. This way, the representation of each individual of our population is defined as a vector containing the adjustable parameters and weights.

$$chromosome = [\sigma_1, \sigma_2, \dots, \sigma_M, C_1, C_2, \dots, C_M, w_1, w_2, \dots, w_M]$$

where M is the number of LS-SVMs.

The fitness function of our GA is the error rate of the ensemble and can be seen as:

$$\begin{gathered} \varepsilon(\boldsymbol{\sigma}, \boldsymbol{C}, \boldsymbol{w}) = \|\boldsymbol{d} - \boldsymbol{y}\|^2 \\ \boldsymbol{d} = [d_1, \dots, d_N], \boldsymbol{y} = [y_1, \dots, y_N] \\ y_k = \boldsymbol{o}^T \boldsymbol{w}, k = 1, \dots, N \end{gathered} \tag{10}$$

where $\boldsymbol{d}$ contains the output patterns, $\boldsymbol{y}$ contains the final hypothesis, $\boldsymbol{o}$ contains the LS-SVMs outputs for a given input pattern x_k and $\boldsymbol{w}$ is the weight vector.

So, we can formulate the optimization problem to be solved by the GA:

$$\min_{\boldsymbol{\sigma}, \boldsymbol{C}, \boldsymbol{w}} \varepsilon(\boldsymbol{\sigma}, \boldsymbol{C}, \boldsymbol{w}) = \|\boldsymbol{d} - \boldsymbol{y}\|^2 \tag{11}$$

subject to

1. $\sum_{i=1}^{M} w_i = 1$
2. $\sigma_i, C_i > 0$ and $w_i \geq 0, i = 1, \dots, M$

The initial population of the GA was generated randomly. We employed stochastic-uniform selection, which divides the parents into uniform-sized sections, each parent's size being determined by the fitness scaling function. In each of these sections, one parent is chosen. The mutation function is a Gaussian; the individuals in mutation are added with a random number from a Gaussian distribution with mean zero, and variance declining at each generation. The crossover function is the scattered crossover; a binary vector is generated, and the elements of this vector decide the outcome of the crossover. If the element is a 1, the corresponding gene of the child will come from the first parent. If it's a 0, said gene will come from the second parent. The size of the population is 20, at each generation two elite individuals are kept for the next generation, and the fraction generated by crossover is 0.8. The GA runs for 100 generations. Table 1 show the method's pseudo code.

Table 1. RSGALS-SVM algorithm

Given:

$S = \{(x_1, d_1), \ldots, (x_N, d_N); x_k \in \mathbb{R}^n, d_k \in \{-1,1\}\}$, the input set

Procedure:

Generate from S the training set P and the test set V

Randomly divide P into M subspaces of features

Generate M LS-SVM to compose the ensemble, each one will be trained using one of those groups of features

Call the GA to solve the optimization problem:

$$\min_{\boldsymbol{\sigma},\boldsymbol{C},\boldsymbol{w}} \varepsilon(\boldsymbol{\sigma}, \boldsymbol{C}, \boldsymbol{w}) = \|\boldsymbol{d} - \boldsymbol{y}\|^2$$

subject to

1. $\sum_{i=1}^{M} w_i = 1$
2. $\sigma_i, C_i > 0$ and $w_i \geq 0, i = 1, \ldots, M$

Retrieve the optimal values for $\boldsymbol{\sigma}^*, \boldsymbol{C}^*$ and the optimal weight vector $\boldsymbol{w}^*$

Evaluate the ensemble using V with the same division made in P

Output: Final Classification:

$$C(x) = sign(\boldsymbol{o}^T \boldsymbol{w})$$

4 Experimental Results

To evaluate the performance of proposed method, tests were performed using 11 two-class benchmark data sets from [16] that include various types of classification problems (real-world problems and artificial). Table 2 shows the number of inputs, training data, test data, and training and test data sets of each data set. The data sets chosen vary across a number of dimensions including: the type of the features in the data set (continuous, discrete or a mix of the two) and number of examples in the data set. We compared the proposed method to a single RBF Network, AdaBoost with RBF Networks and SVM (with Gaussian kernel) and their results were obtained from [17]. All results are averaged over all sets in every problem. In all tests, we used an ensemble composed of 5 LS-SVM.

Table 3 shows the average recognition rates and their standard deviations of the validation data sets by RBF, AdaBoost, SVM and proposed method.

The results of AdaBoost are in almost all tests worse than the single classifier. Analyzing these results, this is clearly due to the overfitting of AdaBoost. In [17], the authors explain that if the early stopping is used then this effect is less drastic but still observable.

The averaged results of RSGALS-SVM are a bit better than the results achieved by other classifiers in most tests (7/11). In 4 tests (German, F. solar, Twonorm and Waveform) RSGALS-SVM is significant better than SVM. The results of SVM are often better than the results of RBF classifier.

Table 2. Two-class benchmark data sets

Data	Inputs	Training	Test	Sets
B. cancer	9	200	77	100
Diabetis	8	468	300	100
German	20	700	300	100
Heart	13	170	100	100
Image	18	1300	1010	20
Ringnorm	20	400	7000	100
F. solar	9	666	400	100
Splice	60	1000	2175	20
Thyroid	5	140	75	100
Twonorm	20	400	7000	100
Waveform	21	400	4600	100

Table 3. Comparison between the RSGALS-SVM, a single RBF classifier, AdaBoost (AB) and Support Vector Machine (SVM). The best average recognition rate is shown in boldface.

Data	RBF	AB	SVM	RSGALS-SVM
B. cancer	72.4±4.7	69.6±4.7	74±4.7	**74.1±0.1**
Diabetis	75.7±1.9	73.5±2.3	**76.5±1.7**	76.3±0.3
German	75.3±2.4	72.5±2.5	76.4±2.1	**78±0.6**
Heart	82.4±3.3	79.7±3.4	84±3.3	**87±0.5**
Image	96.7±0.6	**97.3±0.7**	97±0.6	96.3±0.1
Ringnorm	**98.3±0.2**	98.1±0.3	**98.3±0.1**	97.5±0.3
F. solar	65.6±2.0	64.3±1.8	67.6±1.8	**70.3±0.6**
Splice	90±1.0	89.9±0.5	89.1±0.7	**90.1±0.1**
Thyroid	95.5±2.1	**95.6±0.6**	95.2±2.2	94.7±0.6
Twonorm	97.1±0.3	97±0.3	97±0.2	**97.9±0.2**
Waveform	89.3±1.1	89.2±0.6	90.1±0.4	**92.7±0.2**

5 Conclusion

In this work, we proposed two changes in relation to the previous work [9], we incorporated the RSM to make the feature selection, creating diversity among the LS-SVMs in the ensemble, and we extended the use of GA to find good values for the parameters (σ, C). The search space of these parameters is enormous in complex problems due to their large range of values. This is why we extended this global search technique (GA) to find their values. We tested the previous work using 4 data sets (Image, Ringnorm, Splice and Waveform) and this work got better results in all cases.

We compared the proposed method RSGALS-SVM to a single RBF classifier, AdaBoost with RBF networks and SVM (with Gaussian kernel) and it achieved better results than these traditional classifiers in most tests.

Many improvements are possible and need be explored. For example, we can investigate further expand the use of GA to make the feature selection, as in [18], but keeping the fitness function used in this work.

References

1. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons Inc., New York (1998)
2. Suykens, J.A.K., Vandewalle, J.: Least-Squares Support Vector Machine Classifiers. Neural Processing Letters 9(3) (1999)
3. Osuna, E., Freund, R., Girosi, F.: An Improved Training Algorithm for Support Vector Machines. In: NNSP 1997 (1997)
4. Lima, N., Dória Neto, A., Melo, J.: Creating an Ensemble of Diverse Support Vector Machines Using Adaboost. In: Proceedings on International Joint Conference on Neural Networks (2009)
5. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
6. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings 13th International Conference on Machine Learning, pp. 148–156 (1996)
7. Ho, T.K.: The Random subspace method for constructing decision forests. IEEE Transactions Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
8. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman & Hall, New York (1993)
9. Padilha, C., Lima, N., Dória Neto, A., Melo, J.: An Genetic Approach to Support Vector Machines in classification problems. In: Proceedings on International Joint Conference on Neural Networks (2010)
10. Fletcher, R., Johnson, T.: On the stability of null-space methods for KKT systems. SIAM J. Matrix Anal. Appl. 18(4), 938–958 (1997)
11. Castro, L., Zuben, F.V.: Algoritmos Genéticos. Universidade Estadual de Campinas (2002), `ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia707_02/topico9_02.pdf`
12. Kuncheva, L., Whitaker, C.: Measures in diversity in classifier ensembles and their relationship with ensemble accuracy. Machine Learning 51(2), 181–207 (2003)
13. Hansen, L., Salamon, P.: Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 993–1001 (1990)
14. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: Advances in Neural Information Processing Systems, vol. 7, pp. 231–238. MIT Press, Cambridge (1995)
15. Opitz, D., Shavlik, J.: Actively searching for an effective neural-network ensemble. Connection Science 8(3/4), 337–353 (1996)
16. Benchmark Repository (2012), `http://www.fml.tuebingen.mpg.de/Members/raetsch/benchmark`
17. Rätsch, G., Onoda, T., Müller, K.-R.: Soft Margins for Adaboost. Machine Learning 42 (2001)
18. Opitz, D.: Feature Selection for Ensembles. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence (1999)

Optimizing the Extreme Learning Machine Using Harmony Search for Hydrologic Time Series Forecasting

Ivna Valença[1] and Mêuser Valença[2]

[1] Federal University of Pernambuco
Informatics Center
Recife-PE, Brazil
icbv@cin.ufpe.br
[2] University of Pernambuco
Department of Systems and Computer
Recife-PE, Brazil
meuservalenca@gmail.com

Abstract. Lately, the research related to time series forecasting has been an area of considerable interest in different fields. It is very important to predict the behavior of the time series but it is not an easy task. Several models to aim this issue have been developed over the years, taking into account their peculiarities. Artificial Neural Networks (ANNs) are one of them. ANNs received much attention, and a great number of papers have reported successful experiments and practical tests. In this paper, a hybrid approach is proposed based on Harmony Search (HS) to select the number of hidden neurons and their weights for Extreme Learning Machine (ELM) algorithm, called HS-ELM. In addition, we provide experimental results from the application of our algorithm HS-ELM in real stream flow time series to show its effectiveness and usefulness.

Keywords: Hybrid Intelligent System, Harmony Search, Extreme Learning Machine, Time Series Forecasting.

1 Introduction

The research on time series forecasting is being applied in different areas. Time series forecasting is based on the use of past observations of a data set to build a model which will be able to predict a future value. Over the past decades a lot of searches were done to develop and improve this process. At the beginning, the linear models were used to predict the future values, e.g. ARIMA models [1]. These models assume that the behaviour of the data set are linear, meaning that the variables would have only linear dependence. However, most of the real-world time series are often non-linear. So, to attack this problem some other methods were studied like the Bilinear model, the Threshold Autoregressive (TAR) model, the Autoregressive Conditional Heteroskedasticity (ARCH) and

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 261–269, 2012.

the Generalized ARCH (GARCH). A limitation to use these non-linear models is that we don't know the relationship between the variables of the data set which is generally difficult to predict. Nowadays, the Artificial Neural Networks (ANN) are presented as an alternative approach to traditional statistical methods to solve problems on the prediction of the time series [2].

This study therefore proposes: a heuristic search of an architecture that avoids the need to test all possible architectures for an Artificial Neural Network, implementing a method that is self-adjusting and that has shown better results compared to existing heuristics (these heuristics whose architectural decision is made manually and tediously, taking too long to decide which architecture responds with better results). This method dynamically decides how many processing units are in the hidden layer and their weights, using the method of Harmony Search with the learning algorithm Extreme Learning Machine (which it is considered much faster maintaining the good results than the Backpropagation) [3, 4].

The remaining of this paper is organized as follows. Section 2.1 brings a brief presentation of Harmony Search, followed by section 3 which describes the Extreme Learning Machine. Section 4 presents the proposed solution and an implemented model called Harmony Search with Extreme Learning Machine (HS-ELM). Section 5 presents the performed case study with the experiments and obtained results. Finally, section 6 concludes the paper.

2 Search Method

There are two types of search method used to set the weights of the Neural Networks: evolutionary and non-evolutionary methodologies. The first one, use Evolutionary Algorithms, e.g. Genetic Algorithms (GA). The second, tries to optimize the performance of the network using techniques to make modifications on itself during the training phase, e.g. Cascade-Correlation (CC) and Dynamic Node Creation (DNC). Many works have been done in this area with Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), for example, but even when they present good results, the computational cost is really high, since the GA and PSO works with populations. In this work, we propose the use of Harmony Search to optimize the ELM because it just provide one single solution to be evaluate at each interaction.

2.1 Harmony Search

Harmony Search (HS) is one technique for search and optimization problems introduced in 2001 [5] that have been very successful in managing problems [6–10]. HS is a meta-heuristic algorithm, mimicking the process of improvising music players [5]. The algorithm HS has been very successful in a variety of optimization problems [6–10], having several advantages over traditional optimization techniques [10]. Harmony search has two main differences when compared with other search methods based on population: only generates a new individual to

the population at each iteration so that only one individual needs to be evaluated; and the initial selection of the set of variables is done probabilistically.

3 Extreme Learning Machine

As a learning technique, Extreme Learning Machine (ELM) has demonstrated good potentials to resolving regression and classification problems. Recently, ELM techniques have received considerable attention in computational intelligence and machine learning communities, in both theoretic study and applications [3].

Extreme learning machine (ELM) was proposed in Huang et al. [11] as a new class of learning algorithm for single-hidden layer feedforward neural network (SLFN) much faster than the traditional gradient-based learning strategies. However, ELM random determination of the input weights and hidden biases may lead to non-optimal performance, and it might suffer from the overfitting as the learning model will approximate all training samples well [12]. Usually, the initial input weights and hidden biases of ELM are randomly chosen, and then the output weights are analytically determined by using Moore-Penrose (MP) generalized inverse. However, ELM may need higher number of hidden neurons due to the random determination of the input weights and hidden biases.

4 Proposed Model: HS-ELM

Since it is difficult to find a particular algorithm that is the best for all applications in this article we intend to compare the use of Harmony Search and Extreme Learning Machine with ELM to forecast time series. The focus is to determine the number of neurons and the weights connections of the networks. The steps of the Harmony Search with ELM (HS-ELM) algorithm are:

1. Initialize the algorithm parameters;
2. Initialize at random, the Harmony Memory (HM);
3. Calculate through the ELM the MSE using the neurons and weights that were set in the HM;
4. Update the HM putting smaller errors in the first rank;
5. Improvise a new harmony in HM;
6. Repeat steps 3-5 until the stopping criterion (e.g. maximum number of cycles or/and the error with small variation).

The algorithm returns for each simulated population all the settings of the HM. The last setting is taken into account. The harmony of the first rank has the lowest error found. The Harmony Memory is a vector composed by: the first values correspond to the neurons assets (0 for non-active and 1 for active) and the second part are composed by the weights (values between 0 and 1). The configuration of the Harmony Memory from the HS-ELM are demonstrate in Fig. 1. The h represents the number of harmonies, n is the maximum number of neurons and w is the maximum number of weights.

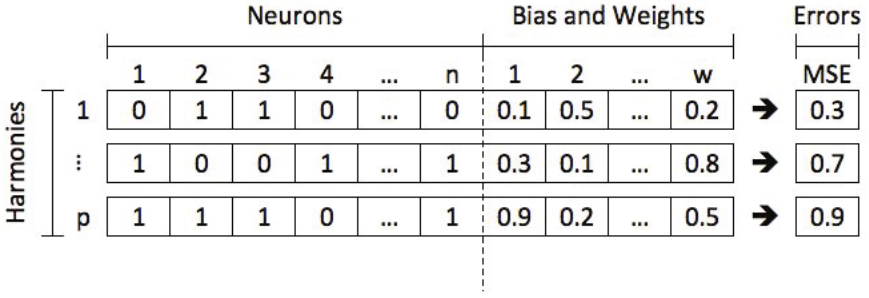


Fig. 1. Configuration of the Harmony Memory from the HS-ELM

For each population simulated the algorithm returns all HM's settings. The last setting is taken into account and the harmony on the first rank that has the lowest error found. The Harmony Search is responsible for selecting the number of neurons according to the probability equals 50%. When a new harmony is created, the neurons (notes) set with 1 correspond to the neurons that will be included in the ELM. The weights of these neurons are continuous and directly encoded in the individuals and passed to the neural network ELM.

5 Case Study

The National Electricity System Operator (ONS) from Brazil with the objective of optimizing the centralized dispatch of power plant members of the National Interconnected System (SIN), has among its main tasks to forecast the weekly streamflow for the Monthly Operation Program (PMO) and its revisions. The main model that supports this prediction is the univariate stochastic model called PREVIVAZ [13]. The methodology of this model includes autoregressive and moving average models, with stationary or periodic structure, in other words, the AR(p) and PAR(p) models, with p up to order 4 and PARMA(p,q) and ARMA(p,q) models, with p up to order 3 and q up to order 1. Transformations can be logarithmic, Box & Cox or without transformation [14]. However, the ONS has sought different technological alternatives for streamflow forecasting, with the aim of improving the quality of this input to the operation of the SIN.

Thus, since January 2006, ONS began using a model that incorporates the use of information provided from observed and predicted rainfall to the process of streamflow forecasting, within the PMO and its weekly review. Among this models we have the SMAPMEL model (for the Paraná river basin, in the incremental portion of Itaipu), MGB-IPH model (for the Paranaíba river basin, the stretch between the Itumbiara and São Simão), MPCV model (for the Uruguai river basin), FUZZY model (for the Iguaçu river basin), SMAP model (for the Rio Grande upper/middle river basin till Porto Colômbia) and NEURO3M model (for the Três Marias basin in São Francisco river basin). For the purpose of planning the national interconnected system, Brazil is divided into four regions:

north, northeast, south and southeast. Among these regions, which has higher forecast errors is the south region (Iguaçu River, Jacuí River and Uruguai River). Therefore, the development of new models that can improve the quality of forecasts especially in this region has fundamental importance.

This study proposes a new model (HS-ELM) for forecasting streamflow. To evaluate the performance of the proposed method will be used mainly plants located in south of Brazil and others considered important for the the remaining regions. Our goal is to evaluate the performance of the proposed model with the ELM and PREVIVAZ models.

5.1 Data Set and Metrics

In order to evaluate the performance of the proposed HS-ELM model some experiments were performed, which is described just below. In this paper, the river flows time series used were:

- *Jordão*, *Foz do Areia*, *Salto Santiago* and *Salto Caxias* located in the *Iguaçu* river; *Passo Real* in *Jacuí* river and *Itá*, *Passo Fundo* and *Quebra Queixo* in *Uruguai* river (located in South of Brazil);
- *Furnas* in *Grande* river, *Manso* in *Paraguai* river and *Itaipu* in *Paraná* river (located in the Southeast of Brazil);
- *Tucuruí* in *Tocantins* river (located in the North of Brazil); and
- *Boa Esperança* in the *Parnaíba* river (located in the Northeast of Brazil).

All bases have average daily natural inflow from 1973 to 2010 available in ONS website [15]. The average used is the mean of each data. The time series of the south region are the most difficult to predict since does not exist seasonality and has higher deviations over the years. The number of ahead forecast is equal to 1 and determines the mean of one week ahead (mean of 7 values) in order to compare with the results of the models actualy used in the electricity sector and published in the ONS website. Also, we present the mean result for 12 days ahead (five days to complete the current week and seven for the weekly operation program). Each data set was divided into three parts: 50% of the examples are used for training the neural network, 25% are used for validation and 25% are used for testing. The validation set was used to verify the generalization capability of the network (setting the number of neurons in the hidden layer).

The metrics used for performance evaluation were three and they are described below. First, the Mean Absolute Percentage Error (MAPE), which is the average absolute difference between predicted and observed values, expressed as a percentage of the observed values. This indicator can take values between $(\infty, 0]$ being MAPE $= 0$ the value corresponding to perfect forecasts in the period (this is characterized as an indicator of "short term").

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{O_t - P_t}{O_t}\right| \tag{1}$$

Where O_t is the actual value and P_t is the forecast value. Second, Nash-Sutcliffe model efficiency coefficient (NS) is used to evaluate the predictive power of hydrological models. This indicator can take values between $(-\infty, 1]$. NS equals 1 corresponds to perfect prediction of the flows in the period in focus. NS less then 0 shows that using the model predictions is worse than using the average value of observations (this is characterized as an indicator of "long term").

$$\mathrm{NS} = 1 - \frac{\sum_{t=1}^{n}(P_t - O_t)^2}{\sum_{t=1}^{n}(P_t - \bar{O})^2} \tag{2}$$

Where $\bar{O}$ is the average value observed. Third, the index proposed by the ONS, Distance Multicriteria (DM) uses the previous results as an ordered pair (NS, 1-MAPE) and calculates the Euclidean distance to the optimal point.

$$\mathrm{DM} = \sqrt{(1 - \mathrm{NS})^2 + \mathrm{MAPE}^2} \tag{3}$$

To validate the results we applied the Wilcoxon test hypothesis with 95% confidence, making sure that the medians of the compared systems are equal or not, as the following:

- h_0: system 1 median = system 2 mean
- h_1: system 1 median $\neq$ system 2 mean

5.2 Experiments and Results

The parameters of the algorithm are described below. All experiments were performed with the following parameters: number of simulations = 30, population size = 12, maximum number of neurons in the hidden layer = 30, maximum number of generations = 150, probabilities of the HS = 0.95. For stopping criteria and avoid overfitting, in each generation the fitness is evaluated, which is calculated as the inverse of the mean square error of validation set. If the network is no longer generalizing, the counter is incremented. When it reaches the value 30, the training stops. If it does not occurs the training stop after 150 generations. The number of inputs variables was 14 past values (lags), in all cases, to forecast the 7 or 12 days ahead. This number was based on another study already done with streamflow forecasting [16, 17].

In Experiment 1, we compared the use of traditional ELM and the proposed HS-ELM model. The results are shown in Table 1. After implementation of Wilcoxon test it was concluded that the MAPE was always less or equal to all series with the proposed model (HS-ELM). The results were considered statistically equal for the series: Passo Real, Passo Fundo, Quebra Queixo, Manso e Furnas. For the other series the MAPEs are smaller using HS-ELM. In Table 1 it is possible to observe the average number of neurons in the hidden layer selected by the proposed method compared with the process of trial and error from the ELM network. The average number of neurons selected by HS-ELM is always inferior to the process of trial and error.

Table 1. Results for the forecasting ELM x HS-ELM (values displayed in percent) and the number of neurons in the hidden layer ELM x HS-ELM

Time Series	MAPE (%)				Number of Neurons		
Models	ELM		HS-ELM		ELM	HS-ELM	
	Mean	STDV	Mean	STDV	Neurons	Neurons	STDV
Jordão	26.00	0.32	25.80	0.19	30	23.80	4.60
Foz do Areia	21.80	0.36	21.50	0.29	30	22.20	4.30
Salto Santiago	21.00	0.45	20.70	0.97	30	22.00	6.40
Salto Caxias	22.40	0.43	22.10	0.43	30	20.30	5.80
Passo Real	28.00	0.23	28.00	0.21	30	23.30	4.70
Itá	35.00	0.36	34.80	0.24	25	21.90	5.00
Passo Fundo	38.10	0.25	38.00	0.17	30	24.50	4.00
Quebra Queixo	37.10	0.50	37.20	0.76	30	22.50	5.80
Tucuruí	8.60	0.46	8.10	0.26	25	21.50	6.50
Manso	24.80	0.17	24.80	0.15	30	24.00	4.60
Furnas	19.50	0.36	19.60	1.80	30	20.40	6.20
Itaipu	8.60	0.17	8.50	0.15	30	21.50	6.30
Boa Esperança	11.10	0.20	11.00	0.19	30	2.00	6.30

The second objective (Experiment 2) was to compare the MAPEs obtained with the proposed method and those obtained and published by the ONS to the model in currently used in the eletric sector (PREVIVAZ model). The results are shown in Table 2. The Wilcoxon statistical test indicates that the results of the proposed method have lower MAPEs for all time series with 95% confidence interval.

The Experiment 3 aimed to compare, for the years 2008, 2009 and 2010, the results obtained for the operational week using the new models that were implemented from 2008, mainly to south region of Brazil. The Wilcoxon statistical test showed that for all metrics the results were better (with 95% confidence level) for the proposed model. Finally the last experiment (Experiment 4) was to calculate the MAPEs with the model proposed for 12 days ahead that is the limit required in PMO (five days to fill the current week and 7 days for the PMO). In Table 2 we can observe the MAPEs for the time series. When performing the statistical test we can conclude that even for 12 days for ahead the errors are lower or equal than those obtained for the operating week with the current models.

6 Conclusion

There isn't a single learning algorithm superior to all others for all problems. Researches in machine learning tries to provide insight into the strengths and limitations of these different algorithms. With this in mind, and background knowledge for a particular problem, it is possible to choose which algorithms will be used to solve that particular problem.

Table 2. Results for the forecasting HS-ELM, PREVIVAZ and HS-ELM for 12 days ahead (values displayed in percent)

Time Series	MAPE (%)					
Models	HS-ELM		PREVIVAZ		HS-ELM-12	
	Mean	STDV	Mean	STDV	Mean	STDV
Jordão	25.00	2.70	40.40	8.90	32.20	0.36
Foz do Areia	26.30	2.90	38.00	3.80	30.10	0.30
Salto Santiago	20.70	0.60	37.10	4.80	28.10	0.20
Salto Caxias	22.10	0.43	–	–	30.20	0.23
Passo Real	27.50	4.60	44.00	5.20	32.90	0.15
Itá	29.20	4.20	67.60	10.30	43.00	0.17
Passo Fundo	24.30	2.60	55.20	11.60	44.00	0.21
Quebra Queixo	34.90	5.80	87.10	12.30	44.40	0.79
Tucuruí	7.60	0.90	12.00	1.70	11.40	0.23
Manso	23.80	1.40	33.30	4.80	29.30	0.12
Furnas	18.00	3.40	26.30	6.40	24.40	0.21
Itaipu	8.90	0.60	15.60	1.50	12.60	0.93
Boa Esperança	10.70	1.70	17.30	2.60	14.30	0.34

We presented, in this paper, the use of a new method for forecasting average daily flows. The proposed method combines the Harmony Search algorithm and the neural network Extreme Learning Machine (HS-ELM). The proposed method obtained results statistically significant, managing to find lower errors than the models that are being used in the electricity sector. The good performance of the model HS-ELM is especially evident in the south region, both seen as difficult when using the previous model PREVIVAZ as compared to new models implanted since 2008.

References

1. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis: Forecasting and Control. Wiley, San Francisco (2008)
2. Hippert, H.S., Pedreira, C.E., Souza, R.C.: Neural Networks for Short-term Load Forecasting: A Review and Evaluation. PAS 16(1), 44–55 (2001)
3. Huang, G.B., Wang, D.H., Lan, Y.: Extreme Learning Machines: a Survey. IJMLC, 1–16 (2011)
4. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Representations by Back-Propagation Errors. Nature 323, 533–536 (1986)
5. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A New Heuristic Optimization Algorithm: Harmony Search. Simulation 76(2), 60–68 (2001)
6. Kim, J.H., Geem, Z.W., Kim, E.: Parameter Estimation of the Nonlinear Muskingum Model using Harmony Search. JAWRA 37(5), 1131–1138 (2001)
7. Geem, Z.W., Kim, J.H., Loganathan, G.V.: Harmony Search Optimization: Application to Pipe Network Design. IJMS 22(2), 125–133 (2002)
8. Lee, K., Geem, Z.H.: A New Structural Optimization Method Based on the Harmony Search Algorithm. Computers and Structures 82(9-10), 781–798 (2004)

9. Geem, Z.W., Tseng, C.-L., Park, Y.-J.: Harmony Search for Generalized Orienteering Problem: Best Touring in China. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 741–750. Springer, Heidelberg (2005)
10. Lee, K., Geem, Z.H.: A New Meta-heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice. CMAME 194(36-38), 3902–3933 (2005)
11. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: Theory and Applications. Neurocomputing 70, 489–501 (2006)
12. Silva, D.N.G., Pacifico, L.D.S., Ludermir, T.B.: An Evolutionary Extreme Learning Machine based on Group Search Optimization. In: IEEE Congress on Evolutionary Computation (CEC), pp. 574–580 (2011)
13. Centro de Pesquisas de Energia Elétrica. Potencialidades de Utilização de Fontes Alternativas para o Atendimento de Energia Elétrica no Interior do Estado de Amazonas. Tech. Report. CEPEL DP/DEA, RJ. 32695/04, 49 p. (2004) (in Portuguese)
14. Guilhon, L.G.F.: Modelo Heurístico de Previsão de Vazões Naturais Médias Semanais Aplicado à Usina de Foz do Areia. Master's Thesis, UFRJ (2003) (in Portuguese)
15. ONS - Operador Nacional do Sistema Elétrico (in Portuguese), http://www.ons.org.br
16. Valença, I., Ludermir, T., Valença, M.: Hybrid Systems to Select Variables for Time Series Forecasting using MLP and Search Algorithms. SBRN, 247–252 (2010)
17. Valença, I., Lucas, T., Ludermir, T., Valença, M.: Selecting Variables with Search Algorithms and Neural Networks to Improve the Process of Time Series Forecasting. IJHIS 8(3), 129–141 (2011)

Towards Efficient Similar Sentences Extraction

Yanhui Gu, Zhenglu Yang, Miyuki Nakano, and Masaru Kitsuregawa

Institute of Industrial Science, The University of Tokyo, Japan
{guyanhui,yangzl,miyuki,kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract. Similar sentences extraction is an essential issue for many applications, such as natural language processing, Web page retrieval, question-answer model, and so forth. Although there are many studies exploring on this issue, most of them focus on how to improve the effectiveness aspect. In this paper, we address the efficiency issue, i.e., for a given sentence collection, how to efficiently discover the top-k semantic similar sentences to a query. The issue is very important for real applications because the data becomes huge and the existing state-of-the-art strategies cannot satisfy the users' performance requirement. We propose efficient strategies to tackle the problem based on a general framework. Extensive experimental evaluations demonstrate that the efficiency of our proposal outperforms the state-of-the-art approach.

Keywords: semantic similarity, top-k, rank aggregation.

1 Introduction

Searching semantic similar sentences is an essential issue because it is the basis of many applications, such as snippet extraction, image retrieval, question-answer model, document retrieval, and so forth [18,3,11,2,13]. From a given sentence collection, this kind of queries ask for those sentences which are most semantically similar to a given one.

The problem can be solved as follows: we firstly measure the similarity score between the query and each sentence in the data collection using the state-of-the-art techniques [7,10,12,14,15], then sort them with regard to the score and finally return the top-k ones. Almost all the previous studies focus on improving the effectiveness aspect (i.e., precision) and the datasets conducted are small. However, when the size of the data collection increases, the scale of the problem will dramatically increase and the state-of-the-art techniques will be impractical. As far as we know, this paper is the first study that aims to address the efficiency issue in the literature. Moreover, most of the previous strategies applied the threshold-based method [7,10,15] that a threshold is set to filter out those dissimilar sentences. However, this threshold is difficult for users to determine. For real applications (e.g., Google), users may prefer the top-k results.

There are mainly four kinds of techniques to measure the similarity between sentences: (1) knowledge-based strategy [15,12]; (2) corpus-based strategy [7]; (3) syntax based strategy [10,7]; and (4) hybrid strategy [7,10]. All of these

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 270–277, 2012.

works, however, is time consuming when testing the candidates for top-k similar sentences extraction. To tackle this issue, we introduce efficient strategies to evaluate as few candidates as possible. Moreover, we aim to progressively output the top-k results, i.e., the top-1 result should be output almost instantly, then the top-2 and more results will be obtained as the execution time becomes longer. This satisfies the requirement of the real applications [5]. As such, these issues are the challenges of the paper, which have not been studied before. Our contributions of this paper are listed as follows:

We propose to tackle the efficiency issue for searching top-k semantic similar sentences, which is different from previous works that focus on the effectiveness aspect. Based on the most comprehensive work [7], we introduce the optimization techniques and improve the efficiency. For each similarity measurement, we introduce a corresponding strategy to minimize the number of candidates to be evaluated. A rank aggregation method is introduced to progressively obtain the top-k results when assembling the features. We conduct experiments and evaluate the performance of the proposed strategies. The results show that the proposed strategies outperform the state-of-the-art method.

2 Problem Statement

The issue we aim to tackle is to extract the top-k similar sentences to a query. Formally, for a query sentence Q, finding a set of k sentences P in a given sentence collection S which are most similar to Q, i.e., $\forall p \in P$ and $\forall r \in (S - P)$ will yield $sim(Q, p) \geq sim(Q, r)$.

To measure the similarity $sim(Q, P)$ between two sentences, we apply the state-of-the-art strategies by assembling multiple similarity metric features together [10,7]. Because we focus on tackling the efficiency issue in this paper, several representative features are selected based on the framework in [7]. Note that a sentence is composed of a set of words and therefore, the similarity score between two sentences is the overall scores of all the word pairs whose components belong to each sentence, respectively [7]. As such, we introduce the representative word similarity in the next section.

2.1 Similarity Measurement Strategies

String-Based Similarity. String similarity measures the difference of syntax between strings. An intuitive idea is that two strings are similar to each other if they have enough common subsequences (i.e., LCS [6]). We focus on three representative string similarity measurement strategies, i.e., NLCS, NMCLCS_1 and NMCLCSn[1] which are denoted as Sim_{NLCS}, Sim_{NMCLCS_1} and $Sim_{NMCLCSn}$ [7]. The following are the formulas:

$$Sim_{StringStrategy}(w_i, w_j) = \frac{length(StringStrategy)^2}{length(w_i)length(w_j)} \quad (1)$$

[1] NLCS: Normalized Longest Common Substring, NMCLCS_1: Normalized Maximal Consecutive LCS starting at character 1, NMCLCSn: Normalized Maximal Consecutive LCS starting at any character n [7].

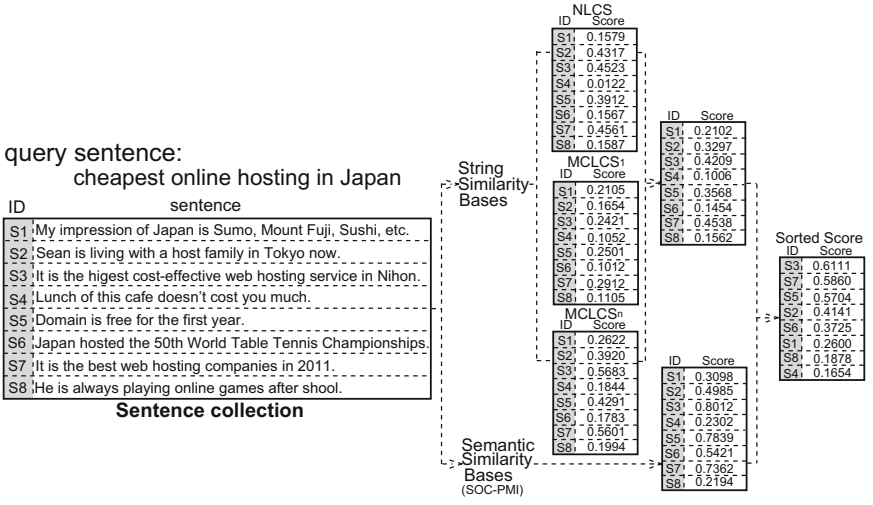


Fig. 1. The implementation of the top-k similar sentences searching framework

Here StringStrategy are NLCS, MCLCS_1 and MCLCS_n.

Corpus-Based Similarity. Corpus-based similarity measurement is to recognize the degree of similarity between words using large corpora [9]. There are several kinds of strategies: PMI [16], LSA [8], HAL [1], chi-square, and so forth. In this paper, we use SOC-PMI (Second Order Co-occurrence PMI) [7] which employs PMI while taking into account important neighbors in the context windows of the two words. The intuitive idea is that the neighbors have abundant semantic context and should be considered. PMI [16] is defined as follows:

$$f^{pmi}(w_i, w_j) = log_2 \frac{f(w_i, w_j) \times m}{f(w_i) f(w_j)} \tag{2}$$

where $f(w_i)$ is the frequency of the word w_i in the corpus, and m is the size of the corpus. Eq. 2 is used to calculate the similarities between words, and then high PMI scores are aggregated to obtain the final SOC-PMI score [7].

2.2 A General Framework

To measure the overall similarity between two sentences, a general framework is presented by incorporating the main similarity features. As far as we know, [7] is the most comprehensive approach which incorporates several representative similarity metrics (i.e., string similarity and semantic similarity[2]). The string similarity is further composed of three different measurement, i.e., NLCS, NMCLCS_1 and NMCLCSn. Fig. 1 illustrates the general framework for searching the top-k similar sentences. Due to lack of large labeled data, the existing state-of-the-art assembling techniques commonly set the weight values arbitrarily

[2] The common word order similarity is neglected here because [7] has demonstrated that it has no influence on the overall score.

[7,10,12,14]. We apply the same strategy (i.e., $weight_{string}$=$weight_{semantic}$=1/2, $weight_{NLCS}$=$weight_{NMCLCS_1}$=$weight_{NMCLCS_n}$=1/3). While obtaining optimal values of these weights is certainly an interesting issue, it is out of the scope of this paper. It is very time consuming to test every candidate especially when the data collection is large. Therefore, efficient strategies on searching top-k semantic similar sentences are necessary.

3 Proposed Approaches

We propose efficient strategies for extracting the top-k similar sentences. The key idea is that by building appropriate index in the preprocessing, we only need to test a small part of candidates in the whole data collection.

3.1 Optimization on String-Based Similarity

We employ NLCS, NMCLCS_1, NMCLCSn as our string-based similarity features [7].

• **NLCS**

The basic idea for improving the efficiency of similar word extraction, is to test as few candidates as possible. To address this issue, in the preprocessing we need to built an effective index which facilitates the candidate test process. A well known technique is n-gram (or q-gram [17]) model, which is utilized in our framework. Moreover, because our purpose is to search the top-k similar words, we can further improve the efficiency based on the property of the similarity.

From Eq. 1 we know that NLCS is based on two factors: (1) length of LCS; (2) length of the candidate w_i (length of the query is not important because we cannot know it in the preprocessing). The similarity increases when $length(LCS)$ increases or $length(w_i)$ decreases. Therefore we have the following lemma.

Lemma 1 (Lower Bound of Gram Length). *Let P be the top-k similar word to the query Q so far. Q and P have the NLCS score as τ_{top-k}. We denote the set R be the untested words and the gram set G be the untested grams. If $\forall g \in G$ and $\forall r \in R$, $|g_r| < |Q| \cdot \tau_{top-k}$, then the top-k similar words w.r.t. string similarity score have been found.*

Proof (Sketch of Proof). (Proof by Contradiction) Assume there is one candidate word r with gram $|g_r| < |Q| \cdot \tau_{top-k}$ in R and it is ranked in the top-k list. Then we have $Sim(Q, r) \geq \tau_{top-k}$, $\frac{|g_r|^2}{|Q||r|} \geq \tau_{top-k}$, $|g_r|^2 \geq |Q| \cdot |r| \cdot \tau_{top-k}$, $|g_r| \geq |Q| \cdot \tau_{top-k}$, which contradicts the assumption. □

This lemma tells us that we can sort the grams in descending order of their lengths in the index. Moreover, for each $gram$, we sort all the corresponding words (which include this gram) in ascending order of their lengths. To search the top-k similar words, we start from the word list which has the longest gram. In the list, we first test shorter words and then the longer ones.

• **NMCLCS$_1$**
This similarity measures the maximal common consecutive prefix substrings of two strings. Similar to NLCS, from Eq. 1, we can see that NMCLCS$_1$ is determined by two factors: length of NMCLCS$_1$ and length of candidate word. NMCLCS$_1$ increases when the length of NMCLCS$_1$ increases or the length of the candidate decreases. Therefore, we can build a gram index that all the grams are in descending order of their lengths. For each *gram*, the related words are in ascending order of their lengths. The difference for indices between NLCS and NMCLCS$_1$ is that the grams for the latter are only a part of the former (i.e., start at the beginning of each string). The lower bound used to terminate is $|Q| \cdot \tau_{top-k}$. The computation of τ_{top-k} is based on Eq. 1. The algorithm of searching top-k similar words for NMCLCS$_1$ is very similar to that for NLCS. We need to modify the index strategy in the algorithm.

• **NMCLCSn**
NMCLCSn measures the maximal common consecutive substring which can starts at any position n. The index for NMCLCSn is similar to that for NMCLCS$_1$. The only difference is the strategy for decomposing words into grams. NMCLCSn parses words into grams which can start at any positions. The lower bound used to terminate is $|Q| \cdot \tau_{top-k}$, where τ_{top-k} is calculated by Eq. 1. The algorithm for NMCLCSn is similar to that for the above mentioned string similarities, with modification according to the definition of NMCLCSn.

3.2 Optimization on Corpus-Based Similarity

We apply SOC-PMI [7] as the corpus-based similarity evaluator. SOC-PMI linearly aggregates the top large PMI values (i.e., Eq. 2) of each pair words with their neighbors. We introduce an efficient technique [19].

Lemma 2 (Upper Bound of Word Frequency). *Let Q be the query word and P be the top-k similar word so far. Q and P have the similarity score τ_{top-k}. If $\forall r \in R$, $f(r) > \frac{m}{2^{top-k}}$, the top-k similar words have been found. Here R is the remaining untested words and m is the size of the corpus.*

Based on this lemma, we sort all the candidates in ascending order of their frequencies in the preprocessing and measure the similarity while querying one by one. When we find the current frequency of candidate word is large than $\frac{m}{2^{top-k}}$, we can terminate the process and avoid to test the remaining candidates.

3.3 Sentence Similarity Computation

We have introduced how to measure the similarity between words. Here we take illustrate how to obtain the similarity between sentences. Let P ="*Cheapest online hosting in Japan.*", Q="*Domain is free to use for the first year*". After removing all stop words and lemmatizing, we obtain P={*cheap online hosting Japan*} and Q={*domain free use year*}. Through similarity measuring on each combination of word pair for the sentences, we construct a similarity matrix

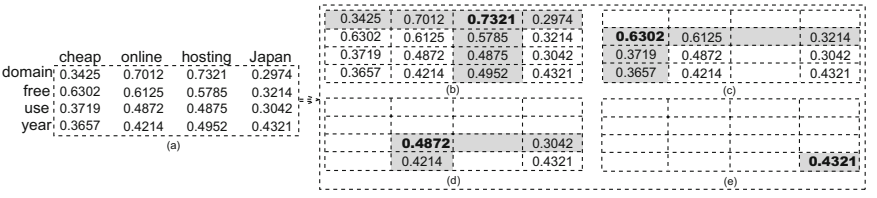


Fig. 2. Similarity measurement between two sentences

which is illustrated in Fig. 2 (a). Each element represents the overall similarity score of the word pair, i.e., aggregation of string-based and corpus-based similarities. To obtain the score between sentences, we first find the maximal-valued element (i.e., representative word). Then the related row and column which includes this element are removed. This process is recursively executed and finally all the similarities of the representative words have been extracted (i.e., Fig. 2 (b-e)). The similarity between sentences is computed as follows: $S(P,Q) = \frac{\sum_{i=1}^{|min(|P|,|Q|)|} \rho_i(|P|+|Q|)}{2|P|*|Q|}$, where ρ_i is the value of the representative word of round i. Therefore we have $S(P,Q)=\frac{(0.7321+0.6302+0.4872+0.4321)(4+4)}{2*4*4}=0.5704$.

3.4 Assembling Similarity Features

We introduce an efficient assembling approach to hasten the process of searching top-k similar sentences [4]. To illustrate the method, we use the concrete example (i.e., Fig. 1) to explain, as shown in Fig. 3. In the first iteration, we obtain the top-1 sentences with their scores in the features, i.e. S7: 0.4358 and S3: 0.8012. Next the threshold is computed, i.e., 0.6185 ($w_A \cdot Sim_{A_i} + w_B \cdot Sim_{B_i}$, where w is the weight value[3]). Because the overall scores of the two accessed sentences are smaller than the threshold, no result is output in this round. In the second iteration (i.e., testing on top-2 sentences in both lists), the threshold is computed as 0.6024. Because the score of S3 is 0.6111 which is larger than the threshold, S3 can be output immediately. We can see that for this example, the performance of obtaining the top-1 result is very efficient because we do not need to evaluate many candidates. The remaining processes are executed in a similar way.

In addition to the above mentioned feature aggregation, we apply the threshold-based strategy in assembling different string similarities (i.e., NLCS, NMCLCS$_1$, and NMCLCS$_n$), and assembling words into sentence to obtain the top elements.

4 Experimental Evaluation

We conducted experiments using 16-core Intel(R) Xeon(R) E5530 server which runs Debian 2.6.26-2. All the algorithms were written in C and compiled by GNU gcc. The baseline is implemented based on the state-of-the-art [7].

[3] The weights are set to 1/2, respectively, as explained in Section 2.

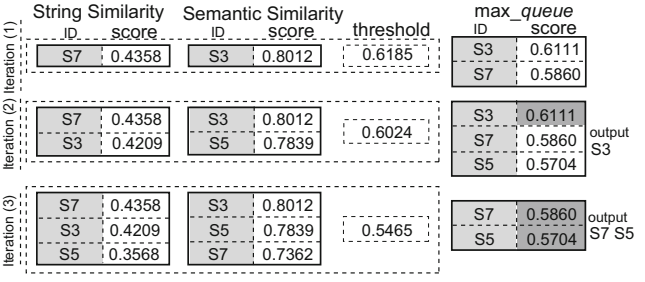
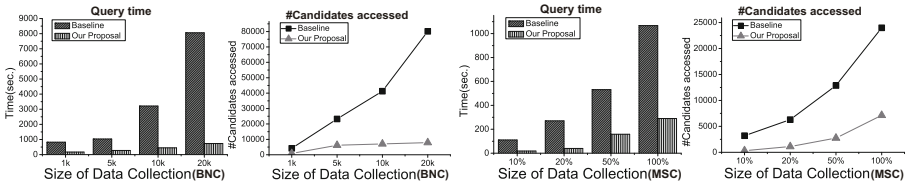
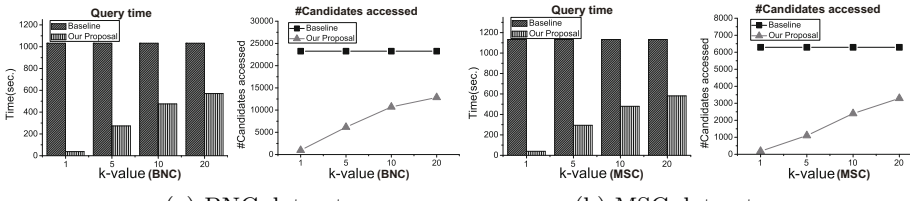


Fig. 3. Efficient searching top-k semantic sentences framework



(a) BNC dataset (b) MSC dataset

Fig. 4. Effect on size of data collection



(a) BNC dataset (b) MSC dataset

Fig. 5. Effect of k-value

Effect of Size of Data Collection. The evaluated datasets come from BNC[4] and MSC[5]. We randomly extracted 1k, 5k, 10k, 20k sentences from BNC and divided MSC into different size, i.e.,10%, 20%, 50%, 100%, as our datasets. Fig. 4 shows the top-5 results under 10 randomly selected queries. We can see that our proposal is much faster than the baseline for both datasets because the proposal largely reduces the number of candidates tested. When the size of data collection increases, the query time of our proposal increases linearly and it scales well.

Effect of k. We randomly chose 10 queries from the collections. The size of BNC was fixed to 5k and the whole MSC was used. From Fig. 5 we see the baseline needs to access all candidates and the query time is the same for all situations. For our proposal, the top-1 can be returned almost instantly. When k increases, the query time increases because more candidates need to be evaluated.

[4] http://www.natcorp.ox.ac.uk/

[5] Microsoft Research Paraphrase Corpus. It contains 5800 pairs of sentences.

5 Conclusion

In this paper, we proposed to tackle the efficiency issue of searching top-k similar sentences which has not been studied before. Several efficient strategies are introduced to test as few candidates as possible in the process. The comprehensive experiments demonstrates the efficiency of the proposed techniques.

References

1. Burgess, C., Livesay, K., Lund, K.: Explorations in context space: words, sentences, discourse. Discourse Processes (1998)
2. Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., Silvestri, F.: Caching query-biased snippets for efficient retrieval. In: EDBT (2011)
3. Cui, H., Sun, R., Li, K., Kan, M.-Y., Chua, T.-S.: Question answering passage retrieval using dependency relations. In: SIGIR (2005)
4. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. In: PODS (2001)
5. Hellerstein, J.M., Avnur, R., Chou, A., Hidber, C., Olston, C., Raman, V., Roth, T., Haas, P.J.: Interactive data analysis: The control project. Computer 32 (1999)
6. Hirschberg, D.S.: A linear space algorithm for computing maximal common subsequences. Commun. ACM (1975)
7. Islam, A., Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data (2008)
8. Landauer, T., Dumais, S.: A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review (1997)
9. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Processes (1998)
10. Li, Y., McLean, D., Bandar, Z.A., O'Shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Transaction on Knowledge and Data Engineering (2006)
11. Metzler, D., Dumais, S.T., Meek, C.: Similarity Measures for Short Segments of Text. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 16–27. Springer, Heidelberg (2007)
12. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI (2006)
13. Radlinski, F., Broder, A., Ciccolo, P., Gabrilovich, E., Josifovski, V., Riedel, L.: Optimizing relevance and revenue in ad search: a query substitution approach. In: SIGIR (2008)
14. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: WWW (2006)
15. Tsatsaronis, G., Varlamis, I., Vazirgiannis, M.: Text relatedness based on a word thesaurus. Journal of Artificial Intelligence Research (2010)
16. Turney, P.D.: Mining the web for synonyms: Pmi-ir versus lsa on toefl. In: EMCL (2001)
17. Ukkonen, E.: Approximate string-matching with q-grams and maximal matches. Theoretical Computer Science 92(1) (1992)
18. Wei, F., Li, W., Lu, Q., He, Y.: Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization. In: SIGIR (2008)
19. Yang, Z., Kitsuregawa, M.: Efficient searching top-k semantic similar words. In: IJCAI (2011)

Codebook Quantization for Image Classification Using Incremental Neural Learning and Subgraph Extraction

Ye Tang[1], Yu-Bin Yang[1], Yang Gao[1], Yao Zhang[2], and Ying-Chun Cao[1]

[1] State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093, China
[2] Jinling College, Nanjing University, Nanjing 210089, China
yangyubin@nju.edu.cn

Abstract. This paper proposes a fast, incremental codebook quantization algorithm for image classification consisting of a fast codebook graph learning algorithm using incremental neural learning, and a subgraph-based coding method. Comparing with the algorithms based on classic Bag-of-Features (BOF) model, it holds the following advantages: 1) it learns codebook fast and effectively simply using a few training data; 2) it models relationships among visual words to guarantee higher discriminative power; 3) it automatically learns codebook with appropriate size. The above characteristics make our method more suitable for handling large-scale image classification tasks. Experimental results on Caltech-101 and Caltech-256 datasets demonstrate that the proposed algorithm achieves better performance while decreasing the computational cost remarkably.

Keywords: Bag-of-Features, codebook quantization, codebook learning, subgraph coding, image classification.

1 Introduction

In recent years, Bag-of-Features (BOF) model [1] has been one of the best methodologies for image classification, which is an important and challenging problem in computer vision. In BoF model, firstly, an image is partitioned into local patches. Then, low-level feature descriptors (e.g. SIFT) extracted from the patches are quantized into visual words which constitute a codebook. After that, an image is described by a histogram representation, and then a discriminative or generative classifier, e.g. SVM or PLSA, is adopted for image classification. The BoF model treats an image as a "bag" of local features, which discards the spatial information of the image. To overcome this problem, a simple but effective extension of BoF model has been proposed, i.e. spatial pyramid matching (SPM) [2]. SPM divides the image into increasingly finer spatial regions and computes histograms of local descriptors for every region, and then concatenates all histograms to form a long vector representation of the image. This method has obtained promising performance on various benchmark datasets. Specifically, it is necessary to determine a coding method which quantizes local features into visual words using the learned codebook. In traditional SPM model, K-means vector quantization (VQ) is employed for local descriptors coding, which finds the nearest word to the feature vector and uses it as the coding result. This hard assignment causes a large

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 278–285, 2012.

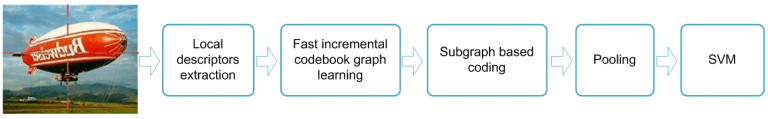


Fig. 1. The flowchart of image classification procedure using the proposed algorithm

quantization error and the loss of useful information for classification. Unlike hard assignment, soft assignment [3,4,5,6] associates the coding result with the contribution of several words and generally outperforms hard assignment. Kernel codebook (KC) [3], ScSPM [6] and Locality-constrained Linear Coding (LLC) [4] are the three most widely used soft assignment coding methods. However, most existing methods still bear the following drawbacks: (1) they cannot handle very large training datasets, or dynamic training data changing over time; (2) they ignore the relationships among visual words; (3) they require a pre-defined codebook size. To address the above issues, we propose a fast, incremental codebook graph learning algorithm and a subgraph-based coding method for image classification. The former, inspired by the studies on self-organizing incremental neural network (SOINN) [7], can process large datasets and models the relationships among visual words; and the latter considers both relationships among visual words and locality of coding results, leading to better performance. The flowchart of image classification procedure using the proposed algorithm is shown in Figure 1.

The rest of this paper is organized as follows. Section 2 describes SOINN and discusses two recently proposed extensional methods of BoF model. The proposed algorithm is elaborated in Section 3. Section 4 provides our experimental results on Caltech-101 and Caltech-256 datasets. Finally, we conclude this paper in Section 5.

2 Related Work

2.1 Self-Organizing Incremental Neural Network

Self-organizing Incremental Neural Network(SOINN) is an online unsupervised classification and topology learning mechanism [7], of which the structure is a two-layer competitive neural network. Each node records its own accumulated error and each edge has its own age. The initial state contains two nodes with weight vectors chosen randomly from input data and the empty connection set. Then, for every new input data $x \in R^n$, it searches node set A to find out the first nearest node s_1 and second nearest node s_2:$s_1 = argmin_{c \in A}\|x - W_c\|$, $s_2 = argmin_{c \in A - \{s_1\}}\|x - W_c\|$, where W_c is the weight vector of node c. If the distances between input data and s_1 or s_2 are greater than pre-defined thresholds, a new node will be created, whose weight vector is identical to the value of the input data. Otherwise, SOINN considers the input data as known information, and the weight vectors of the first nearest node W_{s_1} and its direct neighbors will be updated according to the value of the input data. Meanwhile, an edge connecting the first and second nearest node will be created, if it does not exist; otherwise, its age will be reset to zero. As processing input data incrementally, SOINN removes old edges and deletes isolated nodes considered as noises. In addition, it divides the nodes of which the accumulated error are large.

2.2 Online Codebook Learning and Graph-Based Coding Method

Mairal et al. [8] proposes an online codebook learning algorithm aiming to minimize the expected cost instead of the empirical cost. Based on first-order stochastic gradient descent, the algorithm scaled up gracefully to large-scale datasets. In the most recent research [9], the relations among visual words were considered. By constructing the codebook graph wherein a visual word is linked with other words and adopting a coding method which takes advantage of the edge information of two words, the model contains richer and more discriminative information than BoF model to achieve higher classification accuracy. The online codebook learning method ignores the relations among words, while the graph-based coding method employs a "batch" procedure to construct the codebook graph, and only utilizes the edges of the graph to reflect and describe the properties of codes. We intend to integrate the advantages of the above methods into a unified framework for image classification.

3 The Proposed Algorithm

3.1 Fast Incremental CodeBook Graph Learning

Let $X = [x_1, x_2, ..., x_N] \in R^{D\times N}$ denote a set of SIFT descriptors in a D-dimensional feature space. we expect to solve the following problem and capture the structural information of visual words:

$$\begin{aligned} f(X,B) = \min_{B,\alpha} \tfrac{1}{N} \sum_{i=1}^{N} ||x_i - B\alpha_i||_2^2 + \lambda ||s_i \odot \alpha_i||_2^2 \\ s.t.\ \ ||b_j||_2^2 \le 1\,, \forall j = \{1, ..., K\} \end{aligned} \tag{1}$$

where $\odot$ denotes the element-wise multiplication, $B = [b_1, ..., b_K] \in R^{D\times K}$ denotes the codebook, K is a variable representing the appropriate codebook size, $\alpha = [\alpha_1, ..., \alpha_N] \in R^{K\times N}$ denotes the coding results and $s_i = [s_{i1}, ..., s_{iK}] \in R^K$ denotes the distance constraint of a coding result. Specifically,

$$s_{ij} = \begin{cases} \exp(\frac{||x_i - b_j||_2^2}{\gamma}), & ||x_i - b_j||_2^2 \le \gamma \\ \infty\ , & o.w. \end{cases} \tag{2}$$

the distance constraint s_i ensures that the coding coefficient is non-zero only when its relative word belongs to the neighboring region of input data x_i. γ is used for controlling the radius of the neighboring region.

In order to solve the above problem, we present an improved single-layer SOINN, which is faster and more robust to noisy data in training images. We then combine it into a codebook learning framework. It is worth to note that our incremental codebook graph learning is not bound to obtain the global optimal of the objective function $f(X, B)$ in Eq. (1). In fact, it leads to a satisfied solution with good generalization. Furthermore, our algorithm automatically determines the appropriate number of clusters according to the data's distribution, and the learned codebook is represented as a graph, which can describe relationships among visual words. This allows us to design more effective coding method by using structural information among visual words. Our algorithm is summarized in **Algorithm 1**.

Algorithm 1. Fast Incremental Codebook Graph Learning

Input: Local Feature Training Set $X \in R^{D \times N}$; Remove Node Time λ;
Dead Age age_{MAX}; Proportion Parameter c, Predefined Similarity Threshold T_p
Output: Codebook Graph G(V,E)

1: Initialize node set V to contain two nodes with weight vectors chosen randomly from X: $V \leftarrow \{v_1, v_2\}$, set their winner time to 0: $M_{v_1} \leftarrow 0, M_{v_2} \leftarrow 0$. Initialize connection set E as an empty set: $E \leftarrow \emptyset$.
2: Input new local features $x \in X - V$.
3: Find the nearest node (the winner) v_{w1} and the second nearest node (the second winner) v_{w2} according to their Euclidean distances to the input feature vector x.
4: Compare the distances with the minimum values of adaptive similarity threshold and predefined similarity threshold:
5: **if** $||x - v_{w1}||_2^2 > \min(T_{v_{w1}}, T_p)$ or $||x - v_{w2}||_2^2 > \min(T_{v_{w2}}, T_p)$ **then**
6: Create a new node whose weight is equal to x, set its winner time as 0, and add it to V; goto step 2.
7: Compute adaptive similarity threshold T_i using the following equation:

$$T_i = \begin{cases} \max\limits_{j \in N_i} ||w_i - w_j||_2^2, & N_i \neq \emptyset \\ \min\limits_{j \in V \setminus \{i\}} ||w_i - w_j||_2^2, & N_i = \emptyset \end{cases} \tag{3}$$

where N_i denotes the set of neighboring nodes of node i.
8: Increase the age of all edges linked with v_{w1} by 1.
9: **if** no edge exists between v_{w1} and v_{w2} **then**
10: Create an edge between v_{w1} and v_{w2}, set its age as 0, and add it to E.
11: **else**
12: Set the age of the existing edge as 0.
13: Delete the edges whose ages are greater than age_{MAX}.
14: Add 1 to the winner time of winner v_{w1}: $M_{v_{w1}} \leftarrow M_{v_{w1}} + 1$
15: Adapt the weight vectors of the winner and its direct topological neighbors using the following equations:

$$w_{v_{w1}} = w_{v_{w1}} + \frac{1}{M_{v_{w1}}}(x - w_{v_{w1}}) \tag{4}$$

$$w_{v_{wi}} = w_{v_{wi}} + \frac{1}{M_{v_{w1}} \exp(||w_{v_{w1}} - w_{v_{wi}}||_2^2)}(x - w_{v_{wi}}) \tag{5}$$

16: Delete the noisy nodes:
17: **for** each $v_i \in V$ **do**
18: **if** v_i has two neighbors, and its winner time is less than c times of the mean winner time of all nodes **then**
19: Delete v_i.
20: **else if** v_i has less than two neighbors **then**
21: Delete v_i.
22: If the entire training set X has been processed, stop learning; else goto step 2.
23: **return** Codebook Graph G(V,E)

The winner time of a node records the time it takes to be a winner, and the predefined similarity threshold T_p is used to insert new nodes when the distance between two randomly initialized nodes is too large. The algorithm iteratively processes the

entire training set and constructs a network, of which the nodes represent visual words and the edges capture the relationships among visual words. The adaptively competitive learning mechanism ensures that the preserving nodes and edges appropriately estimate the distribution and structural information of input data, respectively.

3.2 Subgraph-Based Coding Method

Given the learned codebook graph, we then present a new coding method which emphasizes both locality and topological structure of visual words contributing to the coding result. For a local feature x, we search k nearest words $w_1, ..., w_k$ and their direct topological neighbors $w_{11}, ..., w_{1m_1}, ..., w_{km_k}$. Then, the subgraph consisting of these words is used for coding. By introducing parameter β to control the maximum number of each word's neighbors, we can limit the size of subgraph easily. Finally, these words are voted via an appropriate weighting method. In order to assign the closer words with larger weights, we sort the words in the subgraph in ascending order, and compute the corresponding weights according to Eq.(6), similar to the formulation developed in [5]. The algorithm is summarized in **Algorithm 2**.

4 Experimental Results

In this section, we evaluate our algorithm on Caltech-101 [1] and Caltech-256 datasets [2].

Algorithm 2. Subgraph-based Coding Method

Input: Local Feature $x \in R^N$; Codebook Graph G(V,E), ($V \in R^M$);
Number of local neighbors k; Maximum topological neighbor number β
Output: Coding Result $S \in R^K$

1: Search the k nearest words of input feature x from the node set V
2: Sort them in ascending order: $w_1, ..., w_k$
3: **Extract Subgraph** $G^{'}(V^{'}, E^{'})$:
4: **for** i = 1 to k do **do**
5: Search graph G(V,E), and find the topological neighbors of w_i: $w_{i1}, ..., w_{im_i}$(m_i is the minimum of β and the number of w_i's neighbors)
6: Remove repeating words from these neighbors, and construct subgraph $G^{'}(V^{'}, E^{'})$ according to the chosen words
7: Sort the words in $G^{'}(V^{'}, E^{'})$ in ascending order according to their distances to the input feature vector x : $w'_1, ..., w'_{|V'|}$
8: Vote to $w'_1, ..., w'_{|V'|}$ with different weights calculated as:

$$S_{Index(w'_j)} = \frac{2^{|V'|}}{2^j \times d(x, w'_j)} \quad (j \in 1, ..., |V'|) \tag{6}$$

$$Index(w'_j) \ : \ the \ index \ of \ w'_j \ in \ S; d(x, w'_j) \ : \ the \ distance \ between \ x \ and \ w'_j$$

9: **return** Coding Result $S \in R^K$

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[2] http://www.vision.caltech.edu/Image_Datasets/Caltech256/

4.1 Experimental Settings

We follow the common experimental settings, that is, training on 15 images per category on both datasets and testing on at most 50 and 20 images per category for Caltech-101 and Caltech-256, respectively. All experiments are repeated for 10 times to obtain stable performance. Our implementation uses a single descriptor type, i.e. SIFT. The SIFT are extracted from 16×16 pixel patches with a step-size of 6 pixels for every image, in which large images are resized to fit inside a 256×256 box. A subset is randomly sampled as the training set for codebook learning. In our experiments, the subset containing 200,000 features can deliver adequately good results. For fast incremental codebook learning, the parameters are set as: $\lambda = 300$, $age_{MAX} = 50$, $c = 0.001$, $T_p = 128$ and the learned codebook size are 826 and 686 for Caltech-101 and Caltech-256, respectively. We test various values by setting $\lambda = 100, 200, 400$, $age_{MAX} = 100, 200, 300$, $c = 0.01, 0.1, 1$ and $T_p = 256$, but the change of results are not very significant. For the graph-based coding method, we set $k = 6$ and $\beta = 4$. It guarantees the locality and structure of coding results. To form the description of the whole image, we adopt average pooling [2] and the overlapping spatial pyramid in [10], as illustrated in Figure 2. In order to further improve the consistence of image features, we extract features and encode both of the original image and its sobel gradients image. We adopt BSVM[3] and one-vs-all approach for multi-class image classification.

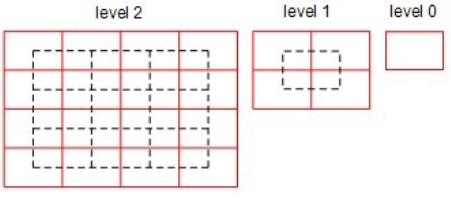


Fig. 2. The overlapping pyramid of three levels

4.2 Codebook Learning Time

We compare the training time of our fast incremental codebook learning algorithm with SPM on Caltech-101 dataset. Since SPM is significant faster than ScSPM and its other extensions based on sparse coding, we only need to compare our method with SPM. Since the learned codebook size obtained by our algorithm is 432 and 826 respectively, for a fair comparison, we set the codebook size as 400 and 800 for SPM correspondingly. All algorithms are implemented using C++ and run on a standard desktop PC with a 3.10GHz i5-2400 CPU. As shown in Figure 3, the training cost of our algorithm is only one fifth or one sixth of that of SPM, and the advantage of our method tends to be more obvious as the codebook size increases. For large-scale datasets, large codebooks are generally important for obtaining high classification accuracy. Therefore, our method is more suitable for image classification tasks in large-scale datasets.

[3] http://www.csie.ntu.edu.tw/~cjlin/bsvm/

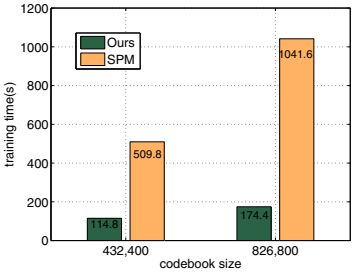


Fig. 3. Comparison of codebook training time (unit: sec.)

4.3 Classification Accuracy

We compare our algorithm with some recently presented algorithms in terms of average classification accuracy on these datasets. Detailed comparison results are shown in Table 1. For Caltech-101, our result significantly outperforms all baseline methods compared. More specifically, our method outperforms SPM by 20%, which is the fastest codebook learning algorithm in all baseline methods. While our codebook learning time is much less than that of SPM. Furthermore, our result is even better than ScSPM [6], which employed sparse coding method and obtained the state-of-the-art classification performance on Caltech-101 dataset before. Although the improvement is only recorded as 1.8%, ScSPM requires to solve the computationally expensive l_1-norm optimization problem. Herewith, it is also the slowest codebook learning algorithm, despite of its highest accuracy, in all baseline methods. For Caltech-256, our method outperforms ScSPM by 3.7% and is slightly better than the method in [11]. It is worth to mention that our learned codebook size is relatively small (i.e. 686), therefore we believe that the performance can be further improved if we could learn a large codebook by adjusting the related parameters carefully. Comparing with graph based coding method in [9], which only utilizes the edges for coding, our approach also achieves better performance. It suggests that our subgraph-based coding method maintains more discriminative information in the image representation.

Table 1. Average classification accuracy comparison on Caltech-101 and Caltech-256

Dataset	Caltech-101	Caltech-256
Lazebnik et al. [2]	56.40	–
Yang et al. [6]	67.00	27.73
Boiman et al [12]	65.00	–
Griffin et al [11]	59.00	28.30
Huang et al [9]	66.88	–
Proposed Method	**68.25**	**28.80**

5 Conclusion

In this paper, we present a fast, incremental codebook learning algorithm and a new subgraph-based coding algorithm. Furthermore, we combine these two algorithms into

a unified framework for image classification. The experimental results on Caltech-101 dataset and Caltech-256 dataset show that our algorithm can not only significantly improve computational efficiency, but also achieve competitive performance to the state-of-the-art classification accuracy. More specifically, our algorithm is able to learn in an online way with no need of pre-defined codebook size, and models relationships among visual words for better performance, which makes it more effective and efficient for handling large-scale datasets in many real applications.

Acknowledgments. We would like to acknowledge the supports from Program for New Century Excellent Talents of MOE China (Grant No. NCET-11-0213), National 973 Program of China (Grant No. 2010CB327903), the National Science Foundation of China (Grant Nos. 61035003, 61021062, 60975043,60875011), and the Key Program of National Science Foundation of Jiangsu, China (Grant Nos. BK2010054, BK2011005, BE2010638).

References

1. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV, pp. 1470–1477 (2003)
2. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
3. van Gemert, J., Veenman, C.J., Smeulders, A., Geusebroek, J.M.: Visual word ambiguity. TPAMI, 1271–1283 (2010)
4. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T.S., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR, pp. 3360–3367 (2010)
5. Jiang, Y.G., Ngo, C.W., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: ACM Intl. CIVR, pp. 494–501 (2007)
6. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classificaiton. In: CVPR, pp. 1794–1801 (2009)
7. Furao, S., Hasegawa, O.: An incremental network for on-line unsupervised classification and topology learning. Neural Networks 19(1), 90–106 (2006)
8. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: ICML, pp. 689–696 (2009)
9. Huang, Y., Huang, K., Wang, C., Tan, T.: Exploring relations of visual codes for image classification. In: CVPR, pp. 1649–1656 (2011)
10. Wu, J., Rehg, J.M.: Beyond the euclidean distance: Creating effective visualcodebooks using the histogram intersection kernel. In: ICCV, pp. 630–637 (2009)
11. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
12. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR, pp. 1–8 (2008)

Visualization of Predictive Distributions for Discrete Spatial-Temporal Log Cox Processes Approximated with MCMC

David Rohde[1], Jonathan Corcoran[2], Gentry White[3], and Ruth Huang[2]

[1] Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil
[2] School of Geography, Planning and Environmental Management, University of Queensland, Brisbane Australia
[3] Institute for Social Science Research, University of Queensland, Brisbane Australia

Abstract. An important aspect of decision support systems involves applying sophisticated and flexible statistical models to real datasets and communicating these results to decision makers in interpretable ways. An important class of problem is the modelling of incidence such as fire, disease etc. Models of incidence known as point processes or Cox processes are particularly challenging as they are 'doubly stochastic' i.e. obtaining the probability mass function of incidents requires two integrals to be evaluated. Existing approaches to the problem either use simple models that obtain predictions using plug-in point estimates and do not distinguish between Cox processes and density estimation but do use sophisticated 3D visualization for interpretation. Alternatively other work employs sophisticated non-parametric Bayesian Cox process models, but do not use visualization to render interpretable complex spatial temporal forecasts. The contribution here is to fill this gap by inferring predictive distributions of Gaussian-log Cox processes and rendering them using state of the art 3D visualization techniques. This requires performing inference on an approximation of the model on a discretized grid of large scale and adapting an existing spatial-diurnal kernel to the log Gaussian Cox process context.

Keywords: non-parametric Bayesian Inference, Markov chain Monte Carlo, Visualization, Spatial-Temporal Cox Processes, Geographical Information Systems.

1 Introduction

There are considerable applications for applying pattern recognition techniques to exploring spatial-temporal datasets of incidents. Applications include modelling the occurrence of disease outbreaks, modelling the observations of new species, or the temporal occurrence of incidents such as coal mine disasters. Another pertinent example is the occurrence in space and in the hour of day (diurnal) of urban fires in Australia which we use as a case study in this paper. The

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 286–293, 2012.

goal of such a spatial-diurnal analysis can be used to evaluate the consequences of different operational decisions of the fire service and to direct attention to preventative action. While a formal model may be employed for the occurrence of fire incidents, the available decisions and their utility under different hypothetical outcomes is usually not formalized. As such it can be preferable to use advanced three dimensional visualizations of the output using 3D rendering techniques such as the iso-surface, cut planes and volume rendering.

Statistical models of incidence are complex entities as they are doubly stochastic models, i.e. identifying the probability mass function of the count of incidence in a region involves computing two integrals instead of the usual one. These statistical models are known as point processes or Cox processes [5].

Existing research on forecasting point processes follows two main paths. The first path is to use simple statistical approaches based on kernel smoothing applied to a point process in essentially the same way as applied to problems of probability density estimation [7]. If a new spatial-diurnal kernel that is Gaussian in space and has a 24 hour period in time is adopted spatial-diurnal surface of the Cox process can be computed and visualized using techniques such as the 3D isosurface [3]. There are several shortcomings in adopting such a simple model. As predictions are obtained using a plug-in point estimate of the underlying intensity of the Cox process model uncertainty is ignored. The non-Bayesian method also does not provide principled means of bandwidth selection or for avoiding edge effects. Finally and perhaps most compelling kernel smoothing is just a simple function of the data which means that the computed surface is most reasonably seen as a summary of history rather than a forecast. A more sophisticated modelling approach may intelligently identified patterns in the data that are likely to continue into the future. Using a smoothed summary of history for forecasting can be particularly problematic in temporal models and simplifications such as only using the hour of day (i.e. diurnal time) component must be employed. However despite the statistical shortcomings of such an approach it is delivering something useful to decision makers in the form of comprehensible outputs related to operational concerns.

A second path uses sophisticated Bayesian models to obtain predictions, but usually on relatively simple problems and without sophisticated 3D visualization in order to make the implications of these complex mathematical objects apparent to the decision maker.

An important class of Bayesian non-parametric models of Cox processes is the Gaussian Cox Processes which models the intensity function as a non-negative function of a Gaussian process, usually an exponential function resulting in a log Gaussian Cox Process.

A general expression for the count C of incidents in a region of space and time R under a log Gaussian process is given by

$$C \sim \text{Poisson}\left(\iiint_R e^{\phi(x,y,t)} \mathrm{d}x\mathrm{d}y\mathrm{d}t\right)$$

where $\phi(x, y, t)$ is a Gaussian Process. A further restriction is that the distribution of counts for non-overlapping regions R and R' should be independent.

Fully Bayesian approaches to this problem are hampered by the presence of doubly intractable integrals [12]. That is the evaluation of the likelihood and as such the posterior itself contains an intractable integral, this makes applying Markov chain Monte Calro (MCMC) algorithms difficult. Two possible solutions have been proposed, the first makes use of the fact that MCMC algorithms have been found to handle doubly intractable integrals for the special cases where the model can be sampled from [11] [13]. While sampling from a log-Gaussian Cox process is not possible sampling from a modified model the sigmoidal Gaussian Cox Process is possible resulting in the first MCMC algorithm to handle Gaussian Cox Processes [1], i.e. the first MCMC algorithm with the stationary distribution of the posterior of a Gaussian Cox process.

Although this approach appears to offer many advantages and allows MCMC inference to be performed on a very appealing model there are some drawbacks. The first is computational cost, this algorithm performs a matrix inversion on a matrix that includes all observed data and additionally latent data incorporated into the model, this is likely to be a considerable burden. Secondly the main advantage of this approach is to avoid imposing the model on a discrete grid as in [9] however one of our goals is to visualize the model using 3D graphical tools such as Mayavi [14] which itself uses a discrete scalar field. Thirdly there are comforting results that given a sufficiently fine grid the discrete model will converge to a continuous model [16]. Finally the discrete model has a much longer history being applied to real problems including to problems of fire incidence [10].

There exist Bayesian non-parametric models of Cox processes which do not rely on the Gaussian process for the intensity function, but rather use a mixture model for the intensity function such as [8] which uses a Dirichlet process mixture of Beta distributions. This has the advantage of allowing a relatively standard MCMC sampler, but the disadvantage of losing the intepretability of the Gaussian process which has been extensively used in both spatial statistics under the name of Krigging [4] and in machine learning [15]. Indeed one of our goals here is to incorporate the spatial diurnal kernel developed in [3] to the log Gaussian Cox process context, it is very unclear if a Cox process with periodicity could be formulated in a mixture model framework.

2 Discretised Log Gaussian Cox Process Model

An outline of the model is developed first in space only and then in space and diurnal time.

2.1 Spatial Model

The discretized spatial only model takes the following form

$$C_{i,j} \sim \text{Poisson}(e^{\phi_{i,j}}) \tag{1}$$

where ϕ is given a matrix variate normal distribution, or alternatively $\gamma = e^{\phi}$ is given a matrix variate log normal distribution. The purpose of the prior is to impose a correlation between elements of C that are close space. For our application the desirable dimension of C and consequently ϕ is $I \times J$ which in the current context is 240×240 or 57600 parameters for ϕ. While this is large it remains manageable, on the other hand the covariance matrix for the normal distribution over ϕ is this size squared which is not easily manageable. Fortunately both for conceptual and computational reasons a matrix variate normal distribution can be employed instead. While the matrix variate distribution can be interpreted as a special case of the multivariate normal distribution in that

$$\mathrm{vec}(\phi) \sim \mathcal{N}_{I\times J}(\mathrm{vec}(\mu), \Omega \otimes \Gamma). \tag{2}$$

here $\mathrm{vec}(\phi)$ denotes that the $I \times J$ array or matrix of ϕ is converted to a single vector $\mathrm{vec}(\phi) = [\phi_{1,1}, ..., \phi_{1,J}, ..., \phi_{I,1}, ..., \phi_{I,J}]^T$, Ω is an $I \times I$ matrix and Γ is a $J \times J$ matrix and $\otimes$ is the Kronecker tensor product, note that for many applications computing $\Omega \otimes \Gamma$ would require allocating very large amounts of memory for a highly redundant array.

It is instructive to consider the covariance between $\phi_{i,j}$ and $\phi_{i',j'}$ which is $\Omega_{i,i'}\Gamma_{j,j'}$, in the context of a Gaussian process a covariance function would specify a covariance between two points (i, j) and (i', j'), in practice there are a number of popular forms that result in close points having high covariance such as exponential and the more general Matern family [4] [6] . If an exponential covariance function is adopted then $\mathrm{Cov}(\phi_{i,j}, \phi_{i',j'}) = \exp(-(\frac{i-i'}{I\sigma})^2 - \frac{j-j'}{J\sigma})^2)$, it is easily seen that this can be obtained by letting $\Omega_{i,i'} = \exp(-(\frac{i-i'}{I\sigma})^2)$ and $\Gamma_{j,j'} = \exp(-(\frac{j-j'}{J\sigma})^2)$.

The main advantage of employing a matrix normal rather than multivariate normal prior is that by factorizing the covariance matrix $\Sigma = \Omega \otimes \Gamma$ it is possible to evaluate the prior and therefore the posterior using reasonable amounts of memory, by using the following expression for the matrix variate normal distribution.

$$P(\phi|\mu, \Omega, \Sigma) = \frac{\exp\left(-\frac{1}{2}\mathrm{tr}\left[\Omega^{-1}(\phi-\mu)^T\Sigma^{-1}(\phi-\mu)\right]\right)}{(2\pi)^{\frac{1}{2}IJ}|\Omega|^{I/2}|\Sigma|^{J/2}} \tag{3}$$

Alternatively $\lambda = e^{\phi}$ can be given a matrix variate log normal distribution

$$P(\lambda|\mu, \Omega, \Sigma) = f_\lambda(\lambda) = \frac{\exp\left(-\frac{1}{2}\mathrm{tr}\left[\Omega^{-1}(\log(\lambda)-\mu)^T\Sigma^{-1}(\log(\lambda)-\mu)\right]\right)}{(2\pi)^{\frac{1}{2}IJ}|\Omega|^{I/2}|\Sigma|^{J/2}\prod_{i=1..I}\prod_{j=1..J}\lambda_{i,j}} \tag{4}$$

By using this expression we are able to formulate a complete expression for the posterior

$$P(\lambda|C) \propto f_\lambda(\lambda)\prod_{i=1..I}\prod_{j=1..J}\mathrm{Poiss}(C_{i,j}|\lambda_{i,j}) \tag{5}$$

Using this expression it is possible to use MCMC algorithms to generate samples from $P(e^{\phi}|C)$. A simple random walk Metropolis algorithm can be used, or because e^{ϕ} is non-negative a multiplicative Metropolis hastings algorithm can also be used with multivariate log normal proposals with low variance and expectation of 1. The advantage of this procedure is that the proposals like the target have non-negative support. This method was applied to the model where $I = J = 10$ and where C is equal to the identity matrix and $\mathrm{Cov}(\phi_{i,j}, \phi_{i',j'}) = \exp(-(i-i')^2 - (j-j')^2)$. A multiplicative Metropolis proposal was used with an expectation of 1 and a variance of 0.25 the chain was run for 2 million iterations, with thinning so that only 1 in 100 samples were retained this resulted in a fraction of 0.08 samples being accepted (rather than the optimal 0.25). A visual display of $E[e^{\phi}|C]$ is shown in Figure 1(a). This output shows some pleasing features in that the highest values are where counts have been observed on the diagonal, and it appears that the covariance is operating correctly as cells adjacent to the diagonal also have an increased value. Very similar results are obtained by applying a standard random walk Metropolis algorithm to ϕ. This distribution can be obtained by multiplying the posterior by the Jacobian, and in our experience resulted in a slightly more efficient algorithm.

$$P(\phi|C) = f_{\phi}(\phi) = f_{\lambda}(e^{\phi}) \left| J(e^{\phi}, \phi) \right| = f_{\lambda}(e^{\phi}) \prod_{i=1..I} \prod_{j=1..J} e^{\phi_{i,j}} \tag{6}$$

where $J(\cdot)$ is the Jacobian.

2.2 Spatial-Diurnal Model

The discretized spatial-diurnal model has the following form

$$C_{i,j,k} \sim \mathrm{Poisson}(e^{\phi_{i,j,k}}) \tag{7}$$

where ϕ is given a 'tensor' variate normal distribution, or alternatively $\gamma = e^{\phi}$ is given a 'tensor' variate log normal distribution. The purpose of the prior is to impose a correlation between elements of C that are close space. For our application the desirable dimension of C and consequently ϕ is $I \times J \times K$ which in the current context is $240 \times 240 \times 24$ or approximately 1.4 million parameters for ϕ. In this model like the previous model, i, j and k index cells with different spatial co-ordinates and j index the diurnal or hourly of day component of time. The covariance between two points is given by $\mathrm{Cov}(\phi_{i,j,k}, \phi_{i',j',k'}) = \exp(-(\frac{i-i'}{I\sigma})^2 - (\frac{j-j'}{J\sigma})^2 - \frac{c^2}{\sigma^2}\Delta(k,k'))$, where $\Delta(k,k') = -\frac{1}{2}\cos(\frac{2\pi(k-k')}{24K}) + \frac{1}{2}$, and the parameter c is a constant which converts a distance in time of 12 hours to a distance in meters, note that $\Delta(k, k'+12K) = 1$, so that $\mathrm{Cov}(\phi_{i,j,k}, \phi_{i,j,k+12K}) = \exp(-\frac{c^2}{\sigma^2})$, an equivalent covariance can be achieved purely at a distance in this case in the i direction as $\mathrm{Cov}(\phi_{i,j,k}, \phi_{i+cI,j,k}) = \exp(-\frac{c^2}{\sigma^2})$, so that

$$\mathrm{vec}(\phi) \sim \mathcal{N}_{I\times J\times K}(\mathrm{vec}(\mu), \Omega \otimes \Gamma \otimes \Psi). \tag{8}$$

where, $\Omega_{i,i'} = \exp(-(\frac{i-i'}{I\sigma})^2)$ and $\Gamma_{j,j'} = \exp(-(\frac{j-j'}{J\sigma})^2)$, and $\Psi_{k,k'} = \exp(-\frac{1}{2}\cos(\frac{2\pi(k-k')}{24K}) + \frac{1}{2})$.

While a tensor variate normal distribution can be expressed using tensor products, it can also be expressed using linear algebra and the Kronecker tensor product available in most scientific programming languages such as GNU Octave [2] as used here. This involves reshaping ϕ and μ which have $I \times J \times K$ to ϕ' and μ' to dimensions $I \times JK$. The distribution over ϕ' can then be expressed as

$$P(\phi'|\mu', \Omega, \Sigma, \Psi) \propto \frac{\exp\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\log \mathrm{Pois}(C'_{i,jK+k}) - \frac{1}{2}\mathrm{tr}\left[\Omega^{-1}(\phi'-\mu')^T(\Sigma^{-1}\otimes\Psi^{-1})(\phi'-\mu')\right]\right)}{(2\pi)^{\frac{1}{2}IJK}|\Omega|^{I/2}|\Sigma\otimes\Psi|^{JK/2}}. \tag{9}$$

The full posterior is then $P(C'|\phi')P(\phi'|\mu', \Omega, \Sigma, \Psi)$, where C' is also reshaped from a tensor to a matrix with dimensions $I \times JK$. The advantage of this solution is that a compact representation of the posterior is available which can be written in standard scientific programming languages (in this case GNU Octave). The disadvantage is that $\Sigma \otimes \Psi$ will require a large amount of memory to be allocated and this is in principle avoidable.

3 Results and Discussion

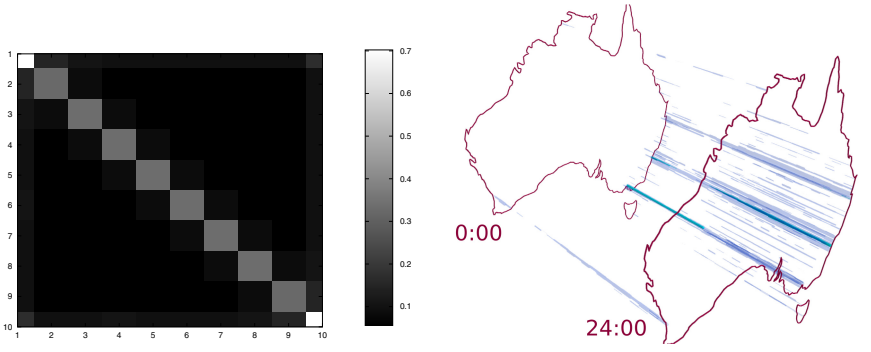
A log Gaussian Cox Process was run on a discrete $I \times J \times K$ grid where $I = J = 240$ and $K = 24$ with the following covariance function. The dataset this was applied to was the occurrence in time and hour of day of malicious hoax calls within metropolitan Australia.

$$\begin{aligned}\mathrm{Cov}(\phi_{i,j,k}, \phi_{i',j',k'}) = \\ \exp\{&-(x_{\max}-x_{\min})((i-i')/(I\sigma))^2 - (y_{\max}-y_{\min})((j-j')/(J\sigma))^2 \\ &+ \frac{c}{\sigma^2}(\frac{1}{2}\cos(2\pi(k-k')/(24K)) - \frac{1}{2})\}\end{aligned} \tag{10}$$

where $x_{\max} - x_{\min} = 4000$ km and $y_{\max} - y_{\min} = 3600$ km and $\sigma = 5$.

The initial value of the Markov chain was obtained by setting $\phi^{(0)}_{i,j,k} = \log(C_{i,j,k} + 1)$, the MCMC algorithm then proceeds using the random walk Metropolis algorithm for 5000 iterations with a burn in of 1000, and with thinning such that only every tenth sample is used.

A 3D iso surface rendered with the Mayavi library is shown in Fig 1(b). Incidence of malicious hoax calls are visible in time and hour of day, with large numbers of incidents in metropolitan regions such as Sydney and Melbourne and with evident trends of higher and lower incidence through different parts of the day. This surface is qualitatively similar to the spatial diurnal iso surfaces



(a) (b)

Fig. 1. (a) Approximation of $E[\lambda|C] = E[e^{\phi}|C]$ depicted graphically, the approximation is computed using multiplicative random walk Metropolis algorithm. (b) Iso surfaces of $E[\lambda|C] = E[e^{\phi}|C]$ for a spatial-diurnal Cox process on a dicretized $240 \times 240 \times 24$ grid, with the expectation computed with MCMC methods.

of Cox processes used in previous studies such as [3]. As such the visual output of this model qualitatively has the same appealing features that are useful in operational context. The work here improves the state of the art by applying the spatial diurnal Cox process a more complex model that previously considered and demonstrating 3D visualization techniques on this.

Several aspects of this problem deserve deeper consideration including statistical modelling issues including inference for the bandwidth and more complex models for time i.e. that incorporate more than the diurnal component. Computational issues are also of interest, it seems likely that more powerful MCMC algorithms may be required if inference is applied to a sufficiently fine grid as the approach described for setting the initial value of the chain will become less effective on a smaller grid. Although there were no problems were observed in the experiments demonstrated if larger grids are considered numerical problems are possible due to the very large matrix representations. Finally the fact that a full posterior distribution is available should enrich visualization possibilities. We are considering these issues in future work.

Acknowledgements. David Rohde acknowledges partial support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

References

1. Adams, R.P., Murray, I., MacKay, D.J.C.: Tractable nonparametric Bayesian inference in poisson processes with gaussian process intensities. In: Proceedings of the 26th International Conference on Machine Learning (ICML 2009), Montreal, Quebec (2009)

2. Eatonand, J.W., Bateman, D., Hauberg, S.: GNU Octave Manual Version 3. Network Theory Limited (2008)
3. Brunsdon, C., Corcoran, J., Higgs, G.: Visualising space and time in crime patterns: A comparison of methods. Computers, Environment and Urban Systems 31(1), 52–75 (2007)
4. Chiles, J., Delfiner, P.: Geostatistics. Wiley, New York (1999)
5. Cox, D.R.: Some Statistical Methods Connected with Series of Events. Journal of the Royal Statistical Society. Series B 17(2), 129–164 (1955)
6. Cressie, N.A.C.: Statistics for spatial data. Wiley series in probability and mathematical statistics: Applied probability and statistics. J. Wiley (1991)
7. Diggle, P.J.: A kernel method for smoothing point process data. Applied Statistics 34, 138–147 (1985)
8. Kottas, A., Sansó, B.: Bayesian mixture modeling for spatial poisson process intensities, with applications to extreme value analysis. Journal of Statistical Planning and Inference 137, 3151–3163 (2009)
9. Møller, J., Syversveen, A.R., Waagepetersen, R.P.: Log Gaussian Cox processes. Scandinavian Journal of Statistics 25, 451–482 (1998)
10. Moller, J., Vej, F.B.: Structured spatio-temporal shot-noise cox point process models, with a view to modelling forest fires. Scandinavian Journal of Statistics 37(1) (2010)
11. Moller, J., Pettitt, A.N., Reeves, R.W., Berthelsen, K.K.: An efficient markov chain monte carlo method for distributions with intractable normalising constants. Biometrika 93(2), 451–458 (2006)
12. Murray, I., Ghahramani, Z.: Bayesian learning in undirected graphical models: Approximate MCMC algorithms. In: Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI 2004), pp. 392–399. AUAI Press, Arlington (2004)
13. Murray, I., Ghahramani, Z., MacKay, D.J.C.: MCMC for doubly-intractable distributions. In: Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI 2006), pp. 359–366. AUAI Press (2006)
14. Ramachandran, P., Varoquaux, G.: Mayavi: 3D Visualization of Scientific Data. Computing in Science & Engineering 13(2), 40–51 (2011)
15. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. Adaptive Computation and Machine Learning. MIT Press (2006)
16. Waagepetersen, R.: Convergence of posteriors for discretized log gaussian cox processes. Statistics & Probability Letters 66, 229–235 (2003)

Soft Sensor for Fluoridated Alumina Inference in Gas Treatment Centers

Alan M.F. de Souza[1], Carolina de M. Affonso[2], Fábio M. Soares[3], and Roberto C.L. de Oliveira[3]

[1] Professional Master Degree in Industrial Processes, Federal University of Pará
[2] Electrical Engineering Faculty, Federal University of Pará
[3] Computing Engineering Faculty, Federal University of Pará
mrcl.fs@gmail.com, {carolina,fms,limao}@ufpa.br

Abstract. The Gas Treatment Center performs a key role in the aluminum smelting process, since it strongly influences the chemical and thermal stability of the electrolytic bath through fluoridated alumina. Therefore this variable should be considered to keep the bath chemistry under control. However, the fluorine concentration measurement in fluoridated alumina is very time-consuming and that information becomes available only after a while. By using Artificial Neural Network we developed a Soft Sensor capable to estimate the fluorine concentration in fluoridated alumina, and to provide that information to plant engineers in a timely manner. This paper discusses the methodology used and the results of an implemented Soft Sensor using Neural Networks on fluorine estimation in fluoridated alumina from a Gas Treatment Center in an important Brazilian Aluminum Smelter.

Keywords: Gas Treatment Center, Aluminum Industry, Fluoridated Alumina, Artificial Neural Network, Soft Sensor.

1 Introduction

Great changes happened in the world since the industrial revolution. Production processes have expanded to large-scale, yielding good quality products in large quantities. The productive control had become more challenging [1]. On the early period of modern control theory, [2] points out that the control strategy was heavily based on instrumentation whose main job was to collect real time data from the plant. However, these devices usually were expensive; their operation normally had to be performed under harsh environments and needed to provide real data with the least possible noise. So, the operating process cost scaled as the production increased.

This problem could be partly solved by using intelligent computational techniques and data mining. On the second half of the twentieth century, major advances in this field, whose results complied with actual data, enabled process experts to identify hidden patterns in a simplistic analysis [3].

These techniques later evolved into the construction of virtual instruments or soft sensors [4]. Through the input variables in real time, we can estimate the value of the

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 294–302, 2012.

output variable without instrumentation or laboratory analysis. This leads to a drastic reduction in the company's operating costs and furthermore enables plant simulation without harming the production process.

Soft sensors are computer applications (software) developed to estimate the behavior of variables whose measure is effort-consuming or requires laboratory analysis. Soft sensors typically have low response time and good accuracy since they are software-based, whose accuracy is based on the process database [4]. Some works using soft sensors show satisfactory results for chemical processes. Damour et al developed a soft sensor to infer sugar crystallization in an on-line manner using a model approach with equations on the process parameters [5]. Assis and M. developed an algorithm for estimation of bioreactor state, using extended Kalman Filters, Artificial Neural Networks [6]. Soares created a soft sensor to estimate bath temperature in Aluminium Smelting Cells [7]. Nelles gives an insight on nonlinear system identification, which encompasses the soft sensor theory, worked both with Neural and Fuzzy Models [8].

Among the many advantages of soft sensor for industry, we highlight:

- Replacement of expensive hardware: in most cases, soft sensors require a computer or a single processor machine to run, while real measures should be performed by high cost hardware both for acquisition and maintenance;
- Flexible deployment: soft sensors can be deployed in devices provided with a processor unit, such as computers, controllers, because they are mathematical algorithms.
- Real time estimation: each sample of target variable is collected daily, but the results are usually obtained one day after. With soft sensors, the analysis is done algorithmically, taking a few seconds to process input variables and provide an estimated value;
- Fault tolerance: soft sensors are not subject to the same problems of real sensors (which may cause noise or break), because they are intangible.

The novel of this work is given in the use an artificial neural network (ANN) based soft sensor to estimate the percentage of fluoride in fluoridated alumina that comes out from a GTC. We explain a bit of the aluminum smelting process and the fluoride alumina production. Then, we specify the soft sensor design. In the next section, we publish the achieved results and the conclusion. The article is finished with the acknowledgements and the references used in this work.

2 Aluminum Smelting Process and Fluoridated Alumina Production

We find aluminum in natural form but it is always linked to some other chemical element as salts or oxides. In order to have pure aluminum, there should be a separation process. The known and worldwide used Hall-Héroult process breaks the alumina (Al_2O_3) molecules through electrolysis achieving over 99% aluminum purity [9].

However, this process requires an exorbitant expenditure of energy since it occurs in a cryolite (Na_3AlF_6) bath whose melting point is higher than 1000ºC and the

reaction must be done under such high temperatures. Some chemical elements should be added in the aluminum smelting pot to help reduce the temperature and also preserve bath thermal and chemical stability. Fluoridated (or secondary) alumina is produced through the Gas Treatment Center (GTC), where the dirty fluoridated gas is collected from the reduction pots and gets in contact with virgin (or primary) alumina. Since the primary alumina is a porous material, it adsorbs fluoride from the gaseous mass, producing clean gas and secondary alumina [10]. The alumina, in a fluoride state, is then deposited into filters, which are polyester structures that separate alumina and clean gas. The clean gas is then expelled through the chimney and the secondary (fluoridated) alumina is deposited in the silo to return to reduction pots as raw material. Figure 1 shows the process schema.

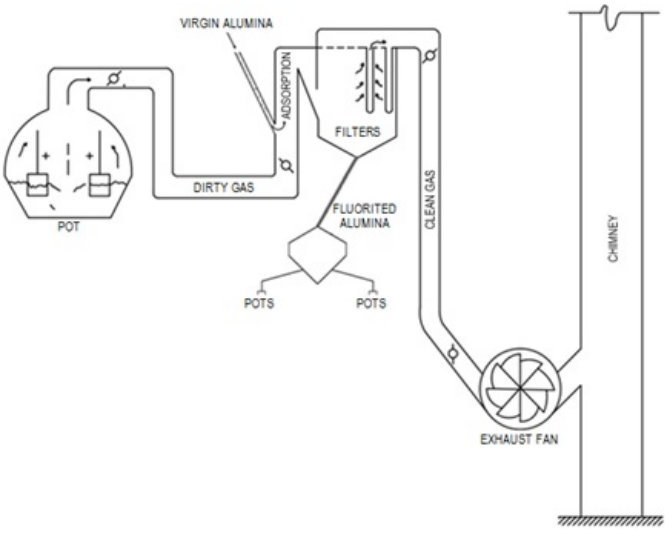


Fig. 1. Process Schema

The chemical and thermal balance of these reduction pots are directly affected by fluoridated alumina. Moreover, strong variations in the fluoridated alumina can bring about strong variations in bath temperature, given that the control system considers only the fluoride existing in the bath.

3 Alumina Fluoride Soft Sensor Design

The strategy to build the soft sensor used in this study follows the methodology described in [4]:

- Collect, sort and filter historical data;
- Define the structure of the model;
- Estimate the model;
- Validate the model.

3.1 Plant Historical Data Collection

In this phase the soft sensor designer and the smelter's process engineers must cooperate to decide which data should be considered. According to the available literature [9][10][11], there are about 20 input variables which influence the GTC behavior; from these, process experts indicated seven variables which should strongly have effect on fluoridated alumina:

- Sodium Concentration in Primary Alumina;
- Particle Size (physical variable): 100 Mesh, 200 Mesh, -325 Mesh;
- Gas Pickups Amperage (GTC variable): Exhaust Fan 1, Exhaust Fan 2, Exhaust Fan 3.

3.2 Data Sorting and Filtering

Data collection period comprises 20 months. Since most records are daily recorded, there are about 630 samples collected. The GTC exhausters' electric current records present spurious values that compromise the neural model giving inaccurate results.

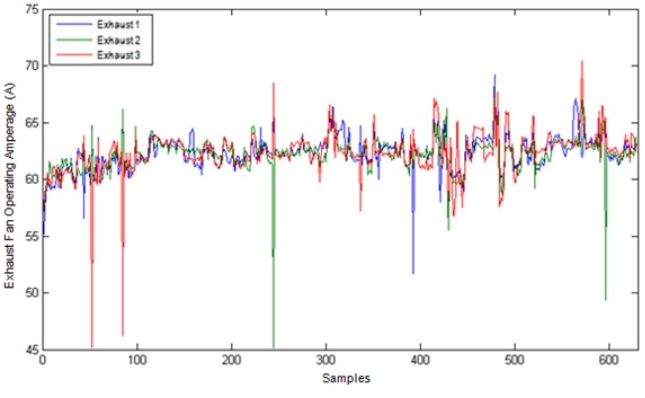


Fig. 2. Exausts Fan Samples

In order to eliminate this kind of spurious data, we used the 3Sigma Rule defined by Equation 1 below.

$$d_i = \frac{x_i - \bar{x}}{\sigma_i} \quad (1)$$

where x_i is variable x at instant i; $\bar{x}$ is variable mean; σ_i is variable standard deviation; d_i is weighted distance from x_i to variable mean according standard deviation. Data whose weighted distance 3 times the standard deviation should be excluded.

The data have been split into subsets for model training (70%), validation (15%) and testing (15%).

3.3 Neural Model Structure

The modeling architecture chosen to represent the ANN model is NARX (*Nonlinear Auto-Regressive with eXogenous Inputs*) represented by a Multilayer Perceptron (MLP), because the applicability of this technique is aimed at mapping dynamic typically non-linear systems [8].

NARX models can be represented in parallel or in series. In the in parallel type, the inputs x(t) and the estimated output $\hat{y}(t)$ are used as network input to calculate the estimated values. On the other hand, the in series type, only the network input x(t) are used and, unlike the parallel type, the real output y(t) is used to calculate the estimated output $\hat{y}(t)$ [12]. According to [13], there are two advantages in using the in series type. The first is that the network input (real output) is more accurate than the estimated output. The second is that the resulting network has a purely feed-forward architecture, allowing to be trained using traditional algorithms. For these reasons, the in series type was chosen in this work.

3.4 Training Algorithm

We chose the Levenberg-Marquardt [12] as training algorithm. This algorithm is able to drive the training of MLP networks faster than traditional algorithms. It is a second order gradient method based on the least squares method for nonlinear models, which uses Newton's method approach in order to enhance the training process efficiency [14]. Normalization also has been applied in order to prevent saturation.

4 Results

In order to build the NARX neural model capable to estimate the real samples the most accurate possible, different network configurations were tested through a computer program. For training, we ranged the number of neurons in the hidden layer and the network delay. We trained each topology 60 times, since the weights initialization and subsets division for training, validation and testing were done randomly.

The quality checker parameter of the model error is given by MSE (Mean Square Error). The following chart (Figure 3) shows the behavior of the MSE error of each candidate for the definitive model ANN. Note that four topologies were pre-selected: ANN Candidates 9, 10, 11 and 12. The criterion for this choice was due to their MSE, which stood within the range [0.01, 0.02].

To determine the Final ANN, we used Absolute Percentage Error (APE) values to check which pre-selected ANN candidates were more concentrated in the range of values 0 to 10%. Figures 4 (a) thru Figure 4 (d) show the respective histograms.

It seems that the APE histogram of ANN 11, represented in Figure 4 (c), is more concentrated near zero than the other histograms and has 458 records in the range of 0 to 10%, compared at 432 of ANN 9, 441 of ANN 10 and 445 of ANN 12.

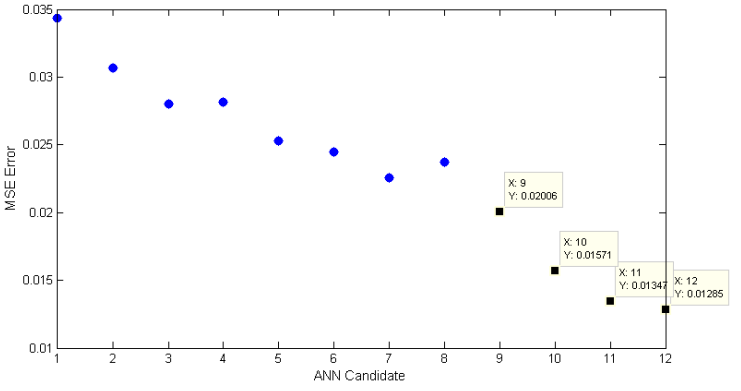


Fig. 3. ANN Candidates' MSE Error. The topologies with error less than 0.02 are highlighted.

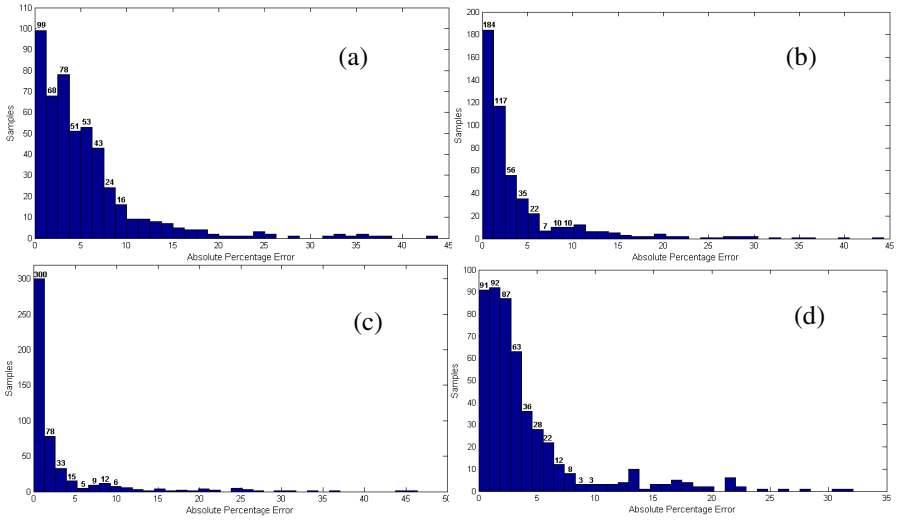


Fig. 4. (a) ANN 9 APE Histogram (b) ANN 10 APE Histogram (c) ANN 11 APE Histogram (d) ANN 12 APE Histogram

In order to assure the ANN 11 quality, the scatter plots of four pre-selected ANN candidates are shown in Figure 5, where the x-axis represents real output while the y-axis represents estimated output. We also attested by this analysis that the ANN 11 best reflects the behavior of the Gas Treatment Center. It generates accurately estimates of fluoridated alumina from the variables mentioned in the previous section, taking into account that the system is a very complex and non-linear.

After reviewing the APE histograms and Scatter Plots, we ratify that ANN 11 is the best model to estimate the percentage of fluoride in fluoridated alumina from the GTC. Table 1 shows the final structure chosen for the model.

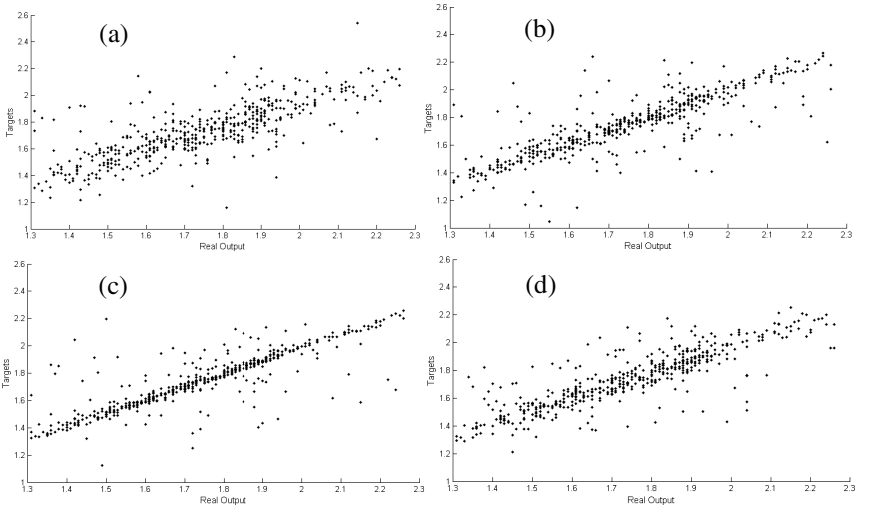


Fig. 5. (a) ANN 9 Scatter plot (b) ANN 10 Scatter plot (c) ANN 11 Scatter Plot (d) ANN 12 Scatter plot

Table 1. Final ANN Structure

Number of Layers	3
Neurons on 1st Layer	35 (7 input neurons times 5 delays for each)
Activation Function	Hyperbolic Tangent
Neurons on 2nd Layer	25
Activation Function	Hyperbolic Tangent
Neurons on 3rd Layer	1
Activation Function	Pure Linear
Number of Delays on Input Variables	5
Number of Delays on Output Variables	1

In this work, each aluminum potline has two GTCs and that the data used for training the ANN are definitely coming from the GTC 1. To demonstrate the generalization of the model, the GTC 2 records were used to simulate the Final ANN. The result estimated by the soft sensor can be seen in Figure 6. Note in Figure 6 (a) that for these almost 220 records, the network could to respond satisfactorily, confirming the generalization ability. Through Figure 6(b), it seems that the absolute error between real output and estimated output has a maximum value of about 0.22. But the vast majority of the values are between 0.01 and 0.15, which is an acceptable error range by the GTC process engineers.

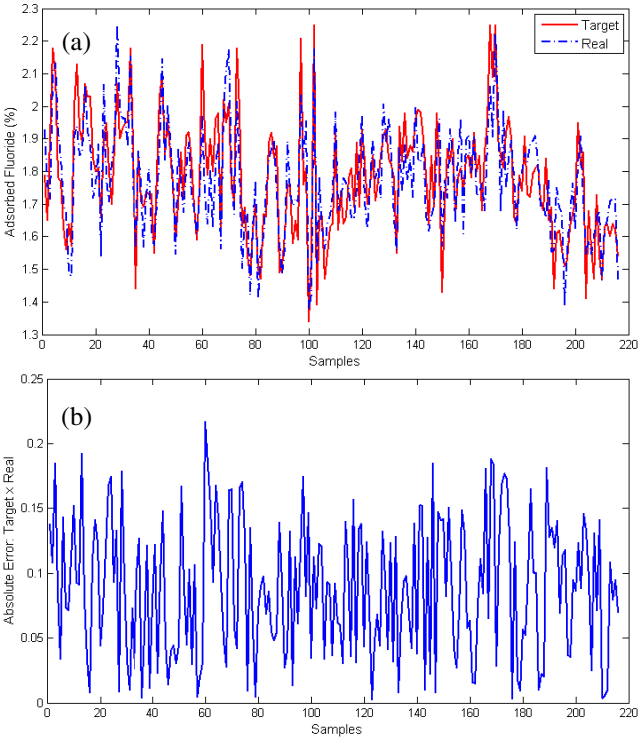


Fig. 6. (a) Comparison Chart between Target and Real Output by considering data from GTC 2. (b) Absolute Error Chart between Target and Real Output.

5 Conclusions

From the identified neural model using the Computational Intelligence technique known as Artificial Neural Network, shown in this work, we conclude that it is possible to make indirect measurements, without having to analyze daily samples in the laboratory, which saves time and reduces operating costs for the company. In addition, the team of engineers from aluminum center will have an option to accurately and quickly simulate the behavior of the Gas Treatment Plant, which contributes to the production of alumina fluoridation of better quality. The results will also serve in the future to build a plant control system, thus increasing the competitiveness of Brazilian industry in the international scenario.

References

1. Stearns, P.N.: The Industrial Revolution in world history, 3rd edn. Westview Press (2007)
2. Ogata, K.: Modern Control Engineering, 4th edn. Prentice-Hall (2007)
3. Haykin, S.: Neural Networks, A Comprehensive Foundation, 2nd edn. Prentice-Hall (1999)

4. Fortuna, L., Graziani, S., Xibilia, M.G.: Soft Sensors For Monitoring And Control Of Industrial Processes, 1st edn. Springer (2007)
5. Damour, C., et al.: Soft-Sensor for Industrial Sugar Crystallization: On-Line Mass of Crystals, Concentration and Purity Measurement. Control Engineering Practice 18(9), 839–844 (2010)
6. Assis, A.J., Filho, R.M.: Soft Sensors Development For On-Line Bioreactor State Estimation. Comput. Chemistry Engineering 24, 1099–1103 (2000)
7. Soares, F.M., Oliveira, R.C.L., Castro, M.A.G.: Bath Temperature Inference through Soft Sensors using Neural Networks. In: TMS Light Metals, Proceedings of the Technical Sessions, 140th Technical TMS Annual Meeting, pp. 467–472 (2010)
8. Nelles, O.: Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models. Springer, Germany (2005)
9. Grjotheim, K., Kvande, H.: Introduction to Aluminium Electrolysis Understanding the Hall-Héroult Process, 2nd edn. Aluminium-Verlag (1993)
10. Shinzato, M.C.: Removal of Lead and Cromium from aqueous solution by natural zeolites associated with eruptive rocks from Serra Geral formation, Paraná sedimentary basin. RevisãoQuímica Nova 32(8) (2009) ISSN 0100-4042
11. Malaviya, A.V., Bundell, G.A.: An Intelligent Controller for Aluminum SmeltersPotlines. IEEE Transactions Industry App. 37(3), 792–805 (2001)
12. Hagan, M., Demuth, H., Beale, M.: Neural Network Toolbox Design Book User's Guide Version 7, 11th edn. The MathWorks Inc. (2010)
13. Hagan, M., Menhaj, M.: Training Feedforward Networks With the Marquardt Algorithm. IEEE Transactions on Neural Networks 5(6), 989–993 (1994)
14. Silva, I.N., Spatti, D.H., Flauzino, R.A.: Rede Neurais Artificiais: para Engenharia e Ciências Aplicadas, 1st edn., São Paulo, Artliber (2010)

Estimation of Aluminium Fluoride Concentration in Aluminium Reduction Cells through a Soft Sensors Approach

Otacilio Fontes, Fábio M. Soares, and Roberto Limão

Federal University of Para
omsfontes@yahoo.com.br, {fms,limao}@ufpa.br

Abstract. This work exploits a model for Aluminium Fluoride Concentration Measurement in the Aluminium Smelting process. This process variable is usually measured every 50-100 hours since it requires long laboratory analysis. This variable has a strong influence on the whole process, thus it should be controlled in a shorter basis. In order to prevent the long time between measurements, we developed a soft sensor based on neural networks which allows estimating the fluoride concentration at any moment by querying against the available process database. This database encompasses the consolidated knowledge on this chemical process and thus can be used both to build and validate the model.

Keywords: Soft Sensors, Aluminium Smelting Process, Bath Chemistry, Neural Networks, Fluoride Concentration.

1 Introduction

The world became dependent of Aluminium. This light metal is present in almost everything in modern life. Cars, planes, computers, cans are just a few examples of what can be made out of this metal. These productions became economically viable after Hall and Heroult found out simultaneously a process to produce pure Aluminium in industrial scale [1]. Since then, this process has improved to allow a larger production year after year. Many technologies have emerged allowing better process control on the plant. One key point on these improvements is regarding the soft sensors, whose aim is to replace instrumentation devices which are subject to failures, noise, and require long term analysis [2]. With the soft sensors approach, expensive sensors and manual jobs can be replaced, and besides, it offers the possibility to simulate the process.

Recent works have shown that these sensors are really useful in these complex and non-linear process [3]. In the Aluminium industry, similar works have shown the use of these devices. Frost and Karri [4] built two neural network based models to model the fluoride concentration in electrolytic cells; Malaviya and Bundell [5] have published an intelligent cell control strategy based on artificial intelligence techniques, in which data are collected to build the model, then the model is simulated to provide intelligent control; Branco [6] developed a neural network based model to emulate

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 303–311, 2012.

the cell voltage resistance; Pereira [7] built a fuzzy based strategy to control fluoride addition on the cell, since operators usually do this job manually by analyzing the variables' values; Soares [8] built a neural based model for bath temperature inference, whose logic was embedded in a soft sensor architecture.

We have analyzed carefully the aforesaid jobs under a methodology available in the literature [3] to build a fluoride concentration estimator. Although this variable is very important in the aluminium smelting process, it is usually measured only within a period of every 2-4 days in laboratory. By the fact that this variable also appears to be dependent on others such as bath temperature [9], one can build an estimator capable of giving the value of fluoride concentration at any time since the input values are available. This work is divided into 4 sections:

- Introduction: In this section we provide general information on the context of the work, its objectives and motivations;
- Bath Chemistry: In this section we detail the bath chemistry process, explaining the impacts this work will cause;
- Model building and results: In this section we show step by step how the model was built and discuss some results;
- Conclusions: Finally we conclude this paper's results and suggest some future works.

2 Bath Chemistry in the Aluminum Smelting Process

Aluminum can be found in nature in salts or oxides. In order to obtain pure aluminum, there should be a separation process. The known and worldwide used Hall-Héroult process breaks the alumina (Al_2O_3) molecules through electrolysis achieving over 99% of aluminum purity [2].

However this process requires a great expenditure of energy since it occurs in a cryolite (Na_3AlF_6) bath whose melting point is higher than 1000ºC and the reaction must be done under such high temperatures. Some chemical elements should be added in the aluminum smelting pot to help reduce the temperature to facilitate the electrolysis and to save energy. Additionally, these elements are essential for maintaining the pot chemical stability.

The bath chemistry process takes place to stabilize cell's bath composition and heat balance. It is a very non-linear and complex process, whose control is very difficult and challenging [9]. This process becomes even more complex if measurements are considered. Most of their variables are measured offline or in laboratory. Usually bath temperature is measured once a day or two, and fluoride concentration is analyzed every 2 or 4 days. This subject has also been well studied, and some partial analytical models on the process have been developed [8, 10].

Although there are some mathematical equations to help process control, in the most of times the control is done manually by an expert. This fact has motivated many modeling works to predict or control bath chemistry variables such as bath temperature and aluminium fluoride, some of them based on artificial intelligence techniques [4, 7, 8], previously detailed in section 1, and also purely mathematical models such as [10]. The multivariable model of McFadden et al [10] provides a basis for which variables are related to each other, which is very important in the model building phase.

We can see through these works that bath chemistry variables can be estimated both through black-box, gray-box or even white-box models. These models have been successfully used since the process is non-linear so it does not make difference whether there are equations dictating the process or not. But the process anyway should be modeled by data from the available process database, because the data represent what really happened in the cell, therefore they can drive model development.

3 Model Building and Results

The strategy to build the model used in this study follows the methodology described in [4]:

- Collect, sort and filter historical data;
- Define the structure of the model;
- Estimate the model;
- Validate the model.

3.1 Data Collection

In this phase we performed a research in the process database in order to find relevant data. We performed interviews with the experts and data statistical analysis by using correlation.

Over 200 variables are kept in the process database, from which 40 are used specifically in the bath chemistry or have effect on it, according to the process team. We performed data correlation between the 40 bath chemistry variables. Through that correlation we chose seven input variables which, according to the process team and the available literature [9, 10], possess larger influence in the bath chemistry. Table 1 shows the pre-selected variables.

Table 1. Pre-selected variables

Variable	Symbol	Correlation with Fluoride
Amount of fed alumina	QAL	0.412
Bath Temperature	TMP	-0.88
Bath Fluoride Concentration	ALF	1
Amount of Fluoride Added into the bath	ALFA	-0,522
Cell Voltage	VMR	0,44
Cell Resistance	RMR	0,541
Metal Level	NME	0,23

From this analysis, we can confirm the opposite trend between bath temperature and aluminium fluoride [10].

3.2 Data Collection and Filtering

The main objective is to develop a system capable to estimate fluoride concentration using the variables listed in the table 1, so the fluoride concentration can be estimated only at times when all these variables are available. In the process, some variables are available real-time, while others require measurements schedule. Table 2 shows sampling periods for each input variable in the smelter where this work was performed.

Table 2. Sampling for the input variables

Variable	Symbol	Sampling Period
Amount of fed alumina	QAL	0.5 s
Bath Temperature	TMP	24-40 h
Bath Fluoride Concentration	ALF	56-72 h
Amount of Fluoride Added into the bath	ALFA	24-40 h
Cell Voltage	VMR	0.5 s
Cell Resistance	RMR	0.5 s
Metal Level	NME	24-40 h

So, in order to establish a fair sampling period under which all the variables should have available values, the sampling time should be the longest of all the model variables, i.e. the ALF variable sampling time. Then, data collection was performed comprising almost 3 months of measurements in 266 cells or 16160 records. The research was done assigning to any cell a time series containing all the selected variables' value in regular intervals observing the sampling time, i.e. the cells chosen for data collection remained working on during this period and were not switched off.

Having performed data collection, we performed filtering. According to Fortuna et al [3], the outliers should be left off. The outliers are not good source for the model construction, since they indicate plant's malfunctioning. We performed outlier detection by using each variable's regular operating range. Any value outside that range has been left out. When performed filtering, all the cells presenting outliers in their records were excluded, in order to avoid gaps in their time series, as can be seen in the figure 1.

Although the outliers occur in less than 1% of the records, when they are combined with other variables and cells, the amount of good data for modeling is reduced more than 60%. After filtering, 6395 records or 106 cells remained.

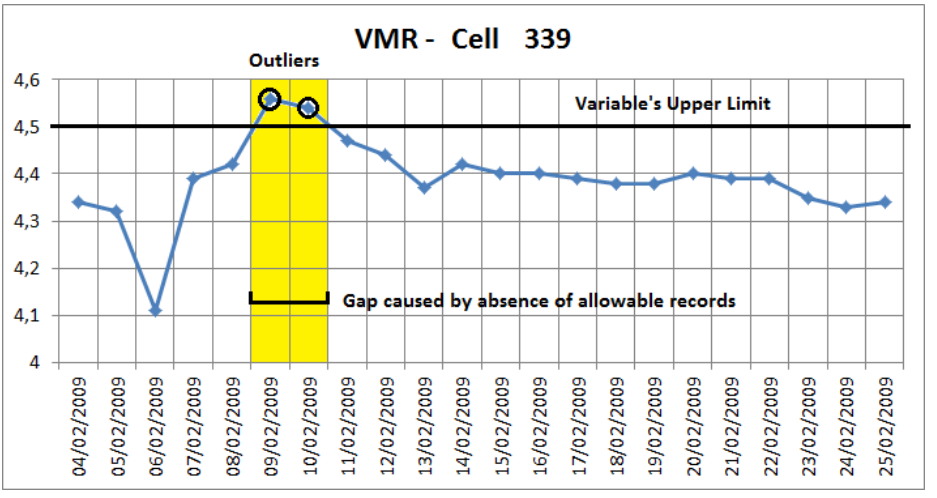


Fig. 1. Gap in the time series caused by filtering

3.3 Model Structure

The model's final structure should consider the variables and their delays [3]. When analyzing the delays' correlation of the selected variables, we decided to consider only 2 delays for ALFA, TMP and ALF.

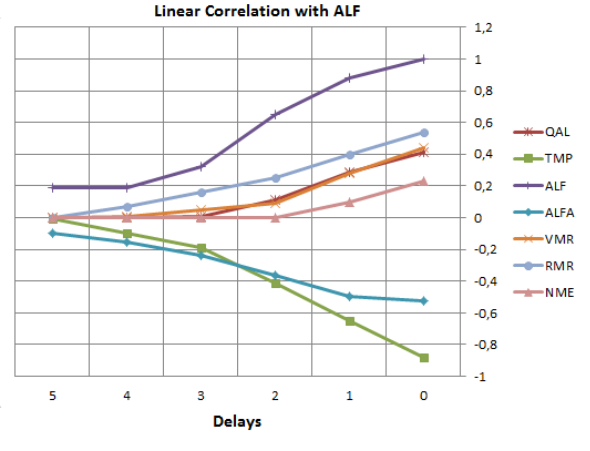


Fig. 2. Linear correlation of the selected variables and their delays with ALF

The delay of other variables was not considered since they do not represent any relevant information for the model. The fluoride estimator was schematically designed as shown in the figure 3.

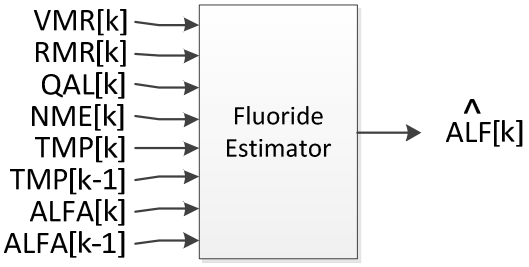


Fig. 3. Fluoride Estimator Final Structure

According to the findings of similar works on the subject and to the Fortuna's methodology, the model is based on neural networks, due to its capacity to generalization [3]. The chosen neural network architecture is the Multilayer Perceptron with 3 layers. The training algorithm chosen is the Levenberg-Marquardt, due to its fast error regression [11]. We defined the training, testing and validation sets as 70%, 13% and 17% respectively. Several neural network configurations were tested, under which the number of neurons in the hidden-layer ranged from 5 to 30. The activation function of each neuron was the hyperbolic tangent, except for the output layer, whose activation function was the linear function. Both the input and output values were normalized between -1 and 1, as recommended by the literature [11].

3.4 Results and Discussion

Every model was trained at least 20 times and at most 100 times, saving the best configuration, which is the one that show the lesser error MSE. All the models were built and tested in the MATLAB® software. The least error MSE is 0.01563, which means an average error 2.89% for ALF. This result is within the reliable interval for fluoride analysis (+/- 3%). Table 3 shows the best neural network architecture.

Table 3. Best Neural Network Configuration

Best Neural Network Configuration Found	
Layer	Neurons
1	6
Activation Function	Tang. Hiperb.
2	22
Activation Function	Tang. Hiperb.
3	1
Activation Function	Linear
Training epochs	6
Relative MSE error	0,01563
Average MSE error for ALF	2,89 %

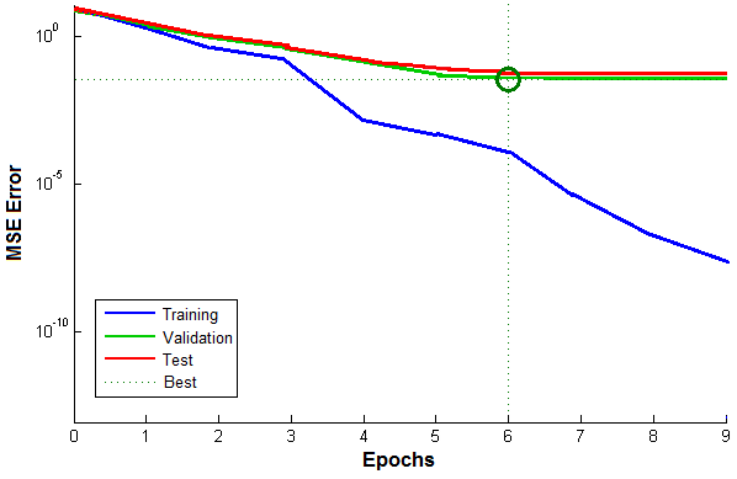


Fig. 4. Model Training Error

The dispersion and time series plots show how good are the inferences made by the estimator.

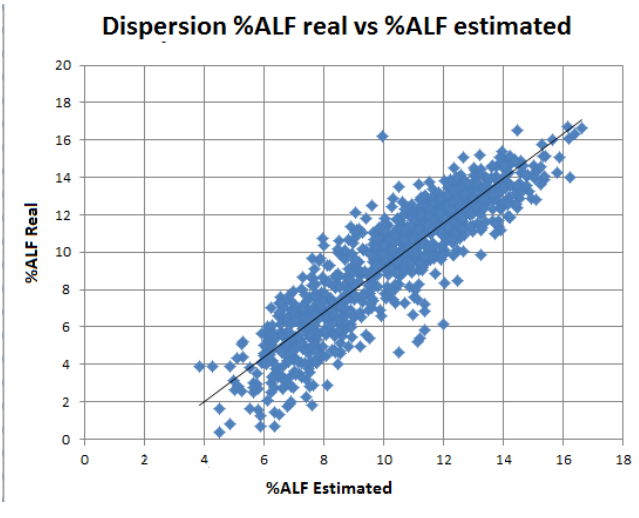


Fig. 5. Dispersion of records comparing the estimated %ALF to the real %ALF

Through these results we could confirm the correlation between the Fluoride Concentration and the selected input variables. The works developed by Branco [6] and Soares [8] have already proved the possibility of building neural models for aluminum smelting variables, and this fact has been confirmed in this work. It is known that all the variables dealt in this work have an associated behavior [10], so the initiative to forecast fluoride concentration is fully realizable, as can be seen through the results.

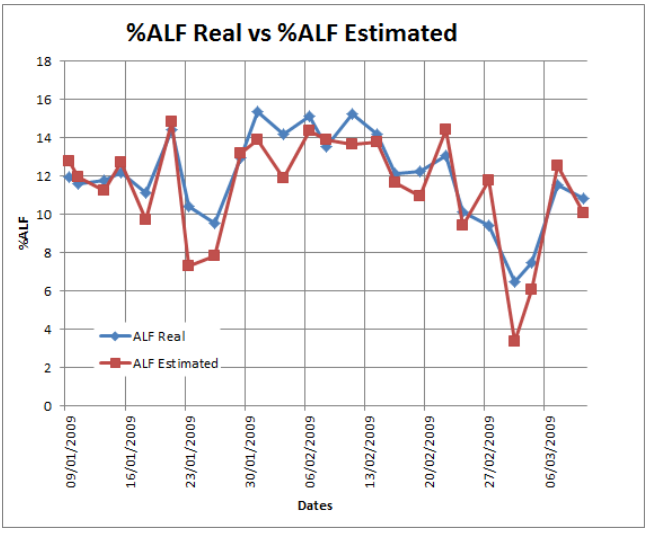


Fig. 6. Chart showing the real and estimated ALF time series

4 Conclusion

This work has proposed the modeling of fluoride concentration, an important variable for the bath chemistry control in the aluminium smelting process. Since this variable has a strong correlation with other available variables and requires a certain effort to measure it, its modeling is really desired and becomes realizable as it is well reliable on an established methodology. With this value available for the process control, many possibilities become feasible, such as bath chemistry simulation, bath chemistry control strategy essays, extend the same methodology to model other variables, to name a few. Also, this model could be used with analytical models using chemical or physical equations, thus making a hybrid approach considering both intelligent computing and theory.

References

1. Prasad, S.: Studies on the Hall-Héroult Aluminium electrowinning Process. Journal of Brazilian Chemistry Society 11(3), 245–251 (2000)
2. Grjotheim, K., Kvande, H.: Introduction to Aluminium Electrolysis Understanding the Hall-Héroult Process, 2nd edn. Aluminium-Verlag (1993)
3. Fortuna, L., Graziani, S., Xibilia, M.G.: Soft Sensors for Monitoring and Control of Industrial Processes, 1st edn. Springer (2007)
4. Frost, F., Karri, V.: Performance Comparison of BP and GRNN Models of the Neural Network Paradigm Using Industrial Application. In: Proc. 6th International Conference on Neural Information Processing, pp. 1069–1075 (November 1999)
5. Malaviya, A.V., Bundell, G.A.: An Intelligent Controller for Aluminum SmeltersPotlines. IEEE Transactions Industry App. 37(3), 792–805 (2001)

6. Branco, M.V., et al.: Model of Electric Resistance in Reduction Cells of Aluminum to be Applied on the Process Control. In: TMS Light Metals, Proceedings of the Technical Sessions, 135rd Technical TMS Annual Meeting, pp. 9–13 (2006)
7. Pereira, V.G., Oliveira, R.C.L., Soares, F.M.: Fuzzy Control Applied to the Aluminium Smelting Process. Fuzzy Logic Book 2. Intech Online (2012)
8. Soares, F.M., Oliveira, R.C.L.: Bath Temperature Inference through Soft Sensors using Neural Networks. In: Proceedings of Technical Sessions. TMS Light Metals, pp. 467–472 (2010)
9. Welch, B.J.: Aluminium Fluoride Consumption and Control in Smelting Cells. In: 6th International Conference on Molten Slags, Fluxes and Salts, Proceedings (June 2002)
10. McFadden, F.J.S., Welch, B.J., Austin, P.C.: The multivariable model-based control of the non-alumina electrolyte variables in aluminium smelting cells. Journal of the Minerals, Metals and Materials Society 58(2), 42–47 (2006)
11. Haykin, S.: Neural Networks, A Comprehensive Foundation, 2nd edn. Prentice-Hall (1999)

Discovering the Rules of a Elementary One-Dimensional Automaton

Erinaldo L. Siqueira Júnior, Tiago A.E. Ferreira, and Marcelo G. da Silva

Universidade Federal Rural de Pernambuco - UFRPE
Rua Dom Manoel de Medeiros, s/n, Dois Irmãos - CEP: 52171-900 - Recife/PE
erinaldo@barreiros.ifpe.edu.br,taef.first@gmail.com,mgds@cin.ufpe.br

Abstract. In this work we present a way to find the set of rules for the evolution of a one-dimensional cellular automata through its window of evolution using genetic algorithm, a search routine that mimics the behavior of the genetic evolution of living beings. As a result, we present the rules obtained by this strategy to the windows of the development of elementary automata rule 110 presented by Stephen Wolfram.

Keywords: Cellular automata, Genetic algorithm, Cellular space, Tessellation automata, Homogeneous structures, Cellular structures, Tessellation structures, Cellular automaton.

1 Introduction

Cellular automata (CA) are simple tools, but they can represent complex systems that present in its characteristic discrete elements with local interactions. Several studies have CA as a strategy for modeling physical, chemical, as well as phenomena from other areas. These studies always show the relations of interaction of the elements with its neighbors, either in a deterministic or stochastic way. On the other hand, what is possible to do if we do not know these relations of interaction with the neighborhood for a particular phenomenon? It is possible still mold it?

In an attempt to solve this problem, we use a search process known as genetic algorithm to find relations of interaction. The genetic algorithm consists of an optimization method that mimics the process of evolution of species, using genetic recombination, mutation and natural selection.

We will use the evolution sequence of the model presented by Wolfram [6] for which the search process will obtain their relations of interaction. And through that, we get the model itself, and knowing such interactions the model can be completely described. This result provides a formal basis to ensure that is possible to obtain the rules of a deterministic model if we have a sequence of its evolution to capture the dynamics of its interactions with its neighborhood.

This paper is divided as follows: Section 2 presents a brief history of cellular automata theory and some of the latest advances about this topic; in section 3 we explain the methodology used in our work adding the contributions of cellular automata and genetic algorithm in an attempt to discover the evolution rules

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 312–319, 2012.

of a CA through its evolutionary sequence. In section 4 we present the results obtained for the automaton model presented by Stephen Wolfram known as Rule 110. Finally, in section 5 we present some conclusions about the present work and point some directions to follow in future works about the study of automata.

2 Cellular Automata - CA

Cellular automaton is a discrete model studied in computability theory. It can be successfully applied in fields such as physics, mathematics, biology, computing and others in an attempt to model complex dynamics of systems that are unique to each of these areas.

At the beginning in the 60's, automata was studied as a dynamic system when its symbolic approach was first applied. In 1969, Hedlund presented results on your point of view and contributed significantly to the mathematical study of CA's. His greatest contribution is the theorem Curtis-Hedlund-Lyndon on the characterization of the set of automaton's global rules as a set of continuous endomorphisms [1].

In the 70's, a two-dimensional automaton named Game of Life was presented by Conway and published in the paper by Gardner [2]. Its remarkable characteristic are the occurrence of gliders, which are arrays of cells that traverse the automaton mimicking a live performance. As the result on the game of life, it was shown that is possible, based on this automaton, emulate a universal Turing machine [3].

In the book *Calculating Space*, written by Zuse in 1970, it was proposed that the physical laws of the universe are naturally discrete, and that the whole universe is the response of the deterministic calculation of a giant cellular automaton [4].

In 1983, physicist Stephen Wolfram published a basic class of cellular automata. Once noting the complexity that have achieved from simple automata, Wolfram is led to suspect that the complexity in natural events are constructed similarly. Furthermore, in this period Wolfram formulated the concept of intrinsic randomness and computational irreducibility, and suggested that Rule 110, defined bellow, should be universal; this fact came to be proved by the Wolfram research assistant, Matthew Cook at the 90's.

In 2002, Wolfram published a book entitled *A New Kind of Science* that presents the point of view that the discoveries about automata are not isolated incidents and have significance in all sciences. Even causing confusion in academia and the press, the book is not intended to provide a fundamental theory of physics based on automata. Although the book describing some physical phenomena through a model based on CA, also provided qualitatively different models of abstract systems [6].

In 2002, Christoph Durr *et al.* presented a paper showing the relationship between CA and communication. This work used cellular automata as a communicating grid of cells and attached a cell to separate the grid in two parts and presented the complex to carry communication among parts of the grid [7].

In 2011, Stefan Dantchev proposed a change on the dynamics of the relationship of any cell and its neighbors during the evolution of the automaton. To illustrate its modification, he presented a logarithmic time optimal solution to the problem of firing squad synchronization wich ensures that the modification may be feasible for some modeling [8].

Already in 2012, Sahu *et al.* presents in his article a set of rules to model the construction of molecules, showing how it is possible to use CA in theoretical chemistry [9].

As our main contribution, we will show that is relevant not only taken into account the direct neighbors, but also long-range neighbors.

Mathematically, we can define a CA as an array $M = M_t\,(c, d, v, E, \Theta)$ of length c and dimension d where its elements are themselves arrays of d positions, and each position is called a cell. The v-neighborhood of a cell x is defined as the set of cells distant from x up to v steps in an element of M. The set E contains all possible states that any cell can assume. The set Θ determines the interaction between a given cell and its v-neighborhood v defining their status in the next generation. And finally, the parameter t indexing the matrix M indicates the generation in which lies the model [6].

Although the automata has length c, there is communication between its ends, ensuring that the cells edge hae the same logical conditions given by the set Θ and thus is correlated with the neighborhood in the same way at the edges than in the interior of the automaton.

To illustrate the process we introduce a cellular automaton model presented by Stephen Wolfram called rule 110 [6]:

$$M_{100} = (101, 1, 1, \{\circ, \bullet\}, \Theta) \tag{1}$$

in which

$$\Theta = \{(\bullet\bullet\bullet\circ),(\bullet\bullet\circ\bullet),(\bullet\circ\bullet\bullet),(\bullet\circ\circ\circ),(\circ\bullet\bullet\bullet),(\circ\bullet\circ\bullet),(\circ\circ\bullet\bullet),(\circ\circ\circ\circ)\} \tag{2}$$

and the last "dot" is calculated as a function of the three previous ones.

In this example, the matrix M of length 101 is a one-dimensional automata with neighborhood up to 1 cell that has direct connection with the cell under study, that is, only cells bordering the side with the cell under study and the cell under study will be considered to evolve it to the next generation. The set $E = \{\circ, \bullet\}$, presents two possible states for the cells of M in this case, and can be interpreted as $\bullet = ON$ and $\circ = OFF$. The set Θ is a collection of interaction rules with neighbors of the cell defined by v. In a rule as $(\bullet\bullet\bullet\circ)$, $\bullet\bullet\bullet$ shows the behavior of states of the cell under study and its two neighbors, at left and right, in the current generation t, and last entry $\circ$ shows the configuration of the cell under consideration in the next generation $t + 1$.

The figure 1 shows the evolution of the matrix M through generations of $t = 1$ to $t = 100$. The left figure was obtained from a called *single one* individual $0 \cdots 010 \cdots 0$ and the right one from a randomly generated individual.

In this context, we can generate an image through the evolution of one-dimensional automaton, however, one question arises: Once known variables

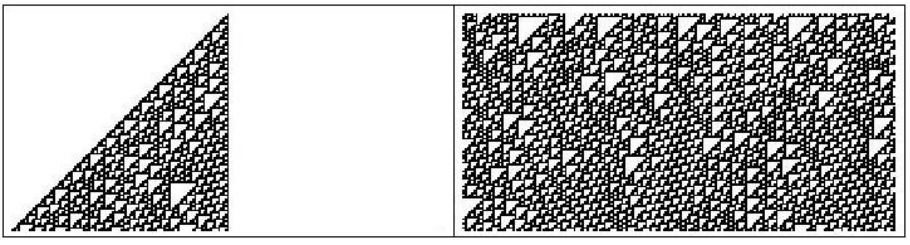


Fig. 1. Evolution of Rule 110 up to generation 100: Single 1 (left) and random (right)

c, d, v and E of the automaton, it is possible to predict the set Θ of rules? That is, once I have the image can I determine the rule that made the automaton evolve to the present moment? This is the objective of this work: give a response to this question.

3 Methodology

Cellular automata are known for their fixed rules (or dynamic) that evolve under a given scenario. However, they can be evaluated under a different context in which a scenario can be interpreted as a result of the evolution of an automaton and our interest is to determine the rules leading to this behavior. It is this situation in which pattern recognition and search algorithms are necessary to find the rules that make a scenario evolve to our target situation.

Vanneschi [10] presents results in wich, given certain conditions, a one-dimensional automata using genetic algorithm has the ability to find the most general and robust solution. Results in the literature use other cellular automata as classifiers such as the *Multiple Attractor Cellular Automata (MACA)*, or search algorithms for maximizing a target function [11]. Mitchell presents a study on the feasibility and use of automatons for classification and synchronization that is the basis for our current work [12].

Following the principle of a genetic algorithm we will generate a population of x individuals, and each individual consisting of y chromosomes (which in our context will be the rules, ie, $\bullet\bullet\bullet\circ$ is a chromosome).

Each chromosome consists of genes; in our encoding 4 genes per chromosome. Hence, each individual will consist of exactly $4y$ genes.

Previous works show that is convenient use a correct amount of individuals per population, because this parameter affects the performance of the algorithm. Due to this, a preliminary study was done trying to get the average size for the population in an attempt to obtain the average performance of algorithm for each cellular automata applied [13,14].

As the chromosomes for our problem has symbolic representations of $\circ$ or $\bullet$, we map these symbols in a unique way, namely, $(\circ = 0)$ and $(\bullet = 1)$. Thus, for example, the rule $\bullet\bullet\bullet\circ$ is mapped in 1110. Therefore, an individual can be

rewritten as a binary vector with $4y$ entries, and finally, the population will be a binary matrix with exactly $4xy$ entries.

To fill the array of individuals that start the algorithm, we use samples of a random variable following a Bernoulli distribution with $p = 1/2$:

$$X \sim Be\left(\tfrac{1}{2}\right)$$

$$P\left(X = 0\right) = P\left(X = 1\right) = 0.5$$

Thus we have completely random individuals made up of chromosomes that determine the interaction with its neighbors for each evolution.

After this step, each individual will be evolved to achieve evolution equal to the target scenario. We compare the evolved individuals against the development of the target scenario and record the amount of error of matching, and then we measure the fit (fitness) for each individual using the expression:

$$fitness_{individual(i)} = \frac{1}{1 + error} \tag{3}$$

where *error* is the amount of entries in the window that matching does not agree with the target scenario. Note that the adjustment varies over the range:

$$fitness_{individual(i)} \in (0, 1]$$

which can be indirectly associating as the likelihood of success of the individual in question against the target scenario.

Since all subjects already have their value set against the scenario target, it is necessary to define those to be used to generate individuals-children. At this point we use the criterion of roulette:

1 - All fitness of the individuals tested are added, and we divide the fitness of each individual by this sum to obtain the relative probability fitness for the individual in question against the other individuals:

$$\rho_{individual(i)} = \frac{fitness_{individual(i)}}{\sum_{i=1}^{x} fitness_{individual(i)}} \tag{4}$$

2 - Calculate the accumulated probability, and a sample is drawn from a uniform distribution $Z \sim U\left(0, 1\right)$, and two individuals-parents are selected.

This confirms the condition of the evolution that the fittest will be selected without, however, disregard the possibility of less fit individuals also generate individuals-children.

3 - Immediately after selected, individuals-parents passing through the crosslinking process of genetic information, ie the transfer of genes. For this, we define the crossover point in the vector of individuals-parents through a random number that can vary from 1 to $4y - 1$. The diagram below illustrates the process of crossing (*crossover*) to a crossing point of value 2:

$$\begin{array}{c} 11|\,01101\cdots10 \text{ and } 10|\,11011\cdots11 \\ \swarrow \qquad\qquad \searrow \\ 11|\,11011\cdots11 \text{ and } 10|\,01101\cdots10 \end{array}$$

for individuals-children it should be checked if their fitness are better than the worst fitness of the population; if yes, the individual-children take the place of the worst individuals in the population, if not, we do a mutation at mutability rate α.

3a- In our case we are using a heavy mutation, ie, whenever an individual mutates he will have all its entries modified:

$$1111011\cdots11 \overset{Mutation}{\rightarrow} 0000100\cdots00$$

After calculated the new value by adjusting ($fitness$) and re-checked if the value of current fitness is better than the worst individual in current population and replacing it if it does, however, otherwise with a probability β the individual-children enters the current population even if not better than the worst individual. Obviously, the value of β is very small, however, such action is necessary to ensure that all individual-children have the possibility of entering the population even this do not improve it at the moment.

The algorithm repeats steps 1, 2, 3 and $3a$ until find an individual with fitness 1 or until it reaches 10,000 repetitions.

4 Results

In this work we encode the automata presented in the book written by Stephen Wolfram as a search problem using genetic algorithms. Tested the success of the method to discover the rule of relationship with the neighborhood for several encodings presented by Wolfram, we present here the evolution of the *fitness* function for the rule 110 automaton that measures how close we are the optimal solution.

In the figure 2 is shown the evolution of *fitness* function under the generation of the population. It is possible to realize the time in which a model (automaton) acquires coordinate 1 indicating the current configuration necessary to capture the dynamics of interaction with the neighborhood.

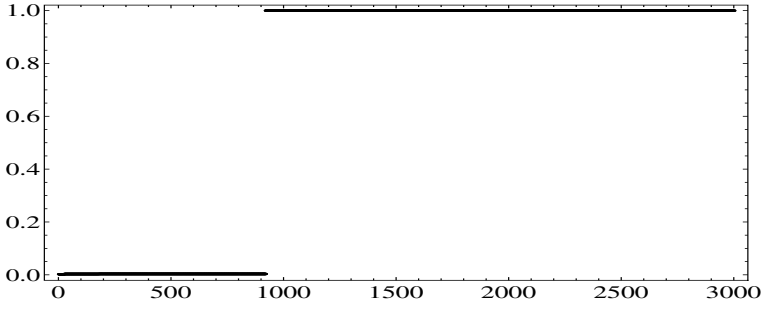


Fig. 2. Fitness for Rule 110 automaton

For each model tested it was used individuals of size 60; that is, capable of storing a class with 15 chromosomes. Note that each automaton has a composition of 8 chromosomes (sinse they have the form $a_1a_2a_3b$ and $a_i \in \{0, 1\}$), and so it is necessary repetition of chromosomes in the individuals tested. This strategy was necessary because we do not know the number of chromosomes that the models present and then there would be no guarantee that an individual formed by 8 chromosomes would be sufficient.

The populations used in each of the simulations have size 300, with selection of 20 individuals by the called tournament method and a mutation probability of 2%.

In the table 1 we present the statistics for 30 repetitions of some Wolfram Rules automata simulation. The high standard deviation is an inherent characteristic of stochastics algorithms like GA. This is a common behavior for all elementary Wolfram automata. We can note also some asymmetry in confidence interval due to large value for σ and a negative value at lower limit for $\mu \pm 1.96\sigma$. This is inconsistent with generations characteristic and so we do a cutoff to threshold 0.

Table 1. Confidence interval containing 95% of cases in which the model achieved fitness 1 in presenting the dynamic relationship with the neighborhood

Automata (Rule)	Average	Standard deviation	Confidence Int. (95%)	Automata (Rule)	Average	Standard deviation	Confidence Int. (95%)
18	38	19	(1, 74)	73	918	965	(0, 2809)
30	799	576	(0, 1978)	10	929	780	(0, 2457)
54	1204	1064	(0, 3290)	126	925	638	(0, 2175)
60	1110	561	(10, 2210)	150	1719	1278	(0, 4224)
62	986	984	(0, 2914)	222	217	156	(0, 523)

5 Conclusions

This result proves the efficiency in the use of genetic algorithms to obtain the interactions with the neighborhood of one-dimensional deterministic automata.

Each of the rules in the Wolfram book shows value 1 obtained for the *fitness*, which indicates how the method was adequate to capture the pattern of automata.

The current research motivates us to the following future work:

- Using genetic algorithm for possible automata based on the initialization vector and the final vector of model evolution, trying to find automata starting and ending in these respective vectors. This is interesting because would not be necessary to know the whole evolution of the model under study to make a complete description, but only its starting and ending vectors;

- Development of a strategy under genetic algorithms to capture the interactions with the neighborhood of one-dimensional stochastic model. The current

results apply to one-dimensional deterministic automata. Thus, the development of this strategy would be crucial in moving to our goal that is to present a methodology to treat any time series, particularly the financial ones, as a one-dimensional stochastic automata and through this approach to find interactions with neighbors, and thus make predictions.

References

1. Hedlund, G.A.: Endomorphisms and automorphisms of the shift dynamical system. Theory of Computing Systems 3(4), 320–375 (1969)
2. Gardner, M.: Mathematical games: The fantastic combinations of John Conway's new solitaire game life. Scientific American 223, 120–123 (1970)
3. IGBLAN: Life universal computer, http://www.igblan.free-online.co.uk/igblan/ca
4. Zuse, K.: Calculating Space. Project MAC Report, MIT Technical Translation (1970)
5. Wolfram, S.: Statistical mechanics of cellular automata. Reviews of Modern Physics 55, 601–644 (1983)
6. Wolfram, S.: A new kind of science. Wolfram Media, Illinois (2002)
7. Dürr, C., Rapaport, I., Theyssier, G.: Cellular automata and communication complexity. Theoretical Computer Science 322(2), 355–368 (2004)
8. Dantchev, S.: Dynamic neighbourhood cellular automata. The Computer Journal 54(1), 26–30 (2011)
9. Sahu, S., Oono, H., Ghosh, S., Bandyopadhyay, A., Fujita, D., Peper, F., Isokawa, T., Pati, R.: On cellular automata rules of molecular arrays. Natural Computing 11(2), 311–321 (2012)
10. Vanneschi, L., Mauri, G.: A study on learning robustness using asynchronous 1D cellular automata rules. Natural Computing 11(2), 289–302 (2012)
11. Maji, P., Shaw, C., Ganguly, N., Sikdar, B.K., Pal Chaudhuri, P.: Theory and Application of Cellular Automata For Pattern Classification. Fundamenta Informaticae 58(3-4), 321–354 (2003)
12. Mitchell, M., Crutchfield, J.P., Das, R.: Evolving cellular automata with genetic algorithms: A review of recent works. In: Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA 1996). Russian Academy of Sciences, Moscow (1996)
13. Koumousis, V.K., Katsara, C.P.: A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. IEEE Transaction on Evolutionary Computation 10(1), 19–28 (2006)
14. Koljonen, J., Alander, J.T.: Effects of population size and relative elitism on optimization speed and reliability of genetic algorithms. In: Proceedings of the 9th Scandinavian Conference on Artificial Intelligence, pp. 25–27 (2006)

Perceptron Models for Online Structured Prediction

Maurício Archanjo Nunes Coelho, Raul Fonseca Neto,
and Carlos Cristiano Hasenclever Borges

Universidade Federal de Juiz de Fora, Programa de Pós-Graduação em Modelagem Computacional, Minas Gerais, Juiz de Fora, Brasil
mauarchanjo@hotmail.com
raulfonsecaneto@ig.com.br
cchb@lncc.br

Abstract. Our structured prediction problem is formulated as a convex optimization problem of maximal margin [5-6], quite similar to the formulation of multiclass support vector machines (MSVM) [8]. It is applied to predict costs among states of paths. Predicting them properly is very important, because the problem of paths planning depends on its correctness. Ratliff [4] showed a maximum margin approach which allows the prediction of costs in different environments using subgradient method. As a contribution of this work, we developed new solution methods: the first one, called Structured Perceptron, has similarities with the correction scheme proposed by [1] and the second one is called Structured IMA. It is derived from the work presented by [2]. Both use the Perceptron model. The proposed algorithms were more efficient in terms of computational effort and similar in prediction quality when compared with [4].

Keywords: Predicting Structured Data, Perceptron, Planning Paths.

1 Introduction

The objective of this work is presenting and evaluating two new methods for solving the structured prediction problem applied to the maximum margin approach, with their respective developments and examples. To demonstrate that our methods are applicable, we use them to predict costs in new environments defining a functional learning between input and output domains, structured and arbitrary. We obtain the learning plans from the perspective of the maps features. This is of great importance in use in navigation systems for mobile robots [4]. Often, there is a clear distinction between the levels of perception and planning, which is gotten only from the prior knowledge of the matrix of costs related to the action-state space of the problem.

We have as input a set of paths on maps chosen by an expert. These paths are selected to benefit some strategy related to the presence or absence of features. Thus, the strategy learned will enable the planning of new paths in similar environments according to the type of strategy picked by the expert. This problem of learning is formulated as a convex optimization problem of maximum margin and their structure is very similar to the formulation of MSVM [8]. As a solution method we primarily implement the MMP algorithm [4]. Alternatively, we propose the algorithms:

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 320–327, 2012.

Structured Perceptron and Structured IMA. In order to prove the efficiency of this new approach and its correctness we performed tests with diverse inputs.

2 Structured Prediction Model

2.1 Maximum Margin Approach

Given a training set $S = \{(x(i), y(i)), i = 1, ..., m\}$, each pair is formed by a sample represented by a structured object $x(i)$ (map) and a desired solution $y(i)$ (path). It is intended to obtain a parameters vector w such that:

$$\arg Max_{y \in Y(i)}\{w^T.((x(i),y)\} \approx y(i), i = 1,\ldots,m\,, \tag{1}$$

where $Y(i)$ is the space of all possible solutions dependent structured object $x(i)$. The cardinality of $Y(i)$ can be very high, but it is possible to use techniques that eliminate efficiently the false examples y. That is, learning the parameter vector w allows the best solution for each pair $(x(i),y)$. This is exactly the solution $y(i)$ proposed in the training set. The approach to solving this problem is based on a generalization of the principle of maximum margin [5-6] used in support vector machine [7-8] and embraces the solution of the following quadratic programming problem:

$$\begin{aligned} &Min\tfrac{1}{2}||w||^2 \\ &Subject\ to \\ &w^T.f_i(y(i)) \geq Max_{y \in Y(i)}\{w^T.f_i(y) + l_i(y)\}, i = 1,\ldots,m\,, \end{aligned} \tag{2}$$

where $f_i(y) = f(x(i),y)$ and the function $l_i(y) = l(y(i),y)$ is defined as a loss function that scales the geometric value of the margin. In the multi-class SVM [8], the margin γ_i of a sample $(x(i),y(i))$ over another $y \in Y(i)$ is interpreted as:

$$\begin{aligned} \gamma_i &= Min_{y \in Y(i), y \neq y(i)}(w^T.f_i(y(i)) - w^T.f_i(y))/||w||_2 \\ &= (w^T.f_i(y(i)) - Max_{y \in Y(i), y \neq y(i)}\{w^T.f_i(y)\})/||w||_2\,. \end{aligned} \tag{3}$$

The maximum margin γ of the whole problem is the value of smallest γ_i, $i = 1, ..., m$. Consequently, minimizing $||w||$ or equivalently maximizing γ results in obtaining a solution of maximum margin [5]. For cases of non-separability can admit the introduction of slack variables. Note that the problem $Max_{y \in Y(i)}\{w^T.f_i(y)+l_i(y)\}$ has precisely the same way the problem of predicting which is needed to learn the parameters.

2.2 Problem of Predicting Solution Cost

The training set is given to $T = \{(F_i, \mu_i), i = 1, ..., m\}$. The path μ_i is chosen by the expert to the map F_i. Each map is represented by a features matrix F with all state-action pairs and each state is characterized by a features set. Each path is represented by a frequencies vector μ indicating the presence or absence of actions in each possible state-action pair. The product $F_i.\mu$ represents the quantity of each feature in the nth map depending of the path represented by μ. Finally, the product $w^T.F_i\mu$ represents the total reward of the path μ. The equation of this problem was presented by Ratliff [4]:

$$\begin{aligned} &Min ½||w||^2 \\ &Subject\ to \\ &w^T.F_i\mu_i \geq Max_{\mu \in G(i)}\{w^T.F_i\mu + l_i^T.\mu\},\ i=1,\ldots,m\,, \end{aligned} \tag{4}$$

where *G(i)* is the representation of all possible choices of paths for the *n*th map. Without loss of generality we chose to study this problem by minimizing costs, in this case the constraint is: $w^T.F_i\,\mu_i \leq Min_{\mu \in G(i)}\{w^T.F_i\mu - l_i^T\mu\}$, *i=1,..., m.*

3 Solution Methods

3.1 MMP Algorithm

To solve the problem of maximizing rewards presented in [4], Ratliff proposed to minimize an objective function not differentiable, but convex, obtained from the relaxation of the problem, using a subgradient technique:

$$L(w) = ½||w||^2 + C\sum_i Max_{\mu \in G(i)}\{w^T.F_i\mu + l_i^T\mu\} - w^T.F_i\mu_i\,. \tag{4}$$

C is a positive regularization parameter. This function has as sub-gradient:

$$g = w + C(\sum_i F_i\mu^* - F_i\mu_i)\,, \tag{5}$$

Where μ^* is a optimal path for each input map obtained from the equation: $\mu^* = arg Max_{\mu \in G(i)}(w^T.F_i + l_i^T\mu)$. Then, the A* algorithm is used, it has quadratic complexity in relation to the number of states. Thus, the rewards vector *w* is updated as:

$$w \leftarrow w - \eta g,\ or\ w \leftarrow w.(1-\eta) - \eta(C(\sum_i (F_i\mu^* - F_i\mu_i))\,. \tag{6}$$

Learning rate η is reduced gradually according to the number of iterations.

3.2 Structured Perceptron

If the condition of maximum margin is removed, we can reformulate the problem of minimizing the cost prediction as a problem of viability of a system of inequalities. Then, we find a feasible costs vector *w*, such that:

$$w^T.F_i\mu_i - Min_{\mu \in Q(i), \mu \neq \mu_i}\{w^T.F_i\mu\} \leq 0,\ i=1,\ldots,m\,, \tag{7}$$

where *Q(i)* represents the set of existing paths μ_i. To solve this problem we can use a variant of the primal perceptron algorithm which was named Structured Perceptron[1]:

[1] Bakir [1] and McDonald [3] presented algorithms for Structured Perceptron derived directly from the formulation of multiclass perceptron; alternatively, we demonstrate a direct derivation of the original model of the perceptron, reaching a slightly different algorithm.

$$IF\ (w^T.F_i\mu_i - Min_{\mu\in Q(i),\mu\neq\mu_i}\{w^T.F_i\mu\}) > 0$$
$$w \leftarrow w - \eta(F_i\mu_i - F_i\mu^*), 0 < \eta \leq 1, \tag{8}$$

i.e. if the inequality (11) is not satisfied, the path chosen does not have a lower cost or equal than the best alternative, then update the vector w. η is a constant learning rate. The vector μ^* is determined by direct comparison of the set of paths proposed in the training set T and it is always the best path disregarding the expert's path:

$$\mu^* = \arg Min_{\mu\in Q(i),\mu\neq\mu_i}\{w^T.F_i\mu\}. \tag{9}$$

The advantage of using the *Q(i)* and not each possible path in *G(i)* is the reduction of computational effort, the complexity is linear: maps in the training set. The rule of correction obtained for this algorithm can be easily derived. We have the difference vectors set for each map collection in the form:

$$\delta_i = F_i\mu_i - F_i\mu^*. \tag{10}$$

The condition of viability of the Perceptron algorithm is defined as:

$$-w^T.\delta_i \geq 0, i = 1,\ldots,m. \tag{11}$$

Thus, the loss function related to the cost minimization strategy is given by:

$$J(w) = \sum_i Max\{0, w^T.\delta_i\}. \tag{12}$$

This function should be minimized. Thus, we can define the local gradient as:

$$\nabla_w J(w) = \delta_i = F_i\mu_i - F_i\mu^*. \tag{13}$$

Therefore, if an error occurs related to the nth sample:

$$-w^T.\delta_i < 0\ or\ w^T.(F_i\mu_i - F_i\mu^*) > 0, \tag{14}$$

we use the following correction rule for the vector of costs:

$$w \leftarrow w - \eta(F_i\mu_i - F_i\mu^*), 0 < \eta \leq 1. \tag{15}$$

Similarly, for a maximization problem of rewards, we have:

$$w \leftarrow w - \eta(F_i\mu^* - F_i\mu_i), 0 < \eta \leq 1. \tag{16}$$

Notice the similarity between the equation (20) and the correction rule (9) of MMP.

3.3 Structured Incremental Margin Algorithm (IMA)

With the addition of margin in the problem[2], we find a viable vector w, such that:

$$w^T.F_i\mu_i \leq Min_{\mu\in Q(i),\mu\neq\mu_i}\{w^T.Fi\mu\} - \gamma\|w\|_2, i = 1,\ldots,m. \tag{17}$$

[2] Note that we do not use the arbitrary loss function l_i^T with the intention to scale the margin instead we use the straightforward definition of the margin equation (3).

Or, alternatively:

$$w^T.F_i\mu_i - Min_{\mu\in Q(i),\mu\neq\mu_i}\{w^T.F_i\mu\} \leq -\gamma \parallel w \parallel_2, i = 1,\ldots,m\,. \tag{18}$$

The correction rule for the Structured IMA follows the same reasoning presented for the Structured Perceptron. The difference vector for each map is defined as:

$$\delta_i = F_i\mu_i - F_i\mu^*\,. \tag{19}$$

The new condition of feasibility of IMA Structured algorithm is given by:

$$-w^T.\delta_i \geq \gamma \parallel w \parallel_2 \; or \; -(w^T.\delta_i + \gamma \parallel w \parallel_2) \geq 0, i = 1,\ldots,m\,. \tag{20}$$

Similarly, the loss function related to cost minimization strategy is given by:

$$J(w) = \sum_i Max\{0, w^T.\delta_i + \gamma \parallel w \parallel_2\}\,. \tag{21}$$

This function will be minimized. The local gradient for the *n*th sample is defined as:

$$\nabla_w J(w) = (\gamma.w)/\parallel w \parallel_2 + \delta_i = (\gamma.w)/\parallel w \parallel_2 + (F_i\mu_i - F_i\mu^*)\,. \tag{22}$$

Thus, if an error occurs related to the *n*th sample:

$$-w^T.\delta_i < \gamma \parallel w \parallel_2 \; or \; w^T.(F_i\mu_i - F_i\mu^*) > -\gamma \parallel w \parallel_2\,, \tag{23}$$

we use the following correction rule for the vector of costs:

$$w \leftarrow w.(1 - \eta\gamma/\parallel w \parallel_2) - \eta(F_i\mu_i - F_i\mu^*), 0 < \eta \leq 1\,. \tag{24}$$

Similarly, for a maximization problem of rewards, we have:

$$w \leftarrow w.(1 - \eta\gamma/\parallel w \parallel_2) - \eta(F_i\mu^* - F_i\mu_i), 0 < \eta \leq 1\,. \tag{25}$$

In this new correction rule occurs the control of the value of the norm. This is necessary due to restriction imposed by the margin. To calculate the approximate value of the maximum margin we adopt an incremental approach in which successive systems of inequalities (22) are solved. It begins with the Structured Perceptron computing γ according to the equation (3), and the next γ gotten doubling the previous γ. The values of the cost vector w are retained and serve as the initial solution to the posterior problem each time we double γ. If this value exceeds the maximum margin, a binary search between the feasible margin and the infeasible margin should be performed.

4 Experimental Results

To validate the algorithms developed we chose to solve a practical problem of planning related to the determination of paths in a grid map (discretized Google maps). It was defined as a costs minimization problem. Each cell may or may not have the presence of tree features (*w[0]*: undergrowth, *w[1]*: trees and *w[3]*: pathway) and their combinations, defining eight different types of ground according to the cardinality of the set power. Six example maps are used in the training set and in the test set.

Figure 1 shows the maps used in the training set and the black paths chosen by the expert. We can see that it is usually better we choose a path without undergrowth and trees. However, if the path requires a greater track, the specialist may suggest passing through undergrowth or trees. All the algorithms converge and reflect the planning strategy adopted by the expert. We define 1000 iterations to MMP algorithm and Structured IMA. As MMP algorithm parameters are used: $\eta = 0.5$ and C = 1. The loss function l_i is the Hamming distance between μ_i and μ. Runtime: 15.919 seconds. Parameter used to Structured Perceptron: $\eta = 0.2$, converge with 10 iterations and runtime: 0.082 seconds. In Structured IMA $\eta = 0.2$ too and runtime: 6.102 seconds.

Fig. 1. Training maps with their respective paths chosen by an expert. Google maps were discretized (55x55) and simplified to embrace our eight different types of ground in each cell.

The choice of the expert reflects the best way when compared to all alternatives in the training set. To demonstrate this, we prepared three tables with the geometric costs of the paths. They are obtained by dividing the cost of the respective paths by the norm. Also we show the values of margins that represent a measure of planning quality. MMP got: *w[0]*: 0.512160, *w[1]*: 0.462381, *w[2]*: -5.283517, with norm: 5.328383 and margin: 0.130939. Structured Perceptron got: *w[0]*: 0.900000, *w[1]*: 0.300000, *w[2]*: -5.30000 with norm: 5.384236 and margin: 0.074291. Structured IMA obtained: *w[0]*: 1.124728, *w[1]*: 0.784634, *w[2]*: -4.777393 with norm: 4.970327 and final margin: 0.168532. The margin value was defined as the minimum difference among the geometric costs of the expert's paths and the alternative paths of lower cost. These values are shown in bold in the respective tables.

Table 1. Geometric costs MMP algorithm

Map	Path 1	Path 2	Path 3	Path 4	Path 5	Path 6	Margin
1	**2.1841**	5.3702	**4.1807**	6.6904	4.4914	14.8344	**1.9965**
2	4.3685	**0.9630**	4.1089	**1.2101**	5.5523	11.3827	**0.2448**
3	4.2574	5.2741	**1.1459**	4.5357	**3.5148**	6.7055	**2.3688**
4	4.6025	5.3608	**1.3888**	**1.2579**	3.8088	12.3780	***0.1309***
5	2.1424	5.1099	**1.8885**	4.7035	**1.0097**	5.5103	**0.8788**
6	3.2751	**2.9912**	4.1957	5.8630	4.4346	**1.7917**	**1.1994**

Table 2. Geometric costs Structured Perceptron algorithm

Map	Path 1	Path 2	Path 3	Path 4	Path 5	Path 6	Margin
1	**2.2287**	5.4789	**4.2717**	6.9833	4.5874	15.1739	**2.0430**
2	4.3645	**1.0957**	4.2160	**1.1700**	5.6832	11.6079	***0.0742***
3	4.1974	5.3118	**1.3186**	4.5688	**3.6031**	6.6676	**2.2844**
4	4.7546	5.3675	**1.5415**	**1.3929**	3.9931	12.8151	**0.1485**
5	2.1730	5.3118	**1.9129**	4.7917	**1.0215**	5.5718	**0.8914**
6	3.3245	**3.0645**	4.2717	5.9989	4.5317	**1.9501**	**1.1143**

Table 3. Geometric costs algorithm Structured IMA

Map	Path 1	Path 2	Path 3	Path 4	Path 5	Path 6	Margin
1	**3.3829**	6.6112	**5.2659**	8.3090	5.9157	18.3305	**1.8829**
2	5.5315	**1.9982**	5.1080	**2.2138**	6.9453	14.4523	**0.2156**
3	5.3052	6.3849	**2.3823**	5.8656	**4.9545**	9.0026	**2.5721**
4	6.1420	6.5428	**2.7665**	**2.5979**	5.5649	15.5504	***0.1685***
5	3.2250	6.1376	**2.7331**	6.2498	**2.0375**	7.7257	**0.6955**
6	4.4125	**3.9205**	5.2659	7.5951	5.7578	**3.6897**	**0.2308**

Figure 2 shows the results of the algorithms applied to the test set. The black solid line represents the path based on the cost vector of the algorithm MMP. In most the maps the paths to all algorithms agreed. However, when they do not coincide, we establish other paths with different shapes for the identification of the others two algorithms. The red dotted path represents the Structured Perceptron, maps three and six. The yellow path with squares represents the Structured IMA, map six. All of them show that there is a strong association between the outputs represented by the costs and the expert's strategy. This confirms that this new approach is an efficient alternative when compared to the MMP. Also we can see that the maximizing of margin originates different paths for some maps, mainly where the pathway is not clear.

Fig. 2. Tests maps with the paths defined by the A* algorithm based on the costs vectors *w*

5 Conclusion

In this work we have presented two new algorithms for solving structured prediction problems. According to the results, the algorithms behave similarly, but our algorithms require a lower cost of memory and processing, because we do not need to run the A* algorithm in the learning process. It predicts effectively the costs on new environments from a small number of training maps, enabling their use in other larger applications. It is important to note that the algorithm IMA Structured obtains a final margin value above the margin value of the MMP algorithm. This demonstrates that our new approach to solve the presented problem is efficient and quite promising.

References

1. Bakır, G., Hofmann, T., Schölkopf, B., Smola, A., Taskar, B., Vishwanathan, S.V.N.: Predicting structured data. MIT Press, Cambridge (2006)
2. Leite, S., Fonseca Neto, R.: Incremental margin algorithm for large margin classifiers. Neurocomputing 71, 1550–1560 (2008)
3. McDonald, R., Hall, k., Mann, G.: Distributed Training Strategies for the Structured Perceptron. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 456–464. Association for Computational Linguistics, Stroudsburg (2010)
4. Ratliff, N., Bagnell, J.A., Zinkevich, M.: Maximum margin planning. In: 23rd International Conference on Machine Learning, pp. 729–736. ACM, New York (2006)
5. Taskar, B., Chatalbashev, V., Koller, D., Guestrin, C.: Learning structured prediction models: a large margin approach. In: 22nd International Conference on Machine Learning, pp. 896–903. ACM, New York (2005)
6. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. J. Mach. Learn. Res. 6, 1453–1484 (2005)
7. Vapnik, V.: Statistical learning theory. John Wiley and Sons, Inc., New York (1998)
8. Weston, J., Watkins, C.: Multi-class support vector machines. Technical Report, CSD-TR-98-04 (1998)

A Weightless Neural Network-Based Approach for Stream Data Clustering

Douglas Cardoso[1,⋆], Massimo De Gregorio[2], Priscila Lima[3], João Gama[4,⋆⋆], and Felipe França[1]

[1] Universidade Federal do Rio de Janeiro, PESC-COPPE, Brasil
[2] Istituto di Cibernetica "E. Caianiello" - CNR, Pozzuoli (NA), Italy
[3] Universidade Federal Rural do Rio de Janeiro, DEMAT-ICE, Brasil
[4] Universidade do Porto, LIAAD-INESC, Portugal

Abstract. One of the major data mining tasks is to cluster similar data, because of its usefulness, providing means of summarizing large ammounts of raw data into handy information. Clustering data streams is particularly challenging, because of the constraints imposed when dealing with this kind of input. Here we report our work, in which it was investigated the use of WiSARD discriminators as primary data synthesizing units. An analysis of StreamWiSARD, a new sliding-window stream data clustering system, the benefits and the drawbacks of its use and a comparison to other approaches are all presented.

Keywords: StreamWiSARD, Clustering, Data Mining, Data Streams, Sliding Window, WiSARD, Weightless Neural Networks.

1 Introduction

Nowadays computing assumed a major role in human life. One consequence of this integration between life and computing is the availability of massive amounts of data from all kinds of different sources, which represent what is happening in the world in a certain period in time. To extract information from what is raw data is the main target of data mining. However, due to some harder constraints we have stream data mining, where information has to be continually extracted and updated, as an expression of what has happened lately. This way, a stream data clustering system should be able to cope with changes in data like the disappearance of clusters or their movement.

A great variety of methods to cluster stream data have been presented recently, based on different ideas and techniques: microclusters [8,3], density clustering [2], graphs [9], hierarchical clustering [2,8], sliding [1] and damped window [2,8] time data models. All of those are valuable contributions, but there is still room for improvement. This work tries to fill part of it, using some popular tools as the microclusters, but relying on the sliding window model and not

⋆ D. Cardoso is partially supported by CAPES, CNPq and CYTED (HAROSA@IB).
⋆⋆ J. Gama is funded by the FCT project PTDC/EIA/98355/2008.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 328–335, 2012.

the most used damped window model, and also bringing an entirely new element into this area as a weightless neural network model, which was quite useful as we will show in the remainder of the paper.

2 WiSARD

The WiSARD (**Wi**lkie, **S**tonham and **A**leksander's **R**ecognition **D**evice) [4] is a Weightless Neural Network Model for data classification. In a basic WiSARD system, each of the data items classes is represented by a discriminator, a structure composed of RAM nodes and an adder, which is responsible for "learning" the class it is responsible for from example observations of it, to become able to identify observations of this class not previously seen.

WiSARD was originally used in the recognition of similar black and white images. The bits of a binary image can be combined into strings which can be used to address locations in the RAM nodes of a discriminator (Fig. 1). When a discriminator is absorbing an observation, in these addressed locations it is written '1' (consider that an unwritten locations stores '0'). When queried about how much similar an input item is to what it learned, the discriminator's answer is the sum of the values stored in the RAM locations referenced. This answer can be normalized by its division by the number of nodes the discriminator has. How input bits are combined is determined by a simple mapping, randomly defined at system startup.

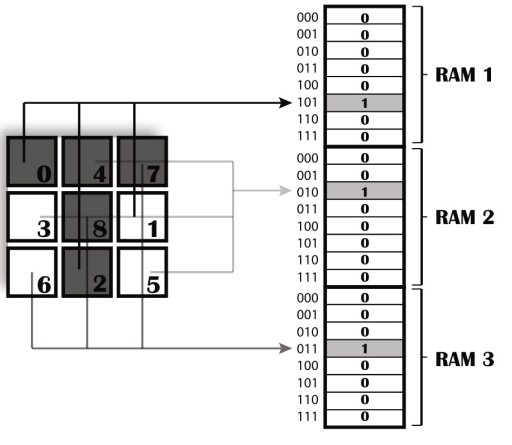


Fig. 1. Mapping a 3 × 3 image into 3 RAM addresses [11]

Altough having a simple structure, a WiSARD discriminator is a powerful classifier, able, for example, to handle not linearly separable data. And because of its simplicity, a discriminator is usually capable to operate at high speed. These charcteristics motivate its use in a stream data mining task as clustering.

3 StreamWiSARD

3.1 Data Processing

During its operation, a StreamWiSARD system mantains a list of discriminators, which have the same basic role of microclusters used in other alternatives to data streams clustering. Initially there are no discriminators.

When an observation is received, the list is traversed until a discriminator that recognizes it well enough is found, which becomes its target. Case none is found, a new discriminator is created to be the data item target. At last, the target absorbs it and takes the front of the list.

Numerically, a discriminator recognizes an observation well enough to absorb it when its answer to a query about the data item is greater than a dynamic threshold, defined as

$$t = \min\left(1, m + \frac{k}{e}\right) \quad m \in]0, 1] \text{ and } e \in \mathbb{N}^* \ , \tag{1}$$

where m and e are system parameters respectively named minimum similarity threshold and maximum absorptions count, and k is the number of absorptions the discriminator already performed. Figure 2 ilustrates the target identification.

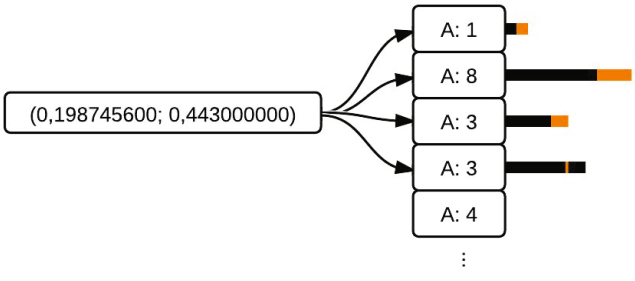


Fig. 2. Identification of the fourth discriminator as target of an example observation. The labels show the number of absorptions already performed by each discriminator. The colored part of the of the bars denote the discriminator absortion threshold, while the black part, the similarity score.

It is easy to see that the discriminators are kept ordered, from most to least recently updated. This affects how bursts of similar observations, a common pattern in data streams, are processed: in such a burst, it is expected that some observations have the same target. So, moving the target to the front of the list reduces the expected duration of the list traversal.

The transient attribute of the system results from what is kept by each RAM location and how they are maintained: they keep a timestamp of when they were last updated, and before every query to a discriminator, each of its RAM nodes have the locations last recorded outside horizon range cleaned. Consequently,

the system always works using valid knowledge. When a discriminator has no written locations, it is discarded, as it does not contain information.

The discriminators maintained during the system operation act as an intermediate clustering layer, reducing the magnitude of the amount of data to be processed to produce the final, high level clustering. We developed a proceure which defines an approximate discriminator centroid, according to the mode of the bits of the addresses of the written locations in its RAM nodes, which is used to group close discriminators when preparing the final clustering previously cited.

3.2 Input Setup

To represent data as multidimensional arrays of real values is a very common practice. However, WiSARD discriminators, our data condensation units, work with binary inputs only. A trivial real-values-to-binary-string codification procedure could be applied, but this could lead to the malfunctioning of the discriminators, as they expect similar instances to be represented as similar binary strings, which could not be guaranteed by this process. We developed a two-phase codification procedure, which turns arrays of reals into arrays of addresses to RAM locations which can be properly used by a StreamWiSARD system.

In the first phase each real value is substituted by its Binary Reflected Gray Code (BRGC) representation, according to a parameter indicating the number of decimal places which should be considered. The result is an array of binary strings. BRGC helps avoid mapping two close values to very dissimilar binary representations, far from each other w.r.t. Hamming distance, as explained in Table 1.

Table 1. An illustration of the kind of problem avoided when using BRGC instead of the traditional Base-2 representation: the highlighted strings have a Hamming distance greater than 1 to the string prior to them

	0	1	2	3	4	5	6	7	8
Base-2	0000	0001	**0010**	0011	**0100**	0101	**0110**	0111	**1000**
BRGC	0000	0001	0011	0010	0110	0111	0101	0100	1100

In the second phase, the bits of the elements of the array produced in the first phase are combined to form the addresses to be accessed in the RAM nodes of the discriminators. However, unlike WiSARD, the number of addresses a bit takes part at is exponentially proportional to its significance.

4 Experimental Evaluation

In this section we will analyze how a StreamWiSARD system performs in various situations. To set up the experiments we used MOA [5], as it provides a nice

workspace for stream data clustering research, having some evaluation measures and other approaches to this task already coded. We compare the results of two of this approaches to ours: ClusTree [8], one of the best solutions known to this problem, and CluStream [3], a historical reference of the research about this theme.

The output of each approach tested was requested periodically, at each data horizon elapsed, and its quality was measured using the following metrics: for microclusters, purity and entropy [9]; for clusters, recall [10]. The measures were taken in a given moment using input data inside horizon w.r.t. this same moment. As in the real datasets used the true number of clusters in a period was unknown, when handling them the microclusters maintained were evaluated.

Considering a set of sets of instances divided according to its classes

$$C = \left\{c_1, c_2, ..., c_{|C|}\right\} \quad , \tag{2}$$

a set of microclusters

$$M = \left\{m_1, m_2, ..., m_{|M|}\right\} \quad , \tag{3}$$

and a set of groups (clusters)

$$G = \left\{g_1, g_2, ..., g_{|G|}\right\} \quad , \tag{4}$$

we define:

1. the **purity of a microcluster** m as the size of the largest group of observations of the same class in m, normalized. The greater, the better.

$$\max_{\forall c \in C} \frac{|m \cap c|}{|m|} \tag{5}$$

2. the **entropy of a microcluster** m as the sum of the relative contributions of each class in m. The lower, the better.

$$-\sum_{\forall c \in C} \frac{|m \cap c|}{|m|} \times \log \frac{|m \cap c|}{|m|} \tag{6}$$

3. the **recall of a class** c as the probability of one of the observations of the group which represents best c be of this class. The greater, the better.

$$\max_{\forall g \in G} \frac{|g \cap c|}{|g|} \tag{7}$$

The average recall of a set as G will be called group recall and will be used to evaluate this kind of set. Differently, sets as M will be evaluated according to the average purity and entropy of its elements. The last one, weighted by the number of observations each microcluster covers.

4.1 Artificial Data

Here we observe how a StreamWiSARD system behaves when it deals with noisy and high-dimensional data: 10^5 observations, each belonging to 1 of 5 hyperspherical classes. The default number of attributes is 25, and the default noise amount is 10% of the data input. The horizon used was 10^3 observations.

Figure 3 presents the results of the experiment on data noise amount. StreamWiSARD generates the best quality clustering, but spends much more time to acomplish this than ClusTree, because the last processes observations with logarithmic complexity, while in first this operation has a linear complexity. The number of microcluster each approach has to maintain grows with the amount of noise, but this has a greater impact on StreamWiSARD's efficiency than on ClusTree's.

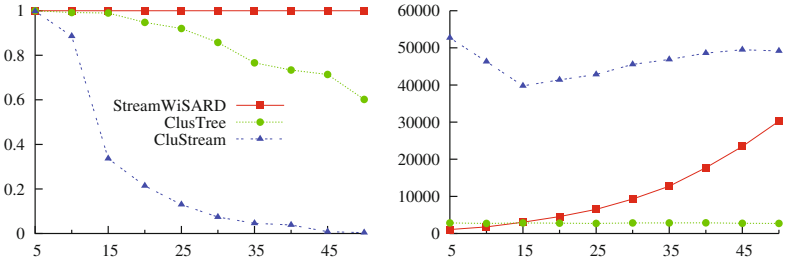


Fig. 3. Noise level (%) vs. class recall and vs. time elapsed (seconds)

Varying the number of dimensions of data causes some interesting events. Figure 4 shows the results for this experiment. StreamWiSARD produced the best quality clustering again. The low scores in the 5 and 50-dimensions runs result from the lack of parameter adjustment: there should be used a greater accuracy in the first and a greater number of nodes in the discriminators in the second. Now, StreamWiSARD is slightly faster than ClusTree, due to the lesser amount of data it deals with by the use of its own data representation.

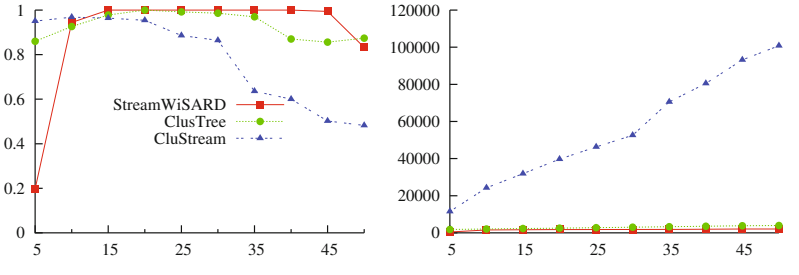


Fig. 4. Number of dimensions vs. class recall and vs. time (seconds)

4.2 Real Data

We also used the Physiological Data Modelling Contest (PDMC) [6] dataset in our tests. This is made of periodical body measures from a group of people who wore a multisensorial device for some time. As it was originally used in a classification task, every observation is class-labeled, according to the activity the person was doing when the sensors readings were captured.

In the original dataset there is a greater number of observations from people identified as 1 and 25. For each of them a new dataset containing only their respective observations was created, in which the attributes besides the 9 real-valued sensors readings and the class label were discarded. Thus, the task description is to cluster observations according to the sensor readings, considering the activities their classes.

The results, shown in Tables 2 and 3, are especially interesting because of the experiments parameters used: the horizon was of 10^2 observations, each discriminator was composed of only 5 RAM nodes, and the decimal values were considered only to its second decimal place. From a coordinate transformation point of view, the use of StreamWiSARD own representation not only reduced the dimensionality of the data but also reduced the number of bits used to represent the values in each dimension: 4 RAM nodes had addresses of 29 bits and the remainder, of 28 bits, while real values with double precision are usually stored using 64 bits.

Table 2. Results for the PDMC dataset: entropy. Best results highlighted.

		StreamWiSARD	ClusTree	CluStream
id 1	avg	**0.047**	0.108	0.092
	stddev	**0.043**	0.093	0.091
id 25	avg	**0.028**	0.078	0.093
	stddev	**0.021**	0.065	0.082

Table 3. Results for the PDMC dataset: purity. Best results highlighted.

		StreamWiSARD	ClusTree	CluStream
id 1	avg	**0.951**	0.913	0.948
	stddev	**0.038**	0.073	0.059
id 25	avg	**0.966**	0.931	0.950
	stddev	**0.024**	0.055	0.050

StreamWiSARD maintained at most 19 discriminators during the experiment. Therefore, we adjusted ClusTree maximum tree height so its number of leaves was close to this, what brought up the benefits of using a more powerful data summarizing unit.

5 Conclusion

In this paper we presented work, which checked the applicability of WiSARD discriminators as basic data condensation units in a two-phase stream data clustering system. We showed that the use of this powerful classifier resulted in a system with quite interesting characteristics, as being able to define high quality clusters although requiring to maintain a quite small number of microclusters, or having a nice parameters set, which permit, for example, the discard of irrelevant input parts as early as possible, what reduces the entire system workload. StreamWiSARD was developed based on the sliding window model for stream data applications, what is also another nice feature as this is not the most popular time data model among those available, but is the preferred one in a variety of situations.

References

1. Zhou, A., Cao, F., Qian, W., Jin, C.: Tracking clusters in evolving data streams over sliding windows. Knowl. Inf. Syst. 15(2), 181–214 (2008)
2. Wan, L., Ng, W.K., Dang, X.H., Yu, P.S., Zhang, K.: Density-based clustering of data streams at multiple resolutions. ACM Trans. Knowl. Discov. Data 3, 14:1–14:28 (2009)
3. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: VLDB, pp. 81–92 (2003)
4. Aleksander, I., Gregorio, M.D., França, F.M.G., Lima, P.M.V., Morton, H.: A brief introduction to weightless neural systems. In: Proceedings of the 17th European Symposium on Artificial Neural Networks, ESANN 2009, Bruges, Belgium, April 22-24 (2009)
5. Bifet, A., Holmes, G., Pfahringer, B., Read, J., Kranen, P., Kremer, H., Jansen, T., Seidl, T.: Moa: A real-time analytics open source framework. In: [7], pp. 617–620
6. Brodley, C.E. (ed.): Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004), Banff, Alberta, Canada, July 4-8. ACM International Conference Proceeding Series, vol. 69. ACM (2004)
7. Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.): ECML PKDD 2011. LNCS, vol. 6913. Springer, Heidelberg (2011)
8. Kranen, P., Assent, I., Baldauf, C., Seidl, T.: The clustree: indexing micro-clusters for anytime stream mining. Knowl. Inf. Syst. 29(2), 249–272 (2011)
9. Lühr, S., Lazarescu, M.: Incremental clustering of dynamic data streams using connectivity based representative points. Data Knowl. Eng. 68(1), 1–27 (2009)
10. Moise, G., Sander, J., Ester, M.: Robust projected clustering. Knowl. Inf. Syst. 14, 273–298 (2008)
11. Bandeira, L.C.: NC-WISARD: Uma interpretação sem pesos do modelo neural neocognitron. M.Sc. thesis, Rio de Janeiro, RJ, Brasil (2010) (in Portuguese)

Cloud Computing Environments for Biomedical Data Services

Marek Penhaker[1], Ondrej Krejcar[2], Vladimir Kasik[1], and Václav Snášel[1]

[1] VSB - Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science, 17. Listopadu 15, Ostrava Poruba, 700 30, Czech Republic
{Marek.Penhaker,Vladimir.Kasik,Vaclav.Snasel}@vsb.cz
[2] University of Hradec Kralove, Faculty of Informatics and Management, Department of Information Technologies, Rokitanskeho 62, Hradec Kralove, 500 03, Czech Republic
Ondrej.Krejcar@ASJournal.eu

Abstract. The goal of the project we describe in this paper was the cloud computer environment establishment for biomedical data processing services using a security access. The relevant cloud application for user interactive interaction with biomedical data were realized on .NET Framework technology platform by the help of rapid calculating by FPGA technology. Interconnection of these technologies allows a large spectrum of users to quick access to pre-evaluated biomedical data. This article discusses the possible topological solutions and implementation of that system.

Keywords: Cloud Computing, FPGA, Biomedical Data, Web Service, Database.

1 Introduction

Biomedical research of last decade includes implementation of a wide spectrum of experimental techniques which produce large amount of biomedical data which need to be processed (using up to date algorithms) in Real Time if possible [1]. This requirement is not possible to observe using only standard techniques for implementation. Moreover, there is no any common standard for storing of biomedical data. Almost all mobile devices manufacturers create and use their own data formats, which are mutually incompatible and in most cases not open.

Research with biomedical data in most cases start with discovering of real data produced by some biomedical device.

The current trend is also to manage data through a variety of Web interfaces that provides many advantages for the user. In that case, data storage using a standard XML format which is often used and supported by many tools and programming languages [2]. Disadvantage of this open format is his power consumption when a parsing of data (extraction of data) is needed.

Cloud computing can be in some simplified meaning recognized as internet for its basic principle – the network is draw as a cloud. Kevin Marks (Google) sad: "The

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 336–343, 2012.

term “cloud” is a metaphor for the Internet” [1]. In most cases the users of biomedical systems need to be connected to system remotely using various kinds of mobile devices or laptops. For these cases some kind of suitable web interface is needed where processed data from system cloud are presented.

Cloud framework can be recognized as three types of Services: (1) Infrastructure as a Service (IaaS); (2) Platform as a Service (PaaS); and (3) Software as a Service (SaaS) [1]. From this point of view, cloud is not only a service of computing power, but i tis recognized mostly as a service which need to be structured by some king of architecture using computing environment [1].

2 Problem Definition

The current presentation of biomedical - electrophysiological data is mostly accessing on local terminals with visualization of measured data using web applications. The processing of these data is done either by post-processing after measurement [16], [18] and in minor cases on-line using Real Time processing which puts high demands on hardware. Such demanding process is therefore destined only for some kind of specialized work where the cost of computing environment is reasonable [17].

The issue of distance evaluation of physiological data using web terminals is solved several ways [3]. To address such complex biological signal processing several tasks like computational parallelism, high computational speed and very large scale integration with FPGA (Field Programmable Gate Array) can be advantageously used [4], [5]. Using of FPGA technology with these features move the processing of stored data from ofline postporocessing to online Real Time processing with a possibility to Real Time visualisation of processed and evaluated data sa well as their sharing to remote clients using Web interface [6].

3 New Trends in Biomedical Cloud Computing

As the current presentation of any kind of biomedical data is mostly provided by a visualization of measured data using web applications [2], [7], when an up to date technology is used for data processing as well as cloud solution, an exciting modern and powerful solution can be accessed to allow users (medicians, physicians) to make a decisions based on “filtered” biomedical data faster than any time before [8].

The power of computer processors as well as storage capacity doubles every 18 months. Innovative cycle of computer technologies is growing even faster especially for scientific are where a special instruments are now developed by the help of new hardware and software features. During the last decade a three important changes have occurred regarding data. We can recognize an explosion in the amount of data produced by not only biomedical solutions. These data need to be managed and processed what bring new fundamental changes like a new algorithms for how these data are analysed using data mining. Mix of these changes is recognized as a beginning of change a biomedicine into a data intensive science [8].

This kind of rapid evolution in biomedical data presentation however brings some new business paradigm for sharing of biomedical information [9]. Clouds

environment generally offer resources on demand, where the most used case is pay per use charges. This principle is based on large farms of inexpensive, dedicated servers, sometimes supporting parallel computing. For research area, the price for service (computing of some biomedical algorithm with data) can be significantly lowered using Cloud environment where the sharing of time slices moves the utilization of such solution up to 100%.

Problem, which appears across all biomedical cloud solutions, can be recognized as a searching for relationships between all the data. We are now not able to compute over all of these data to make discoveries, and, more importantly, there is no conceptual framework to integrate all this data [9].

One step in the long way can be covered by a special design of biomedical system with strong impact on each element of created network [20].

4 Biomedical System Design

For the purposes of medical practice, it is important that biomedical records are accessible from various web sites and networks without the need of specialized software. For this purpose, a system of storage and management of biomedical data has been already designed [3].

The system described in this article uses a web application that provides a view of stored biomedical data and also allows you to store records in one of three predefined formats – EDF, DAT and XML. As part of this web browser application is a viewer of stored records, which replaces the traditional programs and enables a view of recorded data directly on the website without need of additional programs. The system for storage and exchange biomedical data is based on ASP.NET 2.0 platform .NET Framework and Microsoft SQL Server.

For storage of biomedical data is used a data storage server [20]. Every file uploaded to the server has its own unique name that is used as ID in the database. The basic functions of Web applications provide: display a list of all records stored in a database, add a new record and manage users. The Web application has its own biomedical data browser that allows graphical display of such an EEG records from XML file.

Web site applications are for viewing records and allow for a direct interaction between the system and user. The layout was created in Photoshop CS4 Trial with CSS support in Visual Studio Web Developer. The web interface contains a home page, login page, catalog of saved records, and the administration page.

A significant part of the application is the browser of recorded biomedical data, which enables to display the 20-channels EEG records. It is also possible to modify the scale of the axis and to move the view in a direction of x axis. To display the data we used ZedGraph component, which is distributed under LGPL license.

Recently the system has been extended to evaluate the biomedical data with the detection of some time and quality parameters. These functions are implemented in specialized hardware with an FPGA chip. This hardware is exchanging data with a data server, but can also be connected to the analog inputs for on-line processing of biomedical signals.

5 Implementation of New Methods

The processing of biomedical data often requires the detection of various events in recorded measurements chain [10], [11], [12]. The requirements for computing power increases mainly in the evaluation of biomedical signals in real time. Using some kind of transformation like the wavelet is not difficult due to the number of steps in one subscription - 2n, but mainly for the use on large real biomedical data which requires fully parallel computing with high performance. Implementation of this option can be currently done with only two possible architectural designs: (1) a supercomputer or (2) a parallel computing with FPGA structures. Taking into account all possibilities and modern accesses, it seems to be a third alternatives instead of first two for fast signal processing [Fig 1]. That figure represents a system, which transmits stored biomedical data to the coprocessor unit for processing on user request.

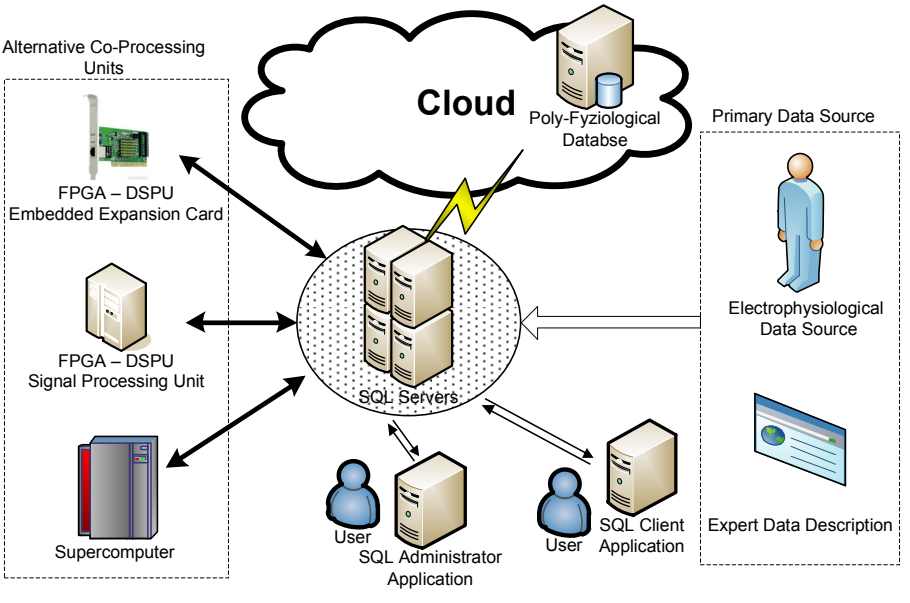


Fig. 1. Overall System Structure. The data is returned after processing at data server, from which it can be visualized in a web application.

Figure [1] also shows the three possible alternatives of co-processing units: 1) FPGA – DSPU (Digital Signal Processing Unit) as Embedded Expansion Card, 2) FPGA – DSPU as stand-alone unit with an Ethernet connection and 3) Supercomputer as an alternative. The first two of these options are based on FPGA programmable logic, which uses both hardware and software approach for data processing. Hardware solution is based on the parallel implementation of some sub-algorithms in cooperation with the DSP units. Software solution allows you to process complex calculations and algorithms supporting MicroBlaze soft processor.

Conceptually, the co-processing unit with an FPGA is designed in two ways. The first solution is a PCI card for installation into a data server. This solution is elegant

and allows a maximum data throughput between the server and the data processing unit. Another solution of co-processing unit is a standalone module which can be connected via a communication network. In this case, a standard Ethernet network with TCP / IP protocol is used. The advantage of the solution is not only independence of data server location, but also a possibility of usage this unit with more servers and / or clients on the network. A significant feature of standalone unit is an expansion possibility of the analog inputs for on-line biomedical signals processing. For this function, the unit is equipped with A/D converter with an appropriate resolution. With the implementation of a Web interface it extends the use of process units for additional functions when the web client can connect directly to the co-processing unit [Fig. 2].

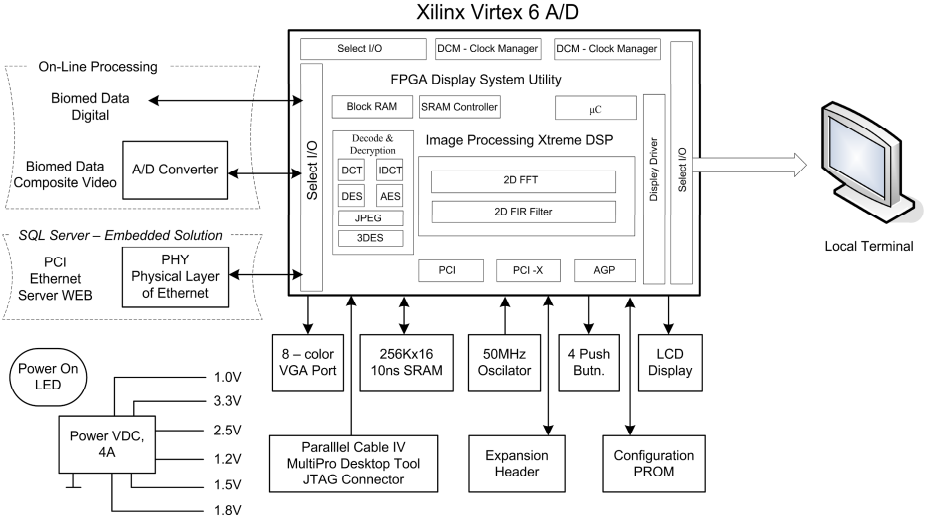


Fig. 2. FPGA – DSPU Detailed Structure [17]

FPGA - DSPU Detailed Structure is based on Xilinx's Virtex 6 FPGA programmable logic. From a point of view of dataflow the FPGA – DSPU co-processing unit constitutes second link between web server and SQL server. The operation of FPGA is controlled by a web application that sends commands to the unit. Individual digital biosignal processing algorithms operate on data from Microsoft SQL Server. The resulting data are processed by web applications and offered to the authorized user's web browser.

One of the implemented features of the system is an EEG evaluation of patients and their association with epileptic attacks. Here the wavelet transform methods are used to evaluate the EEG data. Phase of epileptic attacks are evaluated by detection algorithm that was trained to locate some reference points in the image. Placement of these points and their movement is a basis for subsequent classification of the patient. Left image of Figure [Fig. 3] is the input image and right image is updated with discovered reference points [19]. In another analysis, the arm angles, hand position and movement, posture and movement of the head are observed. The result is an estimation of the overall condition of the patient, focusing on pathological states of the brain.

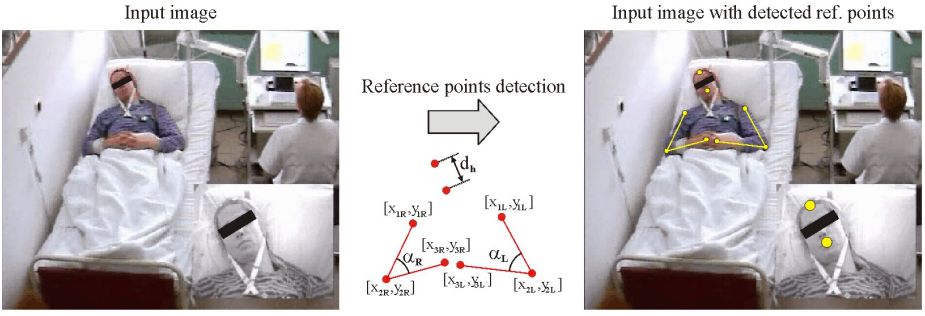


Fig. 3. Evaluation of Medical Image. Real-time processing of input image gives a result in terms of reference points, whose position is further evaluated [3], [17].

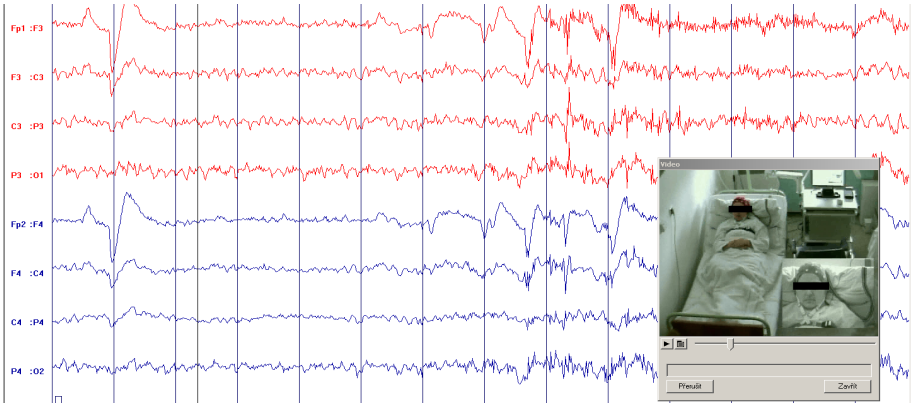


Fig. 4. An epileptic attack detected on the EEG record [3], [17]

To detect epileptic attack from EEG record were used wavelet transform methods implemented on an FPGA platform. Among the fundamental parameters of signal derived from EEG data are mainly amplitude peaks and frequency spectrum [13], [14], [15]. Furthermore, there was observed a delay between the first signs of attack in EEG and the corresponding motion signals [Fig. 4].

6 Conclusions

Development of bio-signals currently comprises the majority of issues, which covers the field of biomedical engineering. Processing of these data in real time is very difficult to implement the task with high computational and financial demands of the workplace. The presented solution treatment and detection of signals in real time using FPGA brings new possibilities not only well equipped for a particular workplace, but also for users of web applications for visualization and processing the distance signals in real time. Using the appropriate topology for connecting the SQL and database servers, and digital signal processing unit can achieve comparable results as a supercomputer intended to be so complicated mathematical calculations.

The proposed system was tested on a specific polysomnographic recording video data and successfully detected using this system, an epileptic seizure and subsequent levels of nerve-muscle movements and to determine the strength of muscle contraction with respect to the degree of fit.

Acknowledgment. The work and the contribution were partially supported by the project (1) “SMEW – Smart Environments at Workplaces”, the Grant Agency of the Czech Republic, GACR P403/10/1310; (2) "Smart Solutions in Ambient Intelligent Environments", University of Hradec Kralove under the project SP/2012/6; (3) SV SP 2012/114 “Biomedical engineering systems VIII”; (4) TACR TA01010632 “SCADA system for control and measurement of process in real time. The paper has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

References

1. Kupfer, D.M.: Cloud Computing in Biomedical Research. Aviation, Space, and Environmental Medicine 83(2), 152–153 (2012), doi:10.3357/ASEM.3242.2012
2. Fusaro, V.A., Patil, P., Gafni, E., Wall, D.P., Tonellato, P.J.: Biomedical Cloud Computing With Amazon Web Services. Plos Computational Biology 7(8) (2011), doi:10.1371/journal.pcbi.1002147
3. Krawiec, J., Penhaker, M., Krejcar, O., Novak, V., Bridzik, R.: System for Storage and Exchange of Electrophysiological Data. In: Proceedings of 5th International Conference on Systems, ICONS 2010, April 11-16, pp. 88–91. IEEE Conference Publishing Services, Menuires (2010), doi:10.1109/ICONS.2010.23
4. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
5. Simsik, D., Galajdova, A., Majernik, J., Hrabinska, I., Zelinsky, P.: The video analysis utilization in rehabilitation for moblity development. Lékař a technika. Česká republika, 4-5, Ročník 35, 87–92 (2004) ISSN 0301-5491
6. Dziakova, M., Zelinsky, L., Simsik, D., Majernik, J., Galajdova, A.: Objektivizácia účinku senzomotorickej stimulácie pomocou SMART-u. Rehabilitácia 44(4), 210–214 (2007) ISSN 0375-0922, pp. 210-214
7. Ren, J.Y., Williams, N., Clementi, L., Krishnan, S., Li, W.W.: Opal web services for biomedical applications. Nucleic Acids Research 38(suppl. 2), W724–W731 (2010), doi:10.1093/nar/gkq503
8. Grossman, R.L., White, K.P.: A vision for a biomedical cloud. Journal of Internal Medicine 271(2), 122–130 (2012), doi:10.1111/j.1365-2796.2011.02491.x
9. Rosenthal, A., Mork, P., Li, M.H., Stanford, J., Koester, D., Reynolds, P.: Cloud computing: A new business paradigm for biomedical information sharing. Journal od Biomedical Informatics 43(2), 342–353 (2010), doi:10.1016/j.jbi.2009.08.014
10. Zivcak, J., Petrik, M., Hudak, R., Toth, T., Knezo, D., Kovalova, E. Embedded tensile strenght test machine FM1000 - An upgrade of measurement and control, Diffusion and Defect Data Pt.B: Solid State Phenomena 147-149, pp. 657–662, ISSN: 10120394, ISBN: 3908451655; 978-390845165-

11. Kristof, M., Hudak, R., Takacova, A., Zivcak, J., Fialka, L., Takac, R.: Contact pressure measurement in trunk orthoses. In: ICCC-CONTI 2010 - IEEE International Joint Conferences on Computational Cybernetics and Technical Informatics, Proceedings, art. no. 5491304, p. 175 (2010) ISBN: 978-142447433-2, doi:10.1109/ICCCYB.2010.5491304
12. Sabol, F., Vasilenko, T., Novotny, M., Tomori, Z., Bobrov, N., Zivcak, J., Hudak, R., Gal, P.: Tradermal running suture versus 3M™ Vetbond™ tissue adhesive for wound closure in rodents: A biomechanical and histological study. European Surgical Research 45(3-4), 321–326, ISSN: 0014312X, doi:10.1159/000320837
13. Aidu, E.A.I., Trunov, V.G., Titomir, L.I., Tysler, M., Turzova, M., Szathmary, V.: Electrocardiographic ST segment changes as an indicator for localization of injury potentials. A Computer Simulation Study kardiologia 15(1), 21–24, ISSN: 12100048
14. Cerny, M.: Movement Activity Monitoringof Elederly People – Application in Remote Home Care Systems. In: Proceedings of 2010 Second International Conference on Computer Engineering and Applications ICCEA 2010, Bali Island, Indonesia, March 19-21, vol. 2, IEEE Conference Publishing Services, NJ (2010) ISBN 978-0-7695-3982-9
15. Cerny, M.: Movement Monitoring in the HomeCare System. In: Dössel, O., Schlegel, W.C. (eds.) WC 2009. IFMBE Proceedings, vol. 25, pp. 356–357. Springer, Berlin (2009)
16. Bernabucci, I., Conforto, S., Schmid, M., D'Alessio, T.: A bio-inspired controller of an upper arm model in a perturbed environment. In: Proceedings The 2007 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp. 549–553 (2007) ISBN: 978-1-4244-1501-4
17. Kasik, V., Penhaker, M., Novák, V., Bridzik, R., Krawiec, J.: User Interactive Biomedical Data Web Services Application. In: Yonazi, J.J., Sedoyeka, E., Ariwa, E., El-Qawasmeh, E. (eds.) ICeND 2011. CCIS, vol. 171, pp. 223–237. Springer, Heidelberg (2011)
18. Korpas, D., Halek, J.: Pulse wave variability within two short-term measurements. Biomedical Papers of the Medical Faculty of the University Palacky, Olomouc, Czechoslovakia 150(2), 339–344 (2006) ISSN: 12138118
19. Brad, R.: Satellite Image Enhancement by Controlled Statistical Differentiation. In: Innovations and Advances Techniques in Systems, Computing Sciences and Software Engineering, International Conference on Systems, Computing Science and Software Engineering, ELECTR NETWORK, December 03-12, pp. 32–36 (2007)
20. Halic, T., Kockara, S., Bayrak, C., Rowe, R., Mixed reality simulation of rasping procedure in artificial cervical disc replacement (ACDR) surgery. BMC Bioinformatics 11(suppl. 6), Article Number: S11 (2010), doi: 10.1186/1471-2105-11-S6-S11

Comparing Particle Swarm Optimization Approaches for Training Multi-Layer Perceptron Neural Networks for Forecasting

Saulo M. Santos, Mêuser J.S. Valença, and Carmelo J.A. Bastos-Filho

Polytechnic School of Pernambuco, University of Pernambuco,
Recife, Pernambuco, Brazil
{smocs,meuser,carmelofilho}@ecomp.poli.br, mestrado@ecomp.poli.br

Abstract. Multilayer Perceptron Artificial Neural Networks (MLP-NN) have been widely used to tackle forecasting problems. The most used algorithm for training MLP-NN is called Backpropagation (BP). Since the BP presents a high chance to be trapped in local minima during the training process for forecasting, we propose in this paper to assess some recently proposed variations of the Particle Swarm Optimization algorithm (PSO) applied for this purpose. We tested the standard PSO, the APSO, the ClanPSO and the ClanAPSO in five benchmark data sets. Although the standard version of the PSO presented worse results when compared to the BP algorithm, we observed that the ClanAPSO outperformed the BP algorithm in most of cases.

Keywords: Artificial Neural Networks, Multi-Layer Perceptron, Particle Swarm Optimization, Forecasting.

1 Introduction

Forecasting algorithms have been applied in many areas of knowledge, as for example medicine, stock market, power generation, traffic control. Although classical statistics is more often used to develop prediction models, Artificial Neural Networks (NN) have been widely used to tackle this type of problem [1], [2], [3]. NN are parallel distributed systems composed by simple processing units (nodes), which are arranged in one or more layers. The nodes are interconnected by a large number of unidirectional connections, which are generally associated to weights. These weights are used to store knowledge [4].

Multi-layer Perceptron Neural Networks (MLP-NN) are feed-forward networks with multiple layers. MLP-NN present the capability to solve non linear problems and the most used algorithm to train MLP-NN is the Backpropagation (BP) algorithm [5]. BP is based on the descendent gradient method. Although BP has been widely used, it presents some limitations such as slow convergence, incapability to tackle multi-objective problems and high probability to be trapped in local minima during the training process [6]. This is even worse for high dimensionality, which is case for forecasting problems.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 344–351, 2012.

In order to overcome these limitations, some swarm intelligence algorithms have been applied for training NN. Among them, we can cite: Particle Swarm Optimization (PSO) [6] and Artificial Bee Colony optimization [7]. Although these techniques present interesting results, they also present some weaknesses.

In this paper we compare the performance of recently proposed PSO variations for training MLP-NN for forecasting. The remainder of the paper is organized as follow: Section 2 present the PSO basic concepts and the recently proposed PSO variations; Section 3 and 4 present the experimental setup and the results, respectively. In Section 5 we give our conclusions.

2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) was proposed by Kennedy and Eberhart in 1995 [8]. PSO was inspired by the behaviour of flocks of birds and is commonly used to solve optimization problems in continuous and hyper-dimensional search spaces. In the PSO, the swarm is composed by a group of particles. Each particle fly through the search space with a velocity that is updated by using its own information and the information received from the neighbour particles.

The attributes of a particle are: the position $\boldsymbol{x}(t)$, that represents a possible solution for the problem; the fitness value for the current position within the search space $f(\boldsymbol{x}(t))$; a cognitive memory that stores the best position found by the particle itself along the search process, and the fitness value for this position $(\boldsymbol{p}_{best}(t); f(\boldsymbol{p}_{best}(t)))$; a social leader which is the best particle in the neighbourhood, and the fitness value for this social leader $(\boldsymbol{n}_{best}; f(\boldsymbol{n}_{best}))$; and the velocity of the particle within the search space $\boldsymbol{v}(t)$ [9].

At each iteration, each particle is updated by using equations (1) and (2). Equation (1) has three terms, the inertial, the cognitive and the social terms.

$$\boldsymbol{v}_i(t+1) = \omega \boldsymbol{v}_i(t) + c_1 r_1 \left[\boldsymbol{x}_{pbest} - \boldsymbol{x}(t)\right] + c_2 r_2 \left[\boldsymbol{x}_{nbest} - \boldsymbol{x}_i(t)\right], \quad (1)$$

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1). \quad (2)$$

Since the major problems for the standard version of the PSO are to maintain the diversity of the population along the search process and avoid premature convergence to a local optimum, many approaches have been proposed recently. In the next subsection we present some of them.

2.1 Adaptive Particle Swarm Optimization

The Adaptive PSO (APSO) was proposed by Zhang *et al.* [10]. In APSO, the parameters are updated by a systematic scheme based on fuzzy rules. An elitist learning strategy was also employed.

The following steps are performed at each iteration of the algorithm:

- Evaluation of the evolutionary factor based on the population distribution;
- Classification of the evolutionary state of the swarm;
- Adaptation of the acceleration coefficients;
- Implementation of the elitist strategy;
- Adaptation of the inertia factor.

Evolutionary Factor. we need to calculate the average distance from each particle to the others using equation (3).

$$d_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} \sqrt{\sum_{k=1}^{D} (x_i^k - x_j^k)^2}, \tag{3}$$

where N is the total number of particles in the swarm and D is the number of dimensions of the problem. Then, the evolutionary factor (f_{evol}) is calculated by equation (4).

$$f_{evol} = \frac{d_g - d_{min}}{d_{max} - d_{min}} \in [0, 1] \tag{4}$$

where d_g is the average distance between the best particle of the swarm and the other particles. d_{min} and d_{max} are the smallest and the greater average distance considering all the particles of the swarm.

Classification of the Evolutionary State. the swarm is classified by fuzzy rules in one of the following states: *exploration*, *exploitation*, *convergence* and *escape*. In the *convergence* state, f_{evol} is close to the minimum. In this case, the distance between the best particle of the swarm to the others is minimal. *exploitation* state occurs when f_{evol} is small. In this case, the swarm is exploiting in a limited area. In the *exploration* state, f_{evol} presents medium of high values. In this case, the swarm is exploring a region of the search space. In the *escape* state, f_{evol} is close to "1". This means that the at least one particle of the swarm escaped from a local minima to other region.

Determining the Acceleration Coefficients. The acceleration coefficients (c_1 and c_2) are updated along the iterations in order to allow an adaptive behaviour for the particles according to f_{evol}. c_1 and c_2 are initialized with values equal to 2.0 [10]. If the swarm is in the *convergence* state, then we slightly increment c_1 and slightly increment c_2. If the swarm is in the *exploitation* state, then we slightly increment c_1 and slightly decrement c_2. If the swarm is in the *exploration* state, then we increment c_1 and decrement c_2. If the swarm is in the *escape* state, then we decrement c_1 and increment c_2. $c_1 + c_2$ must be equal to 4.0. The increment or decrement of the acceleration coefficients is obtained through the acceleration rate (σ), which is calculated by an uniform distribution between 0.01 and 0.05.

2.2 PSO Topologies

The communication scheme to exchange information between the particles is called topology. The topology defines the neighbourhood of the particles, and as a consequence, regulates the flow of information within the swarm.

In a strongly connected swarm, all particles tend to converge faster to the same region of the search space. Because of this, there is a high probability for the entire swarm to be trapped in a single local minima. On the other hand, this probability is lower for less connected topologies, but the convergence is slower.

Several topologies have been developed in order to define more efficient flow of information within the swarm for the PSO algorithm, as for example: Star [11], Von Neumann [11], Multi-Ring [12] and ClanPSO [13]. The ClanPSO topology, proposed in 2008, can achieve better results in high-dimensional search spaces [13].

ClanPSO. In the ClanPSO algorithm, we have different sub-swarms called clans. Each clan is composed by particles that are strongly connected for sharing information. At each iteration, all particles of each clan perform a PSO based search process. In the end of each iteration, the particles which had obtained the best result within each clan during the search process so far are chosen as leaders. Then, we run a PSO solely with the leaders. This process is called *Conference of Leaders*. At the end of the conference, each leader returns to its clan and spread the information acquired during the conference to the other particles. As a consequence, all particles will be indirectly influenced by the best position found by the best particle of the whole swarm.

ClanAPSO Algorithm. was reported by Pontes *et al.* in 2011 [9]. The idea is to use the ClanPSO topology and execute the APSO algorithm during the update process of each clan.

3 Experimental Setup

In this section, we assess the performance of four PSO-based algorithms for training a MLP-NN for forecasting: PSO, APSO, ClanPSO and ClanAPSO. Each particle is composed by the weights of the NN connections. In other words, the position vector of each particle contains all synaptic weights of NN, thereby, the dimension of the position vector of each particle is equal to the number of NN connections.

We use the Root Mean Square Error (RMSE) to evaluate the performance of the training algorithms. We used five well known databases that are used for forecasting: Abalone [14] that represents the age of abalone from physical measurements; California housing [15] that represents the variables used during the 1990s Census in California; Delta Elevator that represents the controlling task of the elevators of a F16 aircraft; Machine [14] that represents the relative CPU performance and Servo [14] that represents a simulation of a servo system involving a servo amplifier, a motor, a lead screw/nut, and a sliding carriage of some sort.

All databases were preprocessed by using a normalization procedure. The data was divided into three sets: Training, Validation and Testing. The training set,

the test set and the validation set represent 50%, 25% and 25% of the whole database information. We performed 30 trials for each training algorithm.

In all cases, the MLP-NN has a single hidden layer containing 20 neurons. The number of inputs and outputs depends on the used database. In the end of training cycle, the MLP-NN is assessed by using the validation set. We used the cross-validation criterion as the stopping criterion for the training. If the mean square error (MSE) remains constant consecutively or increases consecutively over 30 iterations, then the training is stopped.

4 Results

Table 1 shows the Average RMSE and (standard deviation) for the BP and the PSO-based algorithms. The best results are marked in bold for each benchmark database. The BP algorithm achieved the best results for the *Cal_Housing* and *D_Elevator* databases. However, in both cases the results obtained by the ClanAPSO are too close. The APSO algorithm achieved the best result for the *Servo* and *Abalone* databases. Again, the ClanAPSO achieved similar results. One can observe that the ClanAPSO achieved the best or at least a similar performance when compared to the other algorithms, including the BP algorithm.

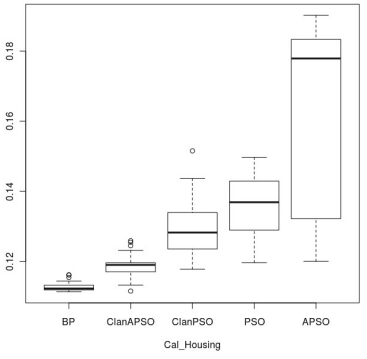
Table 1. Average and (standard deviation) of the RMSE over 30 trials

Bases	*Cal_Housing*	*Abalone*	*Servo*	*Machine*	*D_Elevator*
BP	**0.113(4.8E-5)**	0.066(9.1E-6)	0.169(6.5E-4)	0.072(1.46E-3)	**0.044(4E-7)**
PSO	0.136(2.2E-3)	0.075(2.4E-4)	0.123(9.5E-3)	0.072(5.9E-3)	0.051(3.7E-4)
APSO	0.164 (0,018)	**0.064(1.9E-4)**	**0.086(0.02)**	0.105(0.04)	0.046 (3.3E-5)
ClanPSO	0.13(1.8E-3)	0.071(1.5E-4)	0.148(0.012)	0.065(8E-3)	0.049(3.8E-4)
ClanAPSO	**0.119(3.2E-4)**	**0.064(6.1E-4)**	**0.096(0.012)**	**0.050(0.023)**	**0.045(4.9E-4)**

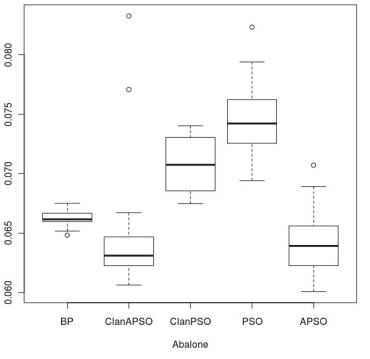
Since some results are quite similar, we depict the boxplot chart of the RMSE for all the training algorithms in all databases in Figure 1.

Yet there are some cases in which is not possible to evaluate if the difference between the performance of the algorithms are statistically significant. Because of this, we performed the statistical Wilcoxon test by using the *R software* [16]. Table 2 presents the results obtained from the Wilcoxon statistical test comparing the PSO-based algorithm to the BP algorithm for all databases. Symbol (−) indicates similar results; Symbols (△) or (▼) indicate that the swarm algorithm presented better or worse results than the BP, respectively.

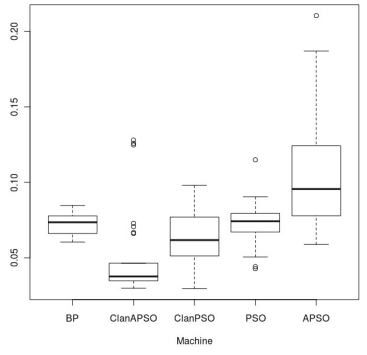
One can observe that the ClanAPSO outperformed the BP algorithm in three databases and achieved a similar performance in the other two databases. However, the standard version of the PSO was outperformed by the BP algorithm in most of cases. The other PSO-based approaches achieved a middle-term performance.



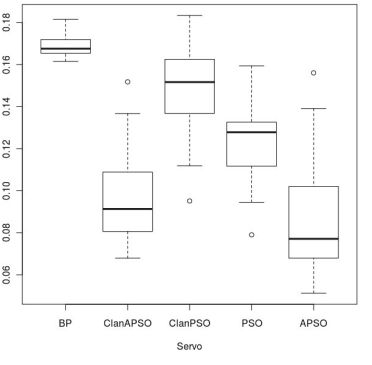
(a) California Housing



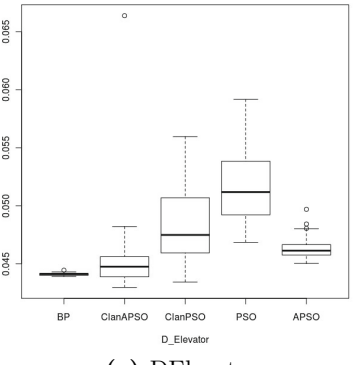
(b) Abalone



(c) Machine



(d) Servo



(e) DElevator

Fig. 1. Boxplot chart of the fitness for all training algorithms for the databases: (a) California Housing, (b) Abalone, (c) Machine, (d) Servo and (e) Delta Elevator

Table 2. Wilcoxon statistical test comparing the PSO-based algorithms to the BP for 5 data sets. Symbol (−) indicates similar results; Symbols (△) or (▼) indicate that the swarm algorithm presented better or worse results than the BP, respectively.

Base	*ClanAPSO*	*ClanPSO*	*APSO*	*PSO*
Cal_Housing	−	▼	▼	▼
Abalone	△	▼	△	▼
Machine	△	△	▼	▼
Servo	△	△	△	△
D_Elevator	−	▼	▼	▼

5 Conclusion

In this paper we have analysed the performance of some recently proposed Particle Swarm based algorithms to train the Multi-layer Perceptron Artificial Neural Networks for forecasting. We assessed the performance of four different variations of the PSO in five different benchmark databases and we observed that the most simple version can not outperform the Backpropagation algorithm. However, more sophisticated approaches can obtain better results and the adaptive PSO with cooperative sub-swarms can outperform the well known Backpropagation algorithm.

References

1. Junyou, B.: Stock price forecasting using pso-trained neural networks. In: Evolutionary Computation, CEC 2007, pp. 2879–2885 (2007)
2. Sun, W., Zang, Y., Li, F.: The neural network model based on pso for short-term load forecasting. In: 2006 International Conference on Machine Learning and Cybernetics, pp. 3069–3072 (2006)
3. Valença, M., Ludermir, T.: Neural networks vs. parma modelling: Case studies of river flow prediction. In: Proceedings Sixth Brazilian Symposium on Neural Networks, pp. 113–116 (November 2000)
4. Braga, A., Carvalho, A., Ludermir, T.: Redes Neurais Artificiais: Teoria e Aplicações. In: LTC (2000)
5. Valença, M.: Fundamentos das Redes Neurais. Livro Rápido (2009)
6. Carvalho, M.: Uma análise de otimização de redes neurais mlp por enxame de partículas. Master's thesis, Universidade Federal de Pernambuco (2007)
7. Pham, D.T., Koç, E., Ghanbarzadeh, A., Otri, S.: Optimisation of the weights of multi-layered perceptrons using bees algorithm. In: Proceedings of 5th International Symposium on Intelligent Manufacturing Systems, pp. 38–46 (2006)
8. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceeding of IEEE International Conference on Neural Networks, vol. 4 (1995)
9. Pontes, M.R., Lima-Neto, F.B., Bastos-Filho, C.J.A.: Adaptive clan swarm optimization. In: IEEE Symposium Series on Computational Intelligence, SSCI, pp. 17–22 (2011)
10. Zhan, Z., Zhang, J., Li, Y., Chung, H.: Adaptive particle swarm optimization. IEEE Transactions on Systems Man, and Cybernetics-Part B: Cybernetics 39(6), 1362–1381 (2009)

11. Kennedy, J., Mendes, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 38–44
12. Bastos-Filho, C.J.A., Carvalho, D.F., Caraciolo, M.P., Miranda, P.B.C., Figueiredo, E.M.N.: Multi-ring particle swarm optimization. In: 10th Brazilian Symposium on Neural Networks, p. 572 (2008)
13. Carvalho, D.F., Bastos-Filho, C.J.A.: Clan particle swarm optimization. In: IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 3044–3051 (June 2008)
14. Aha, D., Asuncion, A., Newman, D.: The uci machine learning repository (2007)
15. FCT: The laboratory of artificial intelligence and decision support (liaad) (1988)
16. Chambers, J.: The r project for statistical computing

Hybrid Architecture to Predict Trends at Stock Exchange of São Paulo: Markowitz Model and a Multilayer Perceptron

Paulo Henrique Kaupa[1], Renato José Sassi[1], and Edinalva Batista Ramalho[2]

[1] Nove de Julho University - Industrial Engineering Post Graduation Program, São Paulo, Brazil
paulo.kaupa@gmail.com, renato.sassi@ieee.org
[2] Fundação Getúlio Vargas CEAFE Master in Financial Economics, São Paulo, Brazil
edinalva.ramalho02@gmail.com

Abstract. The main challenge in Stock Exchange is choose stocks with uptrend in order to compose a profitable stock portfolio and also ensure security of investment, since the risk in stock investment is considered high. One tool that can help investors to identify the stocks behavior and help to select the right stocks becomes essential. The application of intelligent techniques, especially Artificial Neural Networks to forecast trends in stock prices generated good results. Thus, in this paper was created hybrid architecture, composed by the Markowitz Model and an Artificial Neural Network Multilayer Perceptron, in order to support the investor. For the experiments, the information of the ten most traded stocks on Stock Exchange of São Paulo was extracted. The hybrid architecture is given as follows: Processing Stock Information in Markowitz Model then result is presented as one of input variables of neural network. For analyze the results, was applied an investment simulator, where the investment return obtained point to the use of hybrid architecture in investments.

Keywords: Hybrid architecture, Multilayer Perceptron, Markowitz.

1 Introduction

High risk investment as stock exchange, gains new interested (investors) among other options of investments, because this type can bring higher profits to the investor with respect to safer alternatives such as Treasury Bills, where profit can be known at the time of hiring [1].

Risk means that losses may occur in the invested money, and then identify the behavior of stock prices becomes a fundamental activity for investors.

Perform an analysis of stock price fluctuation behavior, can reduce the risk in stock exchange market, because this behavior directly affects the portfolio profit. A stock portfolio composition that can provide good returns on investments is the investors' objective [2].

Predicting what will be the trends of stock price movements (if stock price will fluctuate in a positive or negative or even stagnate), is undoubtedly the biggest

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 352–359, 2012.

challenge for investors. An architecture that can identify such trends, it becomes extremely important at the time to define an investment strategy and select a stock portfolio that will bring good returns and low risk.

As pioneer of techniques to analyze stock prices can be cited, Harry Markowitz, that originated the called Modern Portfolio Theory [3]. His theory is applied today as a base to support decision making [4].

Since then, there were other thoughts and techniques in order to assist the investor select stocks, such as systems that use artificial intelligence techniques.

Application of artificial intelligent techniques such as artificial neural networks (ANNs) to predict stock prices trends led to interesting results [5, 6, 7, 8]. Other intelligent techniques can also be applied to the problem [4, 9].

The objective of this paper was to create hybrid architecture with Markowitz Model and an artificial neural network (ANN) type Multilayer Perceptron (MLP) to forecast trends in stock prices of the Stock Exchange of São Paulo.

This architecture sought to unite the diversification of the stock portfolio offered by the Markowitz model's with the ability to learn and generalize the knowledge learned from ANN.

2 Markowitz Model

Harry Markowitz, in the 50s, was the pioneer of Modern Portfolio Theory, published his paper "Portfolio Selection" in the "Journal of Finance," which suggested the creation of a stock portfolio with minimum risk to investment, this effect can be achieved by diversifying the portfolio considering the correlation between the stocks [3].

The Markowitz Model considers three points for selecting stocks for the portfolio:

- Covariance between stocks
- Stocks average profit
- Standard Deviation of profit stocks

The joint variation of stock prices or covariance represents the relationship between pairs of stocks by measuring the performance of each stock in relation to variation in another stock. When the value of a stock is raising the price (uptrend), another tends to fall (downtrend), so that gains with the stock in an uptrend can hedge the loss with the other downtrend price stock, in order to eliminate or minimize the diversifiable risk.

The expected portfolio return is obtained by calculating the weighted average of stock returns, according to equation (1) [3].

$$RP = \sum_{i=1}^{n} RA_i \,.\, Wi \tag{1}$$

In the equation (1) variable RA represents the return of a stock and variable W is the participation of the stock in the portfolio.

Last point considered by Markowitz is the standard deviation of stock profit, making possible evaluate the risk of portfolio formed.

The stock portfolio diversification minimizes the risk of getting lost in amounts invested, creating a security for investment such that only the market risks can be considered real risks for investment.

3 Artificial Neural Networks

ANNs are inspired in brain models structures aiming to simulate the human behavior in processes such learning, adaptation, association, tolerance fault, generalization and abstraction [10].

An important feature of ANNs is ability to learn. A general definition of what is to be learning in an ANN can be expressed as follows: "Learning is the process by which ANN parameters are adjusted by continuing stimulus for the environment in which the network is operating, and the specific type of learning that is defined by the particular way in which occur the adjustments made in the parameters" [11].

ANNs use a set of data corresponding to a sample input and output signals to the system during the training stage.

For training, the network uses algorithms of training or learning, which form a well-defined set of procedures for adapting the weights of an ANN so that she can learn a certain function [12].

As a result of learning, the ANN will produce similar output values to the data set to values that are equal to training samples.

Intermediate values, the network will produce an interpolation; the ANNs can learn by example and make interpolations and extrapolations of what they learned.

4 Multilayer Perceptron

MLP consists in a number of neurons, constituting the input layer, one or more hidden layers, and an output layer, where input signal propagates through ANN layer by layer.

This network can be trained using initial values for the connections of random weights. The parameters are initialized and the learning patterns of training data vectors are presented to the ANN. Throughout training progress weights connections are adjusted and performance can be monitored.

Error backpropagation is the training algorithm used in MLP. It works as follows: firstly it presents a pattern to the input layer of the network. This pattern is processed layer by layer until the output layer provides the answer processed (f*MLP*), calculated according to equation (2) [10].

$$f_{MLP(X)} = \varphi\left(\sum_{i=1}^{N_{on}} v_i . \varphi\ \left(\sum_{i=1}^{N_{em}} w_{ij} x_i + b_{i0}\right) + b_0\right) \tag{2}$$

Training process aim is to choose appropriate parameters to minimize a cost function pre-determined. This function is dependent on the desired response, and if there is error, this is calculated. Function of the square error sum is the most usual, according to equation (3).

$$E(X) = \sum_{i=1}^{N} \frac{1}{2} [f_{MLP}\ (x_i) - \ y_i]^2 \tag{3}$$

The calculated error is retropropagated from output layer to input layer and when this happens, the weights are adjusted and the processing is done again, until they obtain a minimum error [10].

MLP is used to classify data not linearly separable because it uses a non-linear activation function, typically a sigmoidal function, according to equation (4).

$$f(v) = \frac{1}{1+e^{(-av)}} \quad (4)$$

For this reason, MLP has great ability in dealing with nonlinear data and generalization, which are interesting features for applications in stock exchange.

Markowitz Model has some limitations as not process more than one input variable, which does not happen with the MLP. Another limitation is that the model works only with linear calculations, disregarding information such as dividend policies of firms, capital structure, market competitors, which may be considered by the MLP.

Thus, it was created hybrid architecture with objective to unite the security offered by Markowitz model in portfolio diversification with ability of learning and generalization brings from MLP.

5 Methodology

MLP tests were made by NeuralTool software version 5.7 trial [13, 14, 15]. In Markowitz model tests were used Lingo software version 10.0 [16, 17, 18, 19].

The extracted data reflects the information of the 10 stocks with the highest participation in the theoretical portfolio of Ibovespa (indicator of average performance of stock prices in Brazilian market) from 1st to 31th March, 2011, available in database from Stock Exchange of São Paulo [20]. The list of stocks used in this paper is shown in Table 1:

Table 1. List of 10 Highest Stocks

Stock Negotiation Code	Stock Company	Percentage of stock in Ibovespa Portfolio
PETR4	PETROBRAS	10,184 %
VALE5	VALE	9,948 %
OGXP3	OGX PETROLEO	4,476 %
ITUB4	ITAUUNIBANCO	4,101 %
BVMF3	BMFBOVESPA	3,759 %
BBAS3	BANCO BRASIL	3,114 %
BBDC4	BRADESCO	3,103 %
PETR3	PETROBRAS	2,961 %
GGBR4	GERDAU	2,824 %
USIM5	USIMINAS	2,777 %

Ibovespa index represents the performance of a theoretical portfolio composed by stocks most traded on the Stock Exchange of São Paulo. The stocks that compose Ibovespa theoretical portfolio represent over 80% of the trades in stock exchange of Sao Paulo, and then are used as an indicator of average performance of the market [20, 21, 22].

Following, experimental methodology was adopted: information from 10 stocks is processed on Markowitz model and the result is presented as one of the MLP input variables, thus establishing the hybrid architecture.

For processing stocks information with MLP, the following architecture was established: 10 neurons in input layer, two hidden layers with 15 neurons each, and maximum number of 3000 cycles (epochs). MLP learning rate of 0.03, the error value 0.001.

The file containing the information was divided into two parts (training and testing), the information from 1st to 15th March (110 records) were used to train the MLP and the second part of the file (from 16th to 31th March) also with 110 records for testing.

6 Experiments and Results Discuss

The three stocks in Table 2 compose a profitable and security portfolio to investors, this compose was obtained by the Markowitz methodology. Percentage of each stock in portfolio must be as the values at column called "Percentage of stock in Markowitz Model Portfolio" in Table 2, this values is a result by processing stock information in the Lingo software.

Table 2. Result from Markowitz Model Process

Stock Negotiation Code	Percentage of stock in Markowitz Model Portifolio
ITUB4	59 %
BVMF3	28 %
USIM5	14 %

Result obtained by Markowitz model (Table 2) is presented as a MLP input variable. MLP also received as input variables average values from information attributes listed below regarding the ten stocks selected for the experiments:

- Initial Price
- Higher Price
- Lower Price
- Average Price
- Best bid of buy
- Best bid of sale
- Final price (at the end of day trade)
- Diary variation price

Processing information with MLP obtains stocks with an uptrend, listed in Table 3, thus forming a profitable portfolio insurance obtained with an indication of stocks uptrend made by MLP with the diversification of the Markowitz model.

With the portfolio formed remains the investor to determine what percentage each stock will represent in his portfolio. This value is determined according to equation

(6), which is effected by a proportional calculation based on the percentage that each stock had in the Ibovespa theoretical portfolio in April (column "Percentage of stock in Ibovespa Portfolio" Table 1). This value is added and represented by the variable "wa_i" equation (5).

$$F(X) = X * \frac{100}{\sum_{i=1}^{n} wa_i} \quad (5)$$

Variable "n" is assigned the total number of stocks that will compose the portfolio. Variable "X" receives the stock percentage in the theoretical portfolio Ibovespa, this value is passed in a unified way until all four stocks shown in Table 3 have their percentage calculated.

As a result of equation "F (X)" is the percentages of the stock in portfolio formed by hybrid architecture. Thus, the portfolio is formed, ready to be implemented in the investment simulator [23]. The Percentage that each stock represents in the portfolio is displayed in the column called "Percentage of Stock in Portfolio "of Table 3.

Table 3. Portifolio composed by hybrid architecture

Stock Negotiation Code	Percentage of Stock in Portfolio
BVMF3	17,73
OGXP3	21,11
PETR4	48,04
USIM5	13,10

This portfolio will be used for practical application in a real-time investment simulator [23] from 23rd to 26th May, 2011 in order to verify the actual return on investment.

An estimate for portfolio return of 2.78% can be calculated based on the Markowitz model, which performs an average of the daily variations of the period for which data were extracted [3]. This way, the investor can know his possible profit.

In order to verify that returns from portfolio formed by hybrid architecture of MLP with Markowitz model are good, these returns are compared during the testing period (from 23rd to 26th May, 2011) to Ibovespa that depicts the performance of principal stocks traded on the Stock Exchange of São Paulo, indicator of the average Brazilian market behavior [20, 21, 22].

The portfolio obtained with hybrid architecture brought a gain of 1.76% on investment, when compared to the return obtained by the Ibovespa theoretical portfolio in the same period.

Then in Fig. 1 can be observe 4.15% as return to investment portfolio formed based on hybrid architecture of the Markowitz model with the MLP.

Was obtained a 2.39% return for the traditional index Ibovespa used as a parameter of the Brazilian economy, thus making the implementation of hybrid architecture of Markowitz model and MLP a valid alternative to support decision making of the investor, which wants to get as much profit as possible, minimizing the investment risk.

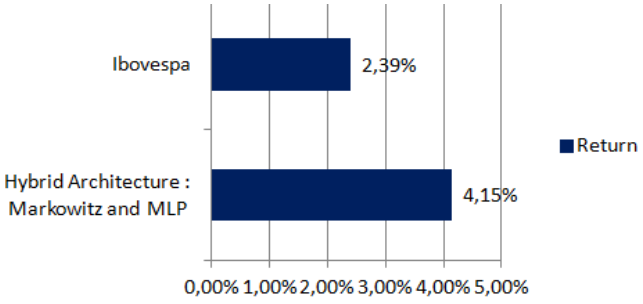


Fig. 1. Investment return (Test period)

7 Conclusion

Selection of stocks to compose a portfolio is a challenge for investors, since the variation of stock prices is constant, making it difficult to identify the stocks that will have a positive change, in order to bring profits for investment.

Identify in advance the behavior of stock prices can bring investors a competitive advantage in this Market. Then this anticipation becomes a differential that can bring gains to the investor who has such knowledge, when compared to the real investors who have no such knowledge.

The development of this paper enabled us to verify that hybrid architecture of the statistical model (Markowitz model) combining with an Artificial Intelligence technique, in this case the MLP, can bring satisfactory results in support of decision making, because it may indicate trends in stock price, helping the investor to pick stocks with uptrend to compose their portfolio, considering the security of investment, which is obtained with the methodology proposed by Markowitz.

Continuity of this research is interesting, since the results from hybrid architecture were positive. As a continuation of this work, you can check the results using different settings for the MLP and different input variables and also validation in different economic times.

Acknowledgments. This work was supported by the Nove de Julho University.

References

1. Bodie, Z., Kane, A., Alan, J.: Investments, 8th edn. The MacGraw-Hill Companies Inc., New York (2009)
2. Li, Z., Ni, W.: Research on Optimizing Security Investment Combination Based on PSO. Published in Second International Workshop on Knowledge Discovery and Data Minings. IEEE (2009), doi: 10.1109/WKDD
3. Markowitz, H. M.: Portfolio Selection. Journal of Finance 7, 77 (1952)
4. Kuang, L., Chien, L.: A Fuzzy Decision Maker for Portfolio Problems. IEEE (2010)

5. Martinez, L.C., Hora, D.N., Palotti, J.R.M., Meira, W.J., Pappa, G.L.: From an Artificial Neural Network to a Stock Market Day-Trading System: A Case Study on the BM&F BOVESPA. In: Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA. IEEE (2009)
6. Khan, A.U., Bandopadhyaya, T.K., Sharma, S.S.: Technical Indicators Based Hybrid Model gives better returns on Investments as Compared to BSE-30 Index. In: Third International Conference on Knowledge Discovery and Data Mining. IEEE (2010)
7. Hanif, T., Tahersima, M., Morteza, F., Navid, H.: Forecasting Stock Exchange Movements Using Neural Networks: A Case Study. In: International Conference on Future Computer Sciences and Application. IEEE (2011)
8. Gao, Z., XU, X.: Stock Bubbles' Nature: A Cluster Analysis of Chinese Shanghai A Share Based on SOM Neural Network. In: International Conference on Business Intelligence and Financial Engineering. IEEE (2009)
9. Khan, A.U., Bandopad, T.K., Sharma, S.: Classification andIdentification of Stocks using SOM and Genetic Algorithm based Backpropagation Neural Network. IEEE (2008)
10. Haykin, S.: Neural Networks: A comprehensive Foundation, New York (1994)
11. Mendel, J.M., Mclaren, R.W.: Reinforcement-learning control and pattern recognition systems. In: Adaptive, Learning and Pattern Recognition Systems, vol. 8, pp. 287–318. Academic Press, New York (1970)
12. Bigus, J.P.: Data Mining with Neural Network: Solving Business Problems from Applications Development to Decision Support. Mcgraw-Hill (1996)
13. Pereira, A.: Health and economic impact of posttransfusion hepatitis B and codt-effectiveness analysis of expanded HBV testing protocols of blood donors: a study focused on the European Union. The Journal Transfusion (2003)
14. http://www.palisade-br.com (acessed from May 18, 2011)
15. Aihua, W., Pochec, P.: Investigation of Data Transmission Logs Using the Best fit Package. In: Conference on Electrical & Computer Engineering. IEEE (2002)
16. Yang, G.: The Optimization Analysis of Supply Chain Reliability Based on Economic Constraints. IEEE (2011)
17. http://www.lindo.com (accessed from May 19, 2011)
18. Wang, Q., Liu, X., An, S.: Data Analysis and capital budgeting model. In: International Conference on Intelligent Human-Machine Systems and Cybernetics. IEEE (2009)
19. Wang, T., Yang, X.: The Study of Model for Portfolio Investment Based on Ant Colony Algorithm. In: International Conference on Future Computer/Communication. IEEE (2009)
20. http://www.bmfbovespa.com.br/ (accessed in May 19, 2011)
21. Oliveira, E.M.J., Ludermir, T.B.: Forecasting the IBOVESPA using NARX networks and random walk mode. In: SBRN. IEEE (2002)
22. Fox, E., Sudderth, E.B., Jordan, M.I., Willsky, A.S.: Bayesian Nonparametric Inference of Switching Dynamic Linear Models. IEEE Transactions on Signal Processing (2011)
23. http://folhainvest.folha.com.br/ (accessed from May 23-26)

Comparative Study of Methods for Segmentation of Digital Images of Birds

Felipe de Sousa Nobre, Paulo César Miranda Machado, and Rodrigo Pinto Lemos

Escola de Engenharia Elétrica e da Computação – Universidade Federal de Goiás (UFG) Caixa Postal 131, 74001-970 - Goiânia – GO – Brasil
{felipesnobre,pcmmachado,rodrigo.p.lemos}@gmail.com

Abstract. The observation of birds is important not only for academic reasons, but also for the monitoring of preservation areas. An important step for developing a program for automatic recognition of birds is the segmentation of the image, that is, highlighting the bird from the background in a digital image. This paper aims to present a comparative study using methods of segmentation in digital images of birds. In this work, a method based on cellular automata is compared with other method based on active contours.

Keywords: Image segmentation, digital images, birdwatching.

1 Introduction

According to research conducted by the U.S. Fish and Wildlife Service in 2006, 21% of the U.S. population is considered birdwatchers. In the evaluated year, $36 million was spent by these people in activities related to bird watching [12] It is known that in Brazil the number of bird watchers has attracted much attention [1]. Whether for hobby or research, the automatic recognition of bird monitoring in their natural habitat is an area of research that gains focus worldwide. It is observed by the research literature, that the most common area of recognition of birds is by singing, being not so common the research using image. However, it must be remembered that not all birds sing throughout the whole year and some birds simply do not sing during any phase of life, so other ways of recognition must be developed in these cases.

Pattern recognition can be defined as "the scientific discipline whose goal is the classification of objects in a number of categories or classes" [4,11]. In digital images, we must first obtain the portion of the image that is sent to a pattern recognition algorithm in a step called segmentation. The segmentation goal is to separate the image into different objects for further analysis. In this paper, the segmentation is used to highlight the bird from the background.

Due to the little literature on researching the automatic recognition of birds in digital images, this paper sets out to compare and analyze segmentation methods already known in the area.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 360–365, 2012.

Two segmentation methods are presented: a cellular automaton-based method that requires user interaction for the segmentation and another method based on active contours, in which user interaction is restricted to initialization parameters.

2 Theory

The Growcut method is interactive: the user selects a set of pixels, called seeds, belonging to an object to separate it from the background, which also must be marked. The method uses cellular automata to solve the task of classifying the pixel [13]. The Growcut method application used a toolbox developed by Lankton [9] that was implemented based on the algorithm by Vezhnevets and Konouchine[13].

Before the explanation of the Growcut algorithm, basic definitions about digital image and cellular automata should be made. A digital image is represented as a bidimensional matrix *k x m* and each matrix element is called picture element or pixel. Each pixel contains values that vary accordingly to the luminosity intensity in that particular point of the image. Different types of color maps can represent color images.In this paper, the RGB color map was adopted. This color map is represented by 3 bidimensional matrices each containing the luminosity of the colors red, green and blue, respectively, therefore the name RGB. Also, each pixel holds an 8 bit integer number in the range [0, 255].

Cellular automata are a computational model for real experiments simulation in time, space and state [5]. In Growcut method, our system is analysed as a bidimensional matrix with the same dimensions of the image, *k x m*. Each matrix element is called cell and contains a state set, a neighborhood and a transition function that determines the state of the cell in time $t + 1$, based on the state of each neighborhood cell at time t. This way, each cell is defined by $A = (S, N, \delta)$, where S is the state set, N is the neighborhood system and $\delta: S^N \rightarrow S$ is the transition function.

Also, S is defined as a set of values that determine the cell state, where l is the label of the cell, θ is the strength of the cell and the vector $\vec{C}$ is the standard vector of the cell. The label l is used to differentiate the cell as background, indifferent or foreground, assuming the values {-1, 0, 1}, respectively. Cell's strength values θ vary in the interval [0, 1]. The vector $\vec{C}$ is the tridimensional vector in the RGB color space of the pixel corresponding to the evaluated cell.

The Growcut method segments the image from the seeds, labeled at the beginning of the method. Each cell attacks the neighborhood cells and changes the values of their label based on the attack force of the attacking cell, F_a, and the defense force of the defending cell, F_d. The attack force is defined as function of the attacker's cell strength, θ_p , and the distance between the attacker's cell, $\overrightarrow{C_p}$, and defender's cell, $\overrightarrow{C_q}$. The defense force, F_d, is the strength of the defender's cell, θ_q, where p is the attacking cell and q is the defending cell.

$$F_a = g(\left\|\overrightarrow{C_p} - \overrightarrow{C_q}\right\|_2).\theta_p^t \quad (1)$$

$$F_d = \theta_q^t \quad (2)$$

Also, g is a monotonous decreasing function limited in [0,1] [13]:

$$g(x) = 1 - \frac{x}{max\|\vec{c}\|_2} \tag{3}$$

From the definitions above, the following pseudo-code can be created:

```
// For each cell
for ∀p ∈P // where P is the set of all pixels of the image
l_p^{t+1} = l_p^t ;
θ_p^{t+1} = θ_p^t ;
 for ∀q ∈N(q)
  if g(‖C_p⃗ − C_q⃗‖_2) * θ_q^t > θ_p^t
     l_p^{t+1} = l_q^t ;
     θ_p^{t+1} = g(‖C_p⃗ − C_q⃗‖_2) * θ_q^t;
  end if
 end for
end for
```

In the method of segmentation with active contour, or also called snakes, created by Kass et al.[6] a contour deformation *Co* is used to detect objects. This deformation is accomplished by minimizing an energy function so that it's local minimum is obtained in the vicinity of the object. Several studies have been conducted to find more effective energy to be minimized [2] and [3] or even improve the performance and speed of the algorithm [7] and [14].

The method developed by Chan and Vese[3] obtain more robust results when compared to the original study developed by Kass et al., achieving satisfactory results in noisy images and it does not depend on the initial position of the contour. This method is developed using level set methods that are easy to implement but are slow to compute [8]. A disadvantage of the method proposed by Chan and Vese is that it does not provide satisfactory results in images with heterogeneous background and objects.

For the study of these methods, the toolbox proposed by Lankton[7] and Lankton and Tannenbaum[8] was used implementing the level set methods for the evolution curves using the Sparse Field method proposed by Whitaker[14] to perform faster and more efficient computation.

Level set methods are used to represent a 2D contour curve *C* into a 3D surface Φ, where the contour *C* is the zero level set of the chosen surface [10]. The surface Φ is usually chosen as the signed distance function, so that, for internal values of *C*, $\Phi < 0$, and, for external values of C, $\Phi > 0$. Figure 1 represents the surface Φ and the contour *C*.

The Sparse Field method does not use all values of the surface Φ, but only few levels close to zero in order to improve the performance of the method without greatly

affecting the outcome [14]. It uses doubly linked lists to store the points belonging to the zero level set and two levels above and two levels below the zero level set.

In addition, the toolbox proposed by Lankton has a more satisfactory result than the one developed by Chan and Vese in order to target heterogeneous backgrounds and objects. According to Lankton, this is due to use of local characteristics beyond the global characteristics that used in the method developed by Chan and Vese.

Fig. 1. The circle represents the contour C_0 and bellow is the representation of the function SDF

3 Result and Discussions

The segmentation was performed on digital images of birds found in the region of the State of Goiás, such as *azulona* (*Tinamus tao*), *inhambu-chororó* (*Crypturellus parvirostris*), *jaó* (*Crypturellus undulatus*), among others.

To implement the Growcut method, several pixels were marked on the bird and also several pixels on the background. The amount of pixels marked varied from image to image in order to find the best result. The active contour methods were implemented using rectangular contour regions containing the bird or regions where the bird was contained.

The Growcut method obtained more satisfactory results than those obtained with the method of active contour, as shown in figures 2 and 3. The Growcut method obtained good segmentation of the birds in the images. Even the errors shown in the pictures above can be minimized choosing a bigger amount of marked pixels.

From the pictures above it can be seen that the segmentation with active contour method didn't achieve satisfactory results for birds. The method in the first picture couldn't differentiate the bird from the background because of the similarity between the bird's color and the background's color. In the other pictures, it can be observed that the contours couldn't capture the difference between the color varying along the bird and the color varying from the bird to the background. That resulted in incomplete segmentation.

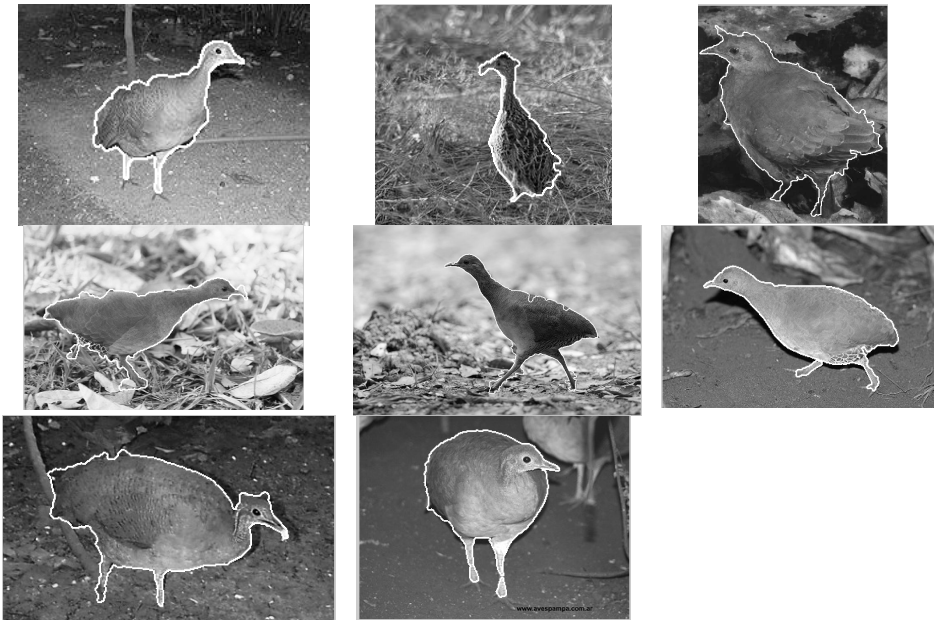


Fig. 2. Segmentation of birds using Growcut method

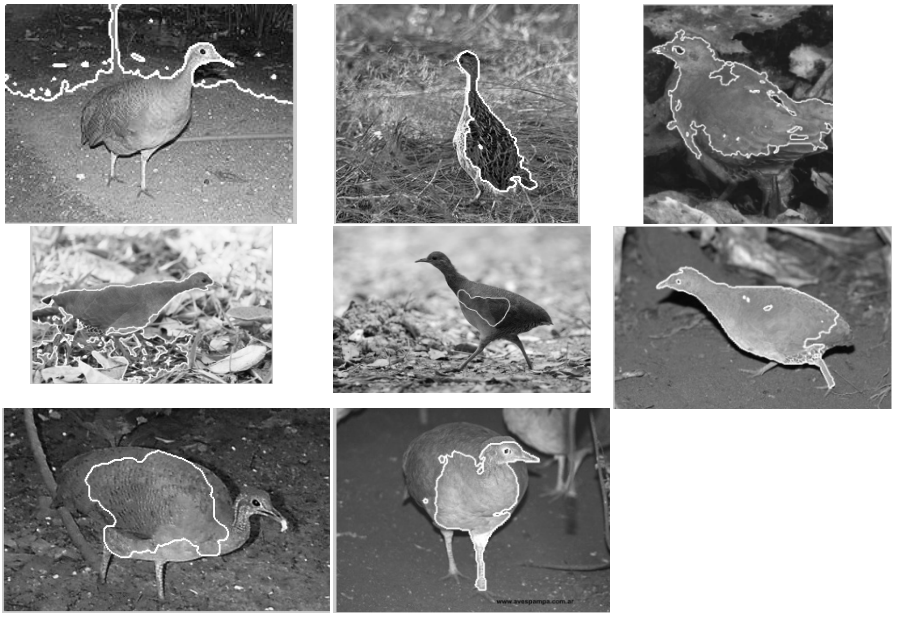


Fig. 3. Segmentation of birds using Active Contour method

4 Conclusions

Although we hadn't got perfect results for segmentation of the birds, the study is promising. Much of the error is due to the limitations of the methods studied or due to the complexity of the objects to be segmented. Birds have very different characteristics in color, for example, some of which have various colors along its length or colors that blend with the background.

The Growcut method was the one that achieved better results to outcome the problem of the various colors present in birds, also it work with N-dimensional images, i.e., with different color spaces. However, he was still limited because it requires user interaction for the perfect segmentation.

The active contour method proved quite efficient in being used without user interaction. However, it is limited to grayscale images, and also presents wrong segmentation of birds with very different tones. Perhaps this last limitation is due to the sensitivity of the method's initial parameters.

References

1. Birdwatcher, http://www.birdwatcher.com.br
2. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. 22(1), 61–79 (1997)
3. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. on Image Processing 10(2), 266–277 (2001)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification, 2nd edn. Wiley, New York (2001)
5. Ermentrout, G.B., Edelstein-Keshet, L.: Cellular automata approaches to Biological modeling. J. Theor. Biol. 160, 97–113 (1993)
6. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. of Computer Vision 1(4), 321–331 (1988)
7. Lankton, S.: Sparse Field Methods. Technical Report (2009)
8. Lankton, S., Tannenbaum, A.: Localizing Region-Based Active Contours. IEEE Trans. on Image Processing 17(11), 2029–2039 (2008)
9. Matlab Central, http://www.mathworks.com/matlabcentral/fileexchange/19091-growcut-image-segmentation
10. Osher, S., Sethian, J.A.: Front Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. Journal Computational Physics 78, 12–49 (1988)
11. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Elsevier (2009)
12. U.S. Fish & Wildlife Service, http://library.fws.gov/pubs/birding_natsurvey06.pdf
13. Vezhnevets, V., Konouchine, V.: 'GrowCut' – Interactive Multi-Label N-D Image Segmentation by Cellular Automata. In: Graphicon 2005, Russia (2005)
14. Whitaker, R.: A level-set approach to 3D reconstruction from range data. Int. J. Comput. Vis. 29(3), 203–231 (1998)

A Robust AdaBoost-Based Algorithm for Low-Resolution Face Detection

Diego Alonso Fernández Merjildo and Lee Luan Ling

Department of Communications, DECOM
School of Electrical and Computer Engineering, FEEC
State University of Campinas, UNICAMP
Campinas, Zipcode: 13083-852, Brazil
{diegofer,lee}@decom.fee.unicamp.br

Abstract. This work presents a face detection algorithm based on Multiscale Block Local Binary Patterns (MB-LBP) and an improved AdaBoost algorithm. The proposed boosting algorithm is capable of avoiding sample overfitting over its training process. This goal is achieved by making use of the information of sample misclassification frequency to update the weight distribution in the training process. Experimental results evidence some advantages of the proposed method over the classical AdaBoost algorithms, including the generalization capacity, overfitting avoidance and high precision rate on low-resolution images.

Keywords: Face Detection, GentleBoost Algorithm, Frequency Factor, Low-Resolution Images, Machine Learning.

1 Introduction

Currently there are many face detection methods highly competitive and efficient in terms of accuracy and speed. However, face detection in images or video is still a challenging problem, especially when images exhibits low resolution, object faces are distant, and cameras need to cover wide view angles.

The face detection method developed by Viola and Jones [1] is probably the most successful approach in this area. They applied an AdaBoost algorithm for feature extraction and achieved competitive object detection rates for real-time applications. The adaptive boosting algorithm proposed by Li et al. [2] focused on the samples being misclassified. They used a threshold to assign large weights to misclassified samples. Such a updating weight procedure may cause a considerable distortion over weight distribution and; therefore, sample overfitting due to the cumulative error acquired from repeated iterations.

In security applications, it is desirable having fast face detection even when those faces are distant from the camera, and the camera need to cover a wide view angle. In others words, we have cases of low resolutions and non-frontal faces images. In this sense, Hayashi and Hasegawa [3] studied the relationship between the resolution and the detection rate for low-resolution images and proposed a dedicated detection technique.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 366–373, 2012.

In this paper, we proposed a modified AdaBoost Algorithm in order to overcome the above described problems and to improve the face detection performance. We compare our approach with two conventional boosting methods (Real AdaBoost and GentleBoost) especially for low-resolution images. Thus, we show experimental results and the application of our method for distinct databases.

2 The Proposed Classification System

In this section, we describe the proposed classification system in terms of the employed pattern feature set and the modified classification algorithm.

2.1 Multiscale Block Local Binary Patterns

Local Binary Pattern (LBP) operator, as described by Ojala et al. [4] uses the relationships between a pixel and its neighborhood pixels in image discretization; for each neighbor pixel a value (0 or 1) is assigned according to its gray intensity. Thus, taking a value of 1 if pixel intensity is higher than the central pixel intensity, and 0 otherwise. In addition, the binary values associated with the neighbors are read sequentially, clockwise, to obtain a binary or decimal number. This operator represents fundamental properties of local image texture; however, it may be too local to be robust.

In order to overcome the limitations of LBP and to be able to apply it to face analysis, Liao et al. [5] proposed a Multi-scale Local Binary Pattern operator. It is an extension of the Local Binary Pattern (LBP) operator using different sizes to describe neighborhoods. The MB-LBP operator uses rectangular regions of pixels instead of single pixels. Figure 1 shows a MB-LBP type filter composed of 9 rectangles.

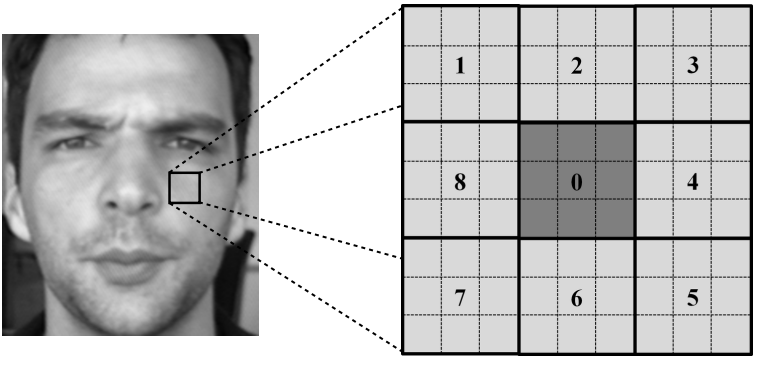


Fig. 1. The *MB-LBP* operator proposed by Liao et al. [5]

In a MB-LBP approach features are computed according to the average intensity of central rectangle g_c and averages intensity of each neighborhood rectangles $\{g_1, \ldots, g_{p-1}\}$, where the notation (p, r) denotes a neighborhood of p equally spaced sampling points on a circle of radius r. As illustrated by (1), $p = 9$.

$$MB\text{-}LBP_{(p,r)} = \sum_{p=1}^{p-1} s(g_p - g_c)2^p, \quad s(x) = \begin{cases} 1 & , x \geq 0 \\ 0 & , x < 0 \end{cases} \tag{1}$$

The MB-LBP features can also be calculated rapidly through an integral image. These features capture more information than Haar-like features and provide a better representation than the basic LBP operator. The reason is that micro-structures and macro-structures are encoded simultaneously.

In compilation an AdaBoost algorithm each weak classifier $h_i(x)$ implements its a feature f_i, through an MB-LBP operator under threshold θ_i and parity p_i. If $p_i f_i \leq p_i \theta_i$ then $h_i(x) = 1$, in otherwise $h_i(x) = -1$.

2.2 Modified AdaBoost Algorithm

A classical Adaboost algorithm combines several weak classifiers to build a strong classifier capable of providing efficient generalization performance. The algorithm is trained by a dataset here denoted by $x_i \in \chi, i = 1 \ldots N$, with labels $y_i \in \{-1, 1\}$.

Iteratively, at each training round $t = 1, \ldots, T$ a new weak classifier $h_t(x_i) : \chi \to [-1, 1]$ is added, aiming to minimize the following error function:

$$\epsilon_t^2 = \sum_{i=1}^{N} W_t(i)(y_i - h_t(x_i)) \tag{2}$$

where $W_t(i)$ is the weight assigned to x_i by the *t-th* weak classifier. Thus, weights are updated according to (3):

$$W_{t+1}(i) = W_t(i) exp(-\alpha_t h_t(x_i) y_i)/Z_t \tag{3}$$

where Z_t is a normalization factor and α_t measures the importance level of the *t-th* weak classifier. We can rewrite (3) as follows:

$$W_{t+1}(i) = \begin{cases} W_t(i) exp(-\alpha_t) & \text{if } h_t(x_i) = y_i \\ W_t(i) exp(\alpha_t) & \text{otherwise} \end{cases} \tag{4}$$

Dietterich [6], Ratsch et al. [7] and Servedio [8] showed that AdaBoost algorithms has great tendencies to cause sample overfitting in the presence of high noise data; that is, the performance on already trained examples increases whereas the performance on unseen data becomes worse. In a classical Boosting procedure when a sample is misclassified, the associated weight is increased; on the other hand, when a sample is correctly classified, the associated weight is diminished as shown by (4). However, this weight updating strategy may cause sample overfitting.

With the purpose of avoiding the sample overfitting problem, this work suggests a modified method to update weights distribution, considering a new variable (denoted by δ_i) associated to each training sample, as shown by (5).

$$W_{t+1}(i) = \begin{cases} W_t(i)exp(-\alpha_t) & \text{if } h_t(x_i) = y_i \\ W_t(i)exp(\alpha_t) & \text{if } h_t(x_i) \neq y_i \wedge \\ & \delta_i \geq k \\ W_t(i) & \text{otherwise} \end{cases} \tag{5}$$

Notice that, the set of new variables indicating the frequency of occurrence of misclassification ($\delta_i \in \Delta = \{0, ..., k\}$) is involved in the training and is responsible for preventing the increase of weights of misclassified samples. In other words, the weights are only changed when samples were in fact misclassified. The updating process proposed here is similar to that of the classical AdaBoost algorithm. At each iteration step, a new weak classifier is used with the set of weak classifiers that are linearly combined to build a single strong classifier (See Equation 5).

$$F(x) = \sum_{j=1}^{T} \alpha_j h_j(x) \tag{6}$$

Over the training process, the variable δ_i assumes its value belonging to $0, 1, 2, \ldots, k$, starting at zero and incremented by the unit when the sample is misclassified. The variable δ_i remains unchanged when the sample is classified correctly. The detailed update step of the variable δ_i is shown by (7).

$$\delta_i = \begin{cases} \delta_i + 1 & \text{if } h_t(x_i) \neq y_i \ \wedge \ \delta_i < k \\ 0 & \text{if } \delta_i \geq k \\ \delta_i & \text{otherwise} \end{cases} \tag{7}$$

3 Experimental Investigation and Results

Experimentally we evaluated our method with others two AdaBoost algorithm variants (Real AdaBoost and GentleBoost). A total of 8094 images were used for training, these image dataset were compiled by selecting images from the following databases: PICS Dataset [9], UMIST Dataset [10] BioId Dataset [11], FEI Dataset [12] and dataset used in [13]. Each selected image was rescaled under 24×24 pixel resolution. The resulted dataset has the following composition: 2526 positive samples (faces) and 5568 negative samples (non faces).

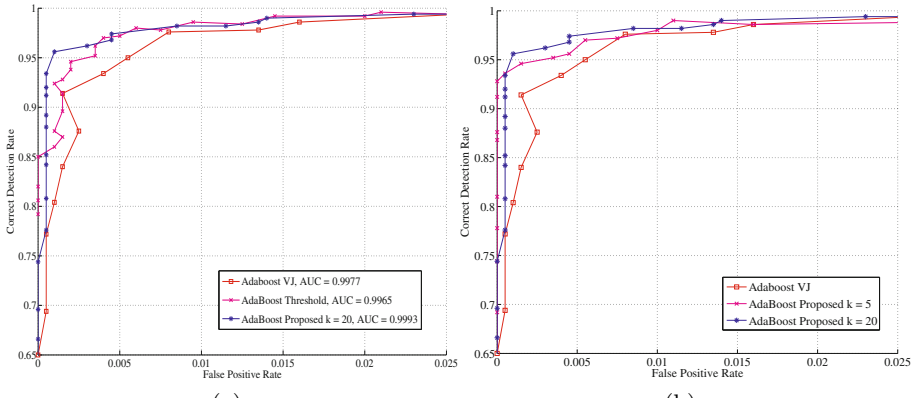
The first experiment focuses on behavior comparative analysis between the Real AdaBoost algorithm and the proposed one. Notice that all weak classifiers adopted Haar pattern features. Figure 2a compares ROC curves of the classical Real Adaboost VJ algorithm [1], the Real AdaBoost Threshold algorithm [2]

and the proposed one. In terms of AUC performance, according to Fig. 2a, we have, respectively, AUC value of 0.9978, 0.9965 and 0.9993. In addition, the proposed method offers faster convergence than the others, the classical Real Adaboost VJ algorithm and the AdaBoost Threshold algorithm. Table 1 shows three particular performance points of three investigated methods in terms of the number of positive and false face detections.

Table 1. Detection Rate Comparison, False Positive and Accuracy, using Haar-like classifier

Method \ Parameters	Faces	Hits	Non Faces	False Alarms	Detection Rate
Real AdaBoost	500	420	2000	14	84.0 %
Adaboost [2]	500	470	2000	16	93.99 %
Frequency Real AB, k=20	500	484	2000	18	96.79 %

Figure 2b plots three ROC performance curves one of the classical Real Adaboost VJ algorithm and two provided by the proposed method for two distinct values ($k = 5$, and $k = 20$). The proposed method with $k = 20$ outperforms other two for a given false positive rate. First test results are summarized in Table 1. It presents the performance of the classical Real Adaboost VJ, the Real AdaBoost Threshold and the proposed one. It shows respectively 93.5%, 95.5% and 96.5% of detection rate, and 0.7%, 0.8% and 0.9% of false positive rate.



(a) (b)

Fig. 2. ROC Curves comparing classifiers of classic Real AdaBoost VJ (Viola-Jones) [1], Real AdaBoost Threshold [2], and AdaBoost proposed, using Haar-like classifiers

The second part of our experimental investigation involved three different datasets, namely CMU (newtest), CMU (test-low) [14] and BAO [15]. This test focuses on performance comparison between the GentleBoost algorithm and the proposed one. Arbitrarily we adopted $k = 2$, MB-LBP features and a 15-level cascade structured classifier.

Table 2 shows comparative results using the BAO dataset. It is necessary mention that the images of the BAO dataset exhibit higher resolution than those of the CMU (newtest) and the CMU (test-low) datasets. According to Table 2, for the BAO dataset, the performance of our method was slightly higher that of the GentleBoost algorithm. This behavior is probably due to good images resolution of the BAO dataset.

Table 2. Detection Rate and False Positive comparison for suitable resolution images, Bao Face DataBase [15], using Multiscale Block Local Binary Patterns

Method \ Parameters	Faces	Hit	False Alarms	Detection rate
GentleBoost	349	257	38	73.63 %
Frequency GentleBoost, k=2	349	264	53	75.64 %

On the other hand, according to Table 3, the proposed method is considerably superior when the CMU (newtest) is used. It is important to highlight that the CMU (newtest) has a relatively poor image resolution.

Table 3. Detection Rate and False Positive comparison for low-resolution images, CMU Test Faces (newtest) [14], using Multiscale Block Local Binary Patterns

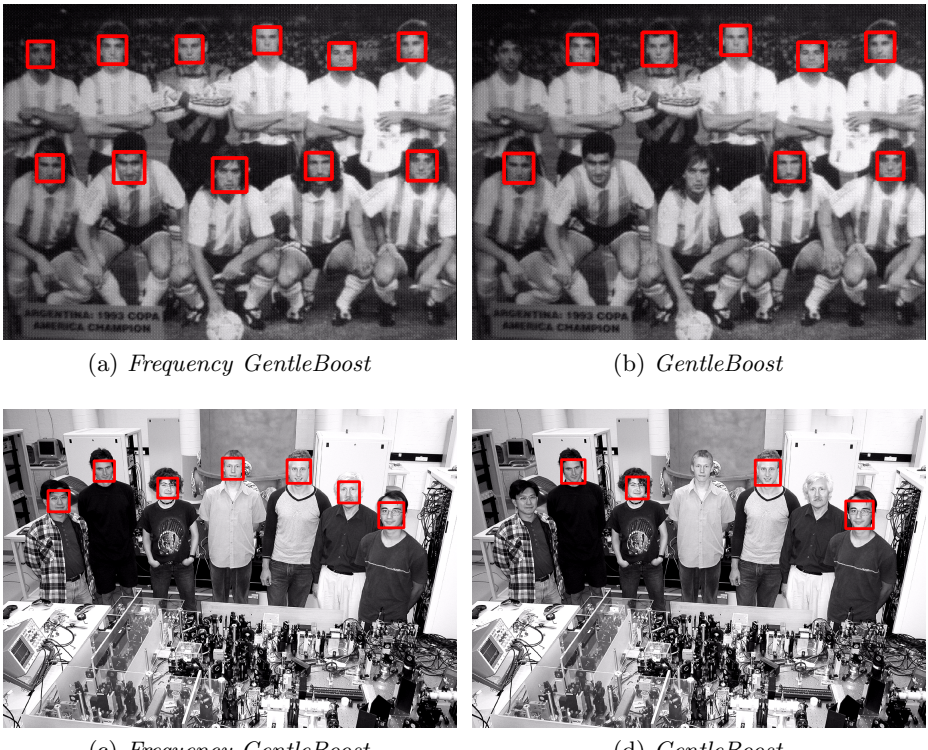
Method \ Parameters	Faces	Hit	False Alarms	Detection rate
GentleBoost	186	120	18	64.51 %
Frequency GentleBoost, k=2	186	139	23	74.73 %

Notice that the behavior of weak classifier tends to become even less effective when resolution images become very poor. This is revealed by Table 4 when the CMU (test-low) dataset was tested. A dramatic decline in detection rate is observed.

For illustration purposes, Figure 3 indicates successfully detected by the proposed method and the GentleBoost algorithm in two images of the CMU (test-low) and the BAO dataset.

Table 4. Detection rate and false positive comparison for low resolution images, CMU Test Faces (test-low) [14], using Multiscale Block Local Binary Patterns

Method \ Parameters	Faces	Hit	False Alarms	Detection rate
GentleBoost	157	45	13	28.66 %
Frequency GentleBoost, k=2	157	54	18	34.39 %



(a) *Frequency GentleBoost* (b) *GentleBoost*

(c) *Frequency GentleBoost* (d) *GentleBoost*

Fig. 3. Comparison of representative detection results

4 Conclusions

In this paper, we present a novel approach to update the weight distribution for AdaBoost algorithm. We consider that the weight updating process is a significantly important step on the feature selection process. Superior performance was achieved when the weights distribution of samples were updated appropriately. In terms of detection rate, our approach outperforms other classical boosting methods. This superiority becomes evident for low-resolution images.

References

[1] Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vision 57, 137–154 (2001)

[2] Li, G., Xu, Y., Wang, J.: An improved adaboost face detection algorithm based on optimizing skin color model. In: 2010 Sixth International Conference on Natural Computation (ICNC), vol. 4, pp. 2013–2015 (August 2010)

[3] Hayashi, S., Hasegawa, O.: Detecting Faces from Low-Resolution Images. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006, Part I. LNCS, vol. 3851, pp. 787–796. Springer, Heidelberg (2006)

[4] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)

[5] Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning Multi-scale Block Local Binary Patterns for Face Recognition. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 828–837. Springer, Heidelberg (2007)

[6] Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)

[7] Rätsch, G., Onoda, T., Müller, K.-R.: Soft margins for adaboost. Mach. Learn. 42, 287–320 (2001)

[8] Servedio, R.A.: Smooth boosting and learning with malicious noise. J. Mach. Learn. Res. 4, 633–648 (2003)

[9] PICS: Psychological image collection at stirling (January 2012), http://pics.stir.ac.uk

[10] UMIST: Face database (January 2012), http://www.sheffield.ac.uk/eee/research/iel/research/face

[11] BioID: Face database (January 2012), https://www.bioid.com/download-center/software/bioid-face-database.html

[12] FEI: Face database, http://fei.edu.br/~cet/facedatabase.html (January 2012)

[13] Samaria, F.S., Samaria, F.S., Harter, A., Site, O.A.: Parameterisation of a stochastic model for human face identification (1994)

[14] Rowley, H., Baluja, S., Kanade, T.: Rotation invariant neural network-based face detection. Technical Report CMU-CS-97-201, Computer Science Department, Pittsburgh, PA (December 1997)

[15] Frischholz, R.: Bao face database at the face detection homepage (January 2012), http://www.facedetection.com

Color Image Segmentation Using Gaussian Mixtures and Particle Swarm Optimization

Wesley Martins Teles[1,2] and Carlos Henrique Quartucci Forster[1]

[1] Instituto Tecnológico de Aeronáutica (ITA),
Program of Electronic & Computer Engineering,
São José dos Campos, State of São Paulo, Brazil
{wesley,forster}@ita.br
[2] Universidade Estadual de Goiás (UEG),
Unidade Universitária de Educação a Distância (UnUEAD),
Anápolis, State of Goiás, Brazil

Abstract. The model of Gaussian Mixture is particularly useful to perform unsupervised learning. Currently, the principal technique to estimate the mixture parameters is the Expectation Maximization method which has a great chance of obtaining sub-optimal results. In this work we opted, instead, for the Particle Swarm Optimization as an alternative way to estimate parameter of Gaussian Mixture applied to multivariate data, which has greater chance of reaching the optimum. To evaluate the proposed approach, color images from fluorescence microscopy are segmented considering the 3D color space. Some particular features of this kind of color image are also considered to improve the performance of the search.

Keywords: Particle swarm optimization, image segmentation, Gaussian mixture models, fluorescence microscopy, unsupervised learning.

1 Introduction

The Gaussian Mixture Model (GMM) is the most common type of distribution among mixture models. Its estimation from data points is often employed as a method of unsupervised learning that can be used in tasks that are not easy for common clustering methods. Its main advantage is the possibility of building soft boundaries between clusters, which would allow the use of the model to classify patterns by determining a probability value.

The usual technique to estimate the mixtures is the Expectation Maximization (EM) method [1][2]. However, EM is known to be costly and the convergence to global maximum (maximum likelihood, for instance) is not guaranteed. In fact, this method is expected to reach and stay at local maxima, which sometimes are good-enough solutions or eventually coincide with the optimum.

Thus, this work proposes the use of a different algorithm to estimate the parameters on multivariate data: using the Particle Swarm Optimization (PSO) metaheuristics as the search mechanism. The proposed technique, although still costly, has the

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 374–381, 2012.

advantage of better chance to converge to a global optimum, once PSO is classified as a global optimization algorithm [3][4], and the potential to do it very efficiently when correctly tuned [5]. In this work, the technique is applied to color image segmentation, exploiting the 3D space of pixel colors.

Image segmentation is an area in which GMMs are considered efficient. A technique based on the GMM to one data dimension using an adapted PSO version as search mechanism appears in [6], in order to do segmentation of grayscale images into black and white. In an evolution of this work [7], the authors propose the segmentation of color images as a challenge, with difficulties imposed by the segmentation of three-dimensional data.

In order to evaluate our approach to GMM, we analyze the problem of segmentation based only on individual pixel classifications by color. This would be called "thresholding" in the case of grayscale images [8]. For this, we chose images generated by fluorescence microscopy as such type of imaging was particularly designed to allow important regions of the image to be visually distinguished by color with certain ease. Differently from other kinds of color images, fluorescence microscopy images have a limited number of classes of regions that are determined by the color related to the fluorophore that are previously selected. This kind of imaging method has become popular to do cell image analysis due the development of lasing scanning that allows to obtain 3D image [9].

These typical characteristics of the imaging method can be used to increase the performance of the inference process. As the representative colors for each class are known beforehand, the unsupervised learning mechanism may be started with some initial information of the classes which is refined until the end of a complete learning process. This way, the search mechanism works to adapt the initial gross estimations with the pixel color data of a specific image, producing complete density model for each pixel class. This strategy not only reduces the number of iterations needed to estimate the parameters, but also the chance of reaching undesired local optima due to the fact that the number of classes and their central values are the harder to estimate parameters of the GMM, as the covariance matrix is easy to compute in the case the mean values are fixed.

In section 2 we present the details of PSO used to adjust the GMM. In section 3, we explain the basic differences between the method used for fluorescence microscopy images in this work and the method used for grayscale images. In section 4, we describe the experiment with color images. In section 5, we describe results of the comparison of EM and PSO. And, in section 6, we conclude and suggest future work.

2 Particle Swam Optimization and GMM

The PSO is a relatively new optimization method inspired by natural phenomena such as the migration of the birds, the movement of the shoals of fishes or the flight of a swarm of bees. It was proposed in 1995 by Kennedy an Eberhart [10]. It is considered a good method comparing with others popular optimization methods such as genetic algorithms [11][12].

The method basically consists of moving particles through the search space and annotating the positions corresponding to the best values of a fitness function. The best position of the route of each particle and a global best position are used at each iteration to calculate the next movement of the particles.

The position of each particle is updated using the equations (1) and (2) adapted from [13]:

$$V_{i+1} = w_i.V_i + cb.rand().(Xb_i - X_i) + cg.rand().(Xg_i - X_i) \tag{1}$$

$$X_{i+1} = X_i + V_{i+1} \tag{2}$$

Where *V* is the velocity of the particle at iteration *i*. *X*, *Xb* and *Xg* are respectively the current position, the best position of the route of the particle and the global best position of all particles at the iteration *i*.

The parameter *w* of PSO is an inertia weight that adjusts dynamically the velocity and controls the exploration of the search space. Small values help to avoid local minima and high values determine faster convergence. A high value should be used at the initial iterations and lower values are used at the final iterations. *Vmax* is the maximum velocity for each particle.

The parameters *cb*, *cg* are constants that control the points where the particle will be placed between *Xg* and *Xb*. With high *cb* the particle will tend to follow its own best partial solution, while with high *cg*, the particle will tend to follow the leader solution.

In this work, PSO is used to optimize the probability density function of the vector ***y*** of pixel color of a given image under a k-component mixture model (Equation 3) [14]:

$$f(y;\Psi) = \sum_{i=1}^{k} \pi_i f_i(y;\theta_i) \tag{3}$$

At the context of this work, each Gaussian of the mixture corresponds to one color cluster. In case of GMM, the function f_i has a Gaussian distribution given by:

$$f_i(y;\theta_i) = (2\pi_i)^{-\frac{d}{2}} \left|\Sigma_i\right|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}(y-\mu_i)^T \Sigma_i^{-1}(y-\mu_i)\right\}, \tag{4}$$

where $\theta_i = (\mu_i, \Sigma_i)$ is the component *i* parameter vector, $\pi_1 + \ldots + \pi_k = 1$ are the weights of the Gaussians, $\Psi = (\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_k)$ is the parameter of the mixture and *d* is the dimension of the data. The PSO searches for Ψ that minimize the fitness error function between *f* and the histogram *H* of color's value of the pixels given by:

$$fitness = \sum_{j} \left[H_j - F(x_{j+1};\Psi) + F(x_j;\Psi)\right]^2 \tag{5}$$

where F is the associated cumulative distribution function.

3 The Adaptation of the Method to Fluorescence Color Images

The method used to segment the images is similar to that used in [6] with the main difference that we worked with color images instead of gray scale. Although, this task is harder since the data is 3-dimensional instead of univariate, the peculiarities of fluorescence microscopy images can be used to accelerate the process and to obtain viable performance.

Due to the fact we already approximately know the colors that represent clusters in the image, all the particles can start their search in regions of the search space corresponding to the estimated set of representative colors. In other words, we can initialize the parameter μ of each Gaussian with the value of the color that it represents and set the other parameters to random values. So, part of the work that PSO would do if all parameter start with random values, is already done, once it does not have to find the principal colors of the image. So this is the main adaptation included in the method.

The PSO searches for a vector v that represents the parameter of the GMM that better represent the images clusters. With 3-dimension data, each Gaussian will need 3 values of the vector to represent μ, 3 to represent the variances $diag(\Sigma)$ e 1 to the weight of the Gaussian. To provide only valid covariance matrices and reduce the size of v, we assume all the values out of the diagonal of Σ as zero. In other words, the adopted model ignores the dependence between the variables, just like the Naïve Bayes classification.

So the size of the vector is $|v| = 7K-1$, where K is the number of Gaussian component of the GMM or the number of colors of the image. Once the sum of all weight must be 1, they are redundant, so one of them needs not to be sought.

4 Experiments with Segmentation

The images used in this experiment were taken from [15] and are either under the terms of GNU Free Documentation License, Version 1.2 or public domain. The first is a 2-class image color separated from the original with 4 color classes, as seen in Fig 1 and 2. The third image has also 4 color classes and is presented in Fig 3.

The segmentation of the five images was accomplished by the proposed technique. In all segmentations the proposed process was successful in identifying the pixel color classes and classifying the pixels. Images with the same number of colors like Fig 2 to 3 had an expressive variation of processing time. However, the total time spent was relative low considering the 3 dimensions of data. All segmentations considered all dimensions of data. The 4-colored images took longer to process than the 2-color images, as expected.

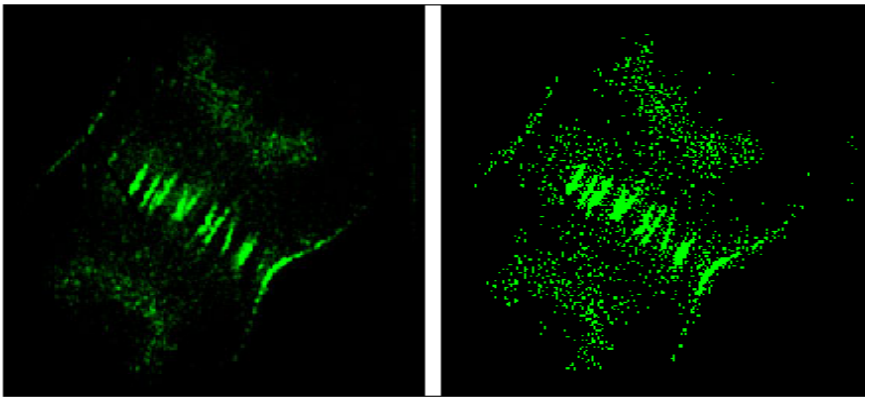


Fig. 1. Original image and segmentation result. Time to process: 0.5 seconds.

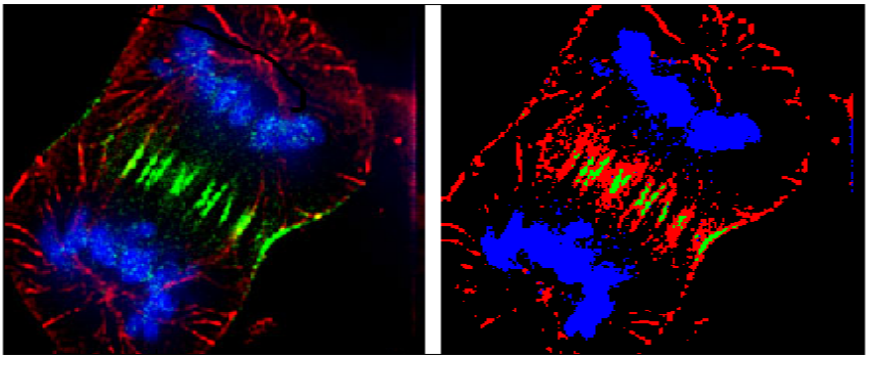


Fig. 2. Original image and segmentation result. Time to process: 19.4 seconds.

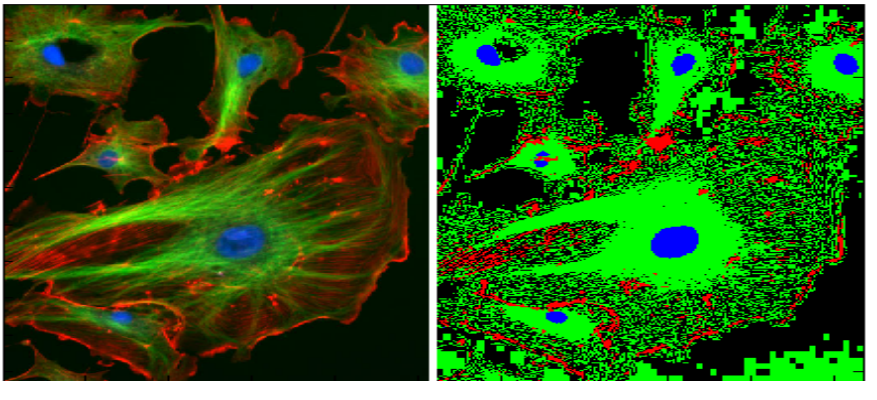


Fig. 3. Original image and segmentation result. Time to process: 111 seconds.

5 Experiments Comparing PSO and EM

We ran experiments to evaluate comparatively the use of the EM and the PSO methods. Both methods were implemented in MATLAB. Due to the stochastic nature of PSO [16], it yields different solutions in successive runs [17] with different time cost, even doing the search with the particles at the same start points. So, in order to evaluate the PSO, a set of 100 executions were conducted and a summary of results is reported in Table 1. The image used in the experiment (Fig 3) has size 512x512.

Similarly to the implemented PSO, the estimated cluster centers were also used to initialize the start point of EM executions, using random values just for $diag(\Sigma)$ and $\pi_1 \ldots \pi_k$. The dependence between the variables also was ignored in EM executions.

The PSO parameters used were $cb=cg=2$ and w decreasing linearly from 0.9 to 0.4 as suggest in [18]. The color value of image pixels varies between [0,1], so we used $Vmax=1$. In each PSO run, 24 particles were used and the max number of iterations was 250. The histogram used by the fitness function was divided in 27 bins distributed in the 3 dimension of data.

As for the evaluated implementations, EM stops when the variation of the likelihood is below 0.001 beetween 2 iterations, while PSO stops when the variation of the quadratic error is below 0.001 within 30 iterations or the max number of iterations is reached. Results of the executions are shown in Table 1:

Table 1. Results of 100 runs of PSO and EM over the image of fig. 3

	Time (s) PSO	Time (s) EM	Error PSO	Error EM	Iterations PSO	Iterations EM
Min	0.1916	2.7407	0.0020	0.0061	30	12
Max	2.9254	6.9157	0.0872	0.0147	250	31
Median	2.0155	5.7843	0.0082	0.0145	219	26
Mean	1.9830	5.6106	0.0104	0.0125	215.73	25.08
Std. Dev.	0.4065	0.7338	0.0124	0.0030	36.03	3.34

6 Conclusion and Future Works

The experiments showed the use of PSO to estimate GMM in more than one dimension data. More specifically, we applied GMM estimation to model and classify pixels of colored images of fluorescence microscopy to perform image segmentation. The segmentation of fluorescence microscopy images were shown to be an ideal problem for this purpose due to its particular characteristics that could be used to enhance search performance. We also compared PSO with EM in terms of time execution and accuracy.

The obtained results showed good performance of the proposed technique for segmentation. The good overall performance of the search algorithm can be credited to the initial assignment to the start position of the particles with the known color classes of the figure. Additionally, the possibility allowed by PSO of choosing an efficient

fitness function and adjusting parameter values during the execution provided means to reach good performance. However, the computational cost is not easily predictable. This is a known disadvantage of the PSO. Comparison to other heuristic techniques regarding performance should be made in future work.

For the next few steps of this research we propose considering the application of the method to more difficult problems, like images with undefined number of colors, data with more than 3 dimensions and non-Gaussian datasets and to compare its performance and robustness with the expectation maximization algorithm, which is the traditional method to estimate parameters of GMMs. This involves tackling the problem of generating only valid covariance matrices to allow modeling intervariable dependence.

Acknowledgments. This work is partially funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

References

1. Bailey, T.L., Elkan, C.: Fitting a Mixture Model by Expectation Maximization to Discover Motifs in Bipolymers (1994)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1–38 (1977)
3. Fan, S.-K.S., Zahara, E.: A hybrid simplex search and particle swarm optimization for unconstrained optimization. European Journal of Operational Research 181, 527–548 (2007)
4. Zhang, Y., Wu, L.: A Hybrid TS-PSO Optimization Algorithm. Journal of Convergence Information Technology 6, 169–174 (2011)
5. Zheng, Y.-L., Ma, L.-H., Zhang, L.-Y., Qian, J.-X.: On the convergence analysis and parameter selection in particle swarm optimization (presented at the November 2003)
6. Zahara, E., Fan, S.-K.S., Tsai, D.-M.: Optimal multi-thresholding using a hybrid optimization approach. Pattern Recognition Letters 26, 1082–1095 (2005)
7. Fan, S.-K.S., Lin, Y., Wu, C.-C.: Image thresholding using a novel estimation method in generalized Gaussian distribution mixture modeling. Neurocomputing 72, 500–512 (2008)
8. Gonzalez, R.C., Woods, R.E.: Processamento de imagens digitais. Editora Edgard Bl cher (2000)
9. Bengtsson, E., Wahlby, C., Lindblad, J.: Robust cell image segmentation methods. Pattern Recognition Letters 14, 157–167 (2004)
10. Kennedy, J., Eberhart, R.: Particle swarm optimization (presented at the December 1995)
11. Hassan, R., Cohanim, B., De Weck, O., Venter, G.: A comparison of particle swarm optimization and the genetic algorithm (presented at the 2005).
12. Garlapati, V., Vundavilli, P., Banerjee, R.: Evaluation of Lipase Production by Genetic Algorithm and Particle Swarm Optimization and Their Comparative Study. Applied Biochemistry and Biotechnology 162, 1350–1361 (2010)
13. Doctor, S., Venayagamoorthy, G.K., Gudise, V.G.: Optimal PSO for collective robotic search applications. In: Proceedings of the 2004 Congress on Evolutionary Computation, pp. 1390–1395. IEEE (2004)
14. McLachlan, G.J., Peel, D.: Finite Mixture Models. John Wiley and Sons (2000)

15. Wikipedia contributors (F Lamiot): Fluorescence microscope (2012), http://en.wikipedia.org/w/index.php?title=Fluorescence_microscope&oldid=486990589
16. Fernandez-Martinez, J.L., Garcia-Gonzalo, E.: Stochastic Stability Analysis of the Linear Continuous and Discrete PSO Models. IEEE Transactions on Evolutionary Computation 15, 405–423 (2011)
17. Bharat, T.V., Sivapullaiah, P.V., Allam, M.M.: Robust solver based on modified particle swarm optimization for improved solution of diffusion transport through containment facilities. Expert Systems with Applications 39, 10812–10820 (2012)
18. Shi, Y., Eberhart, R.: Parameter selection in particle swarm optimization. (presented at the 1998)

Comparison between MLP and LVQ Neural Networks for Virtual Upper Limb Prosthesis Control

Daniel Caetano, Fernando Mattioli, Kenedy Nogueira, Edgard Lamounier, and Alexandre Cardoso

Federal University of Uberlandia,
Av. João Naves de Avila, 2121, Uberlandia, Brazil
{sdc.daniel,mattioli.fernando,prof.kenedy}@gmail.com,
{lamounier,alexandre}@ufu.br
http://www.grva.eletrica.ufu.br

Abstract. During the rehabilitation process, individuals who have experienced a total or partial loss of upper limbs are exposed to many risks. Besides this, a great mental effort is required during the training phase to adapt to a real prosthesis. In many cases, the use of Virtual Reality in Medicine has proven to be an excellent tool for evaluation and support as well as to mitigate risk and to reduce mental effort required. In order to be useful, virtual prosthesis must have a great similarity with the real world. For this reason, artificial neural networks have been explored to be applied in the training phase to provide real time response. The objective of this study is to compare the performance of the LVQ and MLP neural networks in EMG (muscle activity) pattern recognition. To achieve this, different feature extraction techniques for simulation and control of virtual prostheses are investigated.

Keywords: Virtual reality, Neural networks, Rehabilitation, EMG pattern recognition, Feature extraction.

1 Introduction

A prosthesis is a device that aims to recover an amputated limb function. The electromyographic signal (EMG), collected in the remaining muscles of the amputated limb, can be used to control myoelectric prosthesis. The EMG signal is an electric potential produced by a particular muscle contraction. By processing the EMG signal, it is possible to discriminate different upper limb movements. This application has become an important human-machine interface in many areas, such as prosthesis control (on-off and proportional), robotic hands control, and Force Display Devices (FDD) control in Virtual Reality (VR) environments [1]. Due to the its stochastic nature, pre-processing techniques (section 2.2 and 2.3) capable of extracting EMG signal information are required before signal classification.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 382–389, 2012.

Artificial neural network (ANN) are systems that can recognize and classify patterns, such as EMG, from a learning model based on human learning [2]. A striking feature of ANN is its capability of generalization, after a training stage in which some input patterns are presented and processed by the network. In the execution stage, different patterns from those used in training stage may be processed properly by the network.

The use of VR techniques by myoelectric prosthesis in their training stage presents itself as a complementary tool that favors adaptation to artificial limbs [3]. VR techniques also enable performance evaluation of different control systems, ease wear during the training, and provide a good visual feedback [4]. Many authors have investigated the use of EMG signal in upper limb and prosthesis control: Huang et. al [5], Sebelius et. al [4] and Pons et. al [6]. They discussed the question around hands prosthesis control whilst Herle et. al [3], Nogueira et. al [7] and Soares et. al [8] treated the virtual arm control. EMG signal classification, pattern recognition, feature extraction, real-time signal processing, and realistic prosthesis simulation are among the main challenges faced by these authors.

This paper presents a comparative study of two classifiers using different feature extraction techniques. The main objective is to provide, during training stage, a better prosthesis control to the individuals who have experienced upper limbs lost.

2 Materials and Methods

Feature extraction techniques presented in section 2.3 will be applied in each movement database. The database consists of hand movement investigated by Mattioli et. al [9] and arm movement, investigated by Soares et. al [8] and Nogueira et. al [7]. Five replicates were used for each movement: arm (isometric / isotonic contraction) and hand (isometric contraction). This movements will be used to generate the basics training patterns that will feed the LVQ and MLP neural networks. All patterns will be used during the training phase and application, since it is a virtual prosthesis for a single patient.

Configuration parameters of each neural network, such as: learning rate, learning rate decrease, tolerance, number of outputs units, number of hidden layer neurons and momentum, will be variated. In order to evaluate each network classification performance, efficiency (Equation 1) and training time will be computed for each different configuration.

$$E = 100 \times \frac{N_{correct}}{N_{total}}\% \tag{1}$$

The desktop configuration used to run these tests is:

- Operational system: Ubuntu Linux, 10.04(kernel)2.6.32;
- RAM 2Gb;
- Intel(R) Core^{TM} 2 Quad E4700, 2.6GHz Processor;

2.1 System Architecture

Figure 1 shows the structure proposed in the following, all process stages will be detailed.

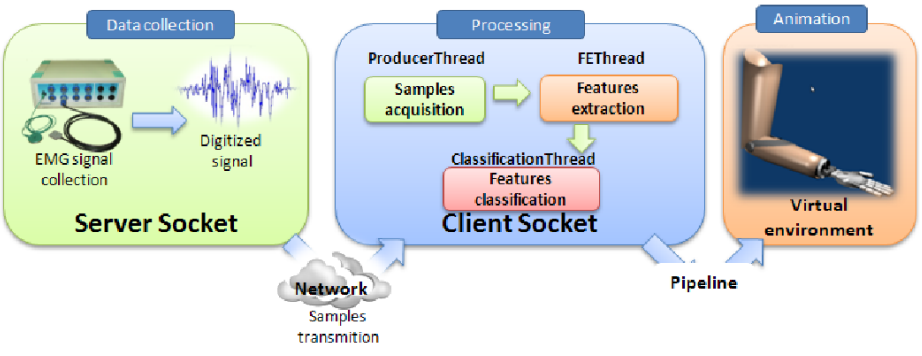


Fig. 1. System architecture

The collection and transmission of samples are performed in "Data collection" stage, which are sent by socket to another block responsible by the "Processing".

All tasks related with samples processing are detailed in sections 2.2, 2.3 and 2.4.

2.2 Signal Windowing

The first phase to be executed in the "Processing" stage is "Samples acquisition". Teager's energy operators (TEO), a real-time boundary detector method created by Peretta [10] and used by Mattioli et. al [11], will be applied to all samples received to extract only the significant parts of the signal. Once the relevant signal extracted it is divided into segments of each samples, similar to the procedure realized by Herle et. al [3].

2.3 Feature Extraction

In order to reduce the amount of information to be presented to neural networks, two techniques of features extraction for each segment are presented: Time-domain features (TDF), used by Herle et. al [3], and Hudgins et. al [12]; and autoregressive model (ARM), studied by Soares et. al [8].

1. TDF: Five features were defined: Mean Absolute Value (MAV), Mean Absolute Value Slope (MAVS), Zero Crossing (ZC), Slope Sign Changes (SSC) and Waveform Length (WL) [12] [3], all of those are calculated within each segment of 40 samples.

2. ARM: This is a representation of a specific signal which depends solely on the output values previously stored by the system. The $\hat{y}(n)$ variable value in a specific time of a ARM may be estimated from some previous variable values $(y(n-1), y(n-2), ...)$. An ARM is defined by Equation (2) [8] and it is applied to each segment of 40 samples.

$$\hat{y}(n) = \sum_{m=1}^{M} a_m(n)y(n-m) + e(n) \tag{2}$$

Where: $\hat{y}$ is a estimated value at time n; a_m is autoregressive (AR) coefficient of order m calculated for each samples within each segment; $e(n)$ is an estimated error; and M the order of ARM which determines the number of coefficients a_m.

2.4 Classification Technique

The feature vectors will be present to neural networks as an input. If TDF technique is used, only five features will be presented at a time for the network for each segment. Moreover, if ARM is used each sample will be represented by M numbers of AR coefficients, i.e., $40 \times M$ coefficients for each segment (of 40 samples) will be presented for the network.

After presenting the feature vector to each of the chosen networks, there's the classification of which movement that feature vector represents.

Two networks — LVQ (Learning Vector Quantization) and MLP (Multi Layer Percetron) — with one hidden layer were chosen, considering that the work performed by Soares et. al [8] and Mattioli et. al [11] achieved a satisfactory performance.

Details on the classification techniques used are described by Mattioli et. al [9] [11].

2.5 Training Environment Prototype

The GUI of the training environment, is showed in Figure 2, in which the network training settings can be adjust.

A virtual prosthesis model was developed using 3Dstudio Max® [13] and after exported to Blende3D^{TM}. The GUI and 3D model are initiated at the same time. After three feature vectors correct classification by the network, a message is sent by pipeline to Blender that triggers an animation of the virtual upper limb prosthesis: The virtual prosthesis movements are: hand movements (extension, flexion, grasping and forearm pronation) and arm movements (elbow flexion, extension and forearm pronation, supination).

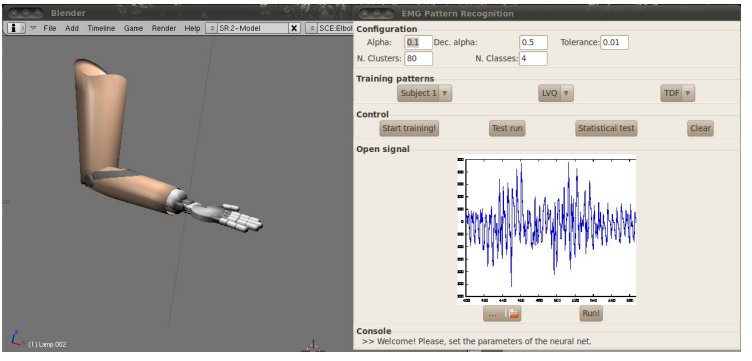


Fig. 2. GUI of the training environment

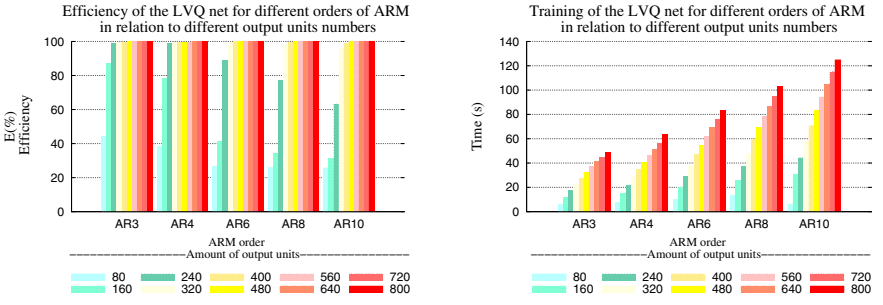
2.6 Testing Methodology

A single parameter will be variated at time, in order to understand its impact on results. The range of each parameter is described below:

1. **Standard settings:**
   – **LVQ Network:** learning rate=0.1, reduce learning rate=0.5, and tolerance=0.001;
   – **MLP Network:** learning rate = 0.1, tolerance=150, momentum 0.5 and number of neurons in hidden layer = 20;
2. **Variable settings:**
   – **LVQ Network:**
   –– **Outputs units:** The maximum percentage of output units will be up to 90% of the total (TDF-hand and arm) patterns, 22% of the total (ARM-hand) patterns and 11% (ARM-arm) patterns, learning rate and reduce learning rate 0.01 to 0.99 and tolerance is 0.001 to 0.099;
   –– The number of repetitions for each test parameter is changed is: — TDF:, 100 repetitions; —ARM: ($3^{rd}, 4^{th}, 6^{th}, 8^{th}$ and 10^{th} order), 25 repetitions.
   – **MLP Network:**
   –– Learning rate: 0.1 to 0.7; tolerance: 100 to 300; number of neurons in hidden layer: 1 to 30;
   –– - 100 repetitions will be performed for each parameter variated, independent of the used feature extraction technique;

3 Discussion

From the LVQ network performed tests follows that: Using TDF for hand movements has reached 97% of efficiency with only 72% of the training patterns. Using TDF for arm movements has reached a maximum of 80% of efficiency. Using the

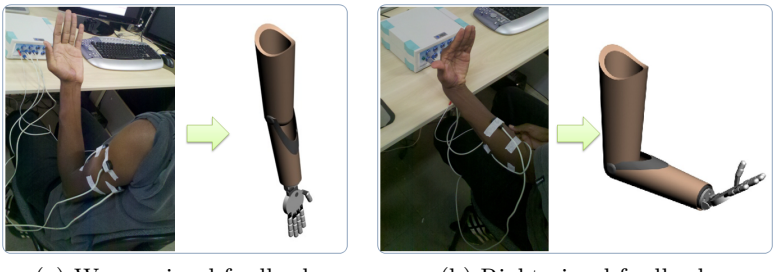


(a) Efficiency (b) Training time

Fig. 3. LVQ network tests for isometric contractions of hand movement using ARM

ARM for hand movements has reached 99% efficiency with only 11% of the training patterns as shown in Figure 3, and 97% efficiency for arm movements with only 10% of all training patterns.

The margin error of 20% for the arm movements increases the probability of error in the classification of the movement performed. Figure 4a illustrates a unacceptable feedback for a virtual training environment. Figure 4b shows the correct virtual movement due to high network efficiency.



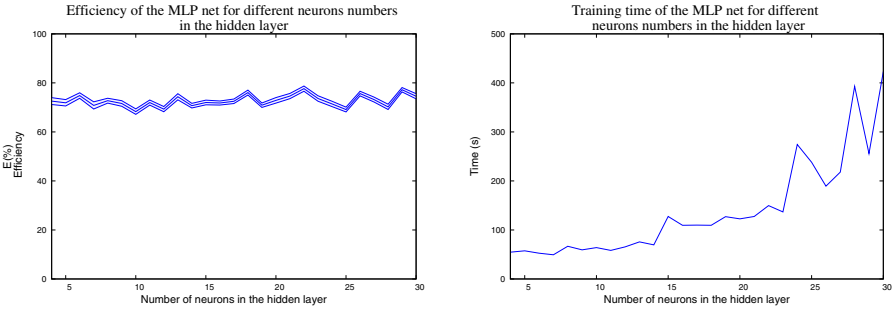
(a) Wrong visual feedback (b) Right visual feedback

Fig. 4. Neural network classification examples

The training time using TDF in both cases did not reach a second. The training time for an 3^{rd} order ARM was 27 seconds.

In tests with MLP network performance rating above 85% was not observed for hand movements, and the average training time was 100 seconds, as shown in Figure 5.

Results as the ones obtained in the MLP network to hand motions using TDF, and results for LVQ using ARM arm movements, are not suitable for controlling virtual prosthesis, since the error rate is too large.



(a) Efficiency (b) Training time

Fig. 5. MLP network tests for isometric contractions of hand movement using TDF

4 Conclusions and Future Work

Machover [14] states that VR systems need to provide a consistent reaction to the user's movements, making the experience consistent. This emphasizes the importance of studying methods and techniques for increasing the efficiency of pattern recognition techniques, in order to have a correct classification of movements performed by the patient.

The results presented assert that the LVQ using ARM is a good alternative for controlling virtual prosthesis for upper limbs (arm and hand movements). This is a bright spot of this work, considering that all previous work tried to simulate only one movement at a time. In Soares et. al [8] it was necessary 50% of the training patterns to achieve the efficiency of 100%; and this work achieved the same only 10% of the training pattern.

Neural networks are often used for pattern recognition of EMG signals. Their efficiency depends on the used pre-processing technique and the way the signal is captured.

As a future work, the authors will perform a new data capture of hand movements signal with a greater number of capture channels and repeat the tests in MLP network using TDF.

References

1. Nishikawa, D., Yu, W., Yokoi, H., Kakazu, Y.: On-Line Learning Method for EMG Prosthetic Hand Control. Electronics and Communications in Japan, 1510–1519 (2001)
2. Fausett, L. (ed.): Fundamentals of neural networks: architectures, algorithms, and applications. Prentice-Hall, Inc., Upper Saddle River (1994)
3. Herle, S., Raica, P., Lazea, G., Robotin, R., Marcu, C., Tamas, L.: Classification of surface electromyographic signals for control of upper limb virtual prosthesis using time-domain features. In: Proceedings of the 2008 IEEE International Conference on Automation, Quality and Testing, Robotics, pp. 160–165 (2008)

4. Sebelius, F., Axelsson, M., Danielsen, N., Schouenborg, J., Laurell, T.: Real-time control of a virtual hand. Technology and Disability 17(3), 131–141 (2005)
5. Huang, H., Liu, Y., Liu, L., Wong, C.: Emg classification for prehensile postures using cascaded architecture of neural networks with self-organizing maps. In: Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2003, vol. 1, pp. 1497–1502. IEEE (2003)
6. Pons, J., Ceres, R., Rocon, E., Levin, S., Markovitz, I., Saro, B., Reynaerts, D., Van Moorleghem, W., Bueno, L.: Virtual reality training and EMG control of the MANUS hand prosthesis. Robotica 23(03), 311–317 (2005)
7. Nogueira, K.L.: O Uso de Técnicas de Realidade Virtual e Aumentada na Simulação de Prótese de Membros Superiores. Master's thesis, Universidade Federal de Uberlndia, Uberlndia (Julho 2007)
8. Soares, A., Andrade, A., Lamounier, E., Carrijo, R.: The development of a virtual myoelectric prosthesis controlled by an EMG pattern recognition system based on neural networks. Journal of Intelligent Information Systems 21(2), 127–141 (2003)
9. Mattioli, F., et al.: Utilização de redes neurais para a classificação de sinais emg aplicados no controle de próteses virtuais de mão. In: Proceedings of the WRVA 2010, pp. 230–235 (2010)
10. Peretta, I.: A novel Word Boundary Detector based on the Teager Energy Operator for Automatic Speech Recognition. PhD thesis, Master's Thesis, Federal University of Uberlândia, UFU (2010)
11. Mattioli, F., Lamounier, E., Cardoso, A., Soares, A., Andrade, A.: Classification of emg signals using artificial neural networks for virtual hand prosthesis control. In: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, pp. 7254–7257. IEEE (2011)
12. Hudgins, B., Parker, P., Scott, R.N.: A new strategy for multifunction myoelectric control. IEEE Transactions on Biomedical Engineering 40(1), 82–94 (1993)
13. Lamounier, E.A., Lopes, K., Soares, A.B.: Using augmented reality techniques to simulate myoelectric upper limb prostheses. Bioengineering & Biomedical Science, S1:010 (2012), doi:10.4172/2155-9538.S1-010
14. Machover, C., Tice, S.: Virtual reality. IEEE Computer Graphics and Applications 14(1), 15–16 (1994)

A Constructive Particle Swarm Algorithm for Fuzzy Clustering

Alexandre Szabo[1], Leandro Nunes de Castro[1], and Myriam Regattieri Delgado[2]

[1] Natural Computing Laboratory,
Mackenzie University, São Paulo, Brazil
alexandreszabo@gmail.com,
lnunes@mackenzie.br
[2] Federal University of Technology of Paraná,
Curitiba, Brazil
myriamdelg@utfpr.edu.br

Abstract. This paper proposes a fuzzy version of the crisp cPSC (*Constructive Particle Swarm Clustering*), called FcPSC (*Fuzzy Constructive Particle Swarm Clustering*). In addition to detecting fuzzy clusters, the proposed algorithm dynamically determines a suitable number of clusters in the datasets without the need of prior knowledge, necessary in cPSC to control the number of particles in the swarm. The FcPSC algorithm was applied to six databases from the literature and its performance was compared with that of *Fuzzy C-Means*, a *Fuzzy Artificial Immune Network*, a *Fuzzy Particle Swarm Clustering* and the crisp cPSC. FcPSC showed to be competitive with the algorithms used for comparison and the number of particles generated was smaller than for cPSC.

Keywords: Fuzzy Clustering, Particle Swarm, Artificial Immune System, Dynamic Population, Bioinspired Algorithms.

1 Introduction

Cluster Analysis refers to a set of techniques used to segment objects (patterns) into clusters according to their similarities. It can be applied to several areas, such as image analysis [1][2] and information retrieval [3][4]. Clustering is considered one of the most important tasks in data mining [5-7], because it allows the organization and summarization of large sets of data, the discovery of classes and the description of their features.

One typical form of clustering is based on prototypes, i.e., one or more specific vectors that are used to represent a whole cluster of data. This type of representation leads to a significant reduction in the dataset size, because the number of prototypes is usually much smaller than the number of objects in the dataset [7]. This is a type of vector quantization that reduces the processing cost, thus facilitating the knowledge extraction from the dataset. At the end of the clustering process, the prototypes must be positioned in regions of the space that suitably represent the input data. Generally,

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 390–398, 2012.

the number of prototypes must be informed *a priori*. Some algorithms for which the number of prototypes is dynamically determined can be seen in [8-10].

Clustering algorithms can be classified as *crisp* or *fuzzy* based on the method used to assign objects to groups [7]. In the crisp algorithms [11-13] each object from the dataset belongs to a single group, whilst in fuzzy clustering [14-17] an object may belong to more than one cluster simultaneously, but with a different membership degree to each group.

The idea of using fuzzy sets concepts in cluster analysis was proposed by Bellman et al. [18], who discussed its relevance in pattern recognition in the context of communication and control theory. However, Ruspini [19] is considered the first to introduce fuzzy set concepts in cluster analysis, although he has used an approach more probabilistic than fuzzy to assign membership degrees to objects [20].

The fuzzy-based approach has some advantages in the cluster analysis context, such as to provide the relationship between objects and clusters found and to quantify their proximity. This is done by analyzing the membership degrees generated by the fuzzy clustering algorithms. Another advantage is the ability to manipulate imprecise information. Thus, using *high* dissimilarity or *low* dissimilarity between object and cluster can be interesting to decide to which group an object may belong to.

This paper proposes an extension of the cPSC algorithm [9], called FcPSC, to determine fuzzy partitions of a given dataset. The cPSC is a bioinspired algorithm that uses swarm and immune system concepts to dynamically determine a suitable number of clusters in a dataset. The cPSC uses an affinity threshold (ε) in the *swarm growing* step, which is empirically obtained for each dataset to be grouped. The proposed algorithm, FcPSC, replaces the affinity threshold by a fixed and unique value for any dataset, which is represented by a membership degree threshold to quantify the dissimilarity between particle and object.

2 Fuzzy Clustering: Basic Concepts

Unlike crisp data clustering algorithms, the fuzzy approach assigns each object from the database to all clusters simultaneously by varying the membership degree between objects and clusters. Fuzzy clustering algorithms use a membership matrix **U**, consisting of n lines and c columns, where n is the number of objects to be grouped and c the number of clusters. The membership matrix is used to guide the clustering process. Each element in **U** provides the membership degree of an object $\mathbf{y}^i$ to a cluster $\mathbf{c}_j$: the closer an object to a cluster, the higher its membership degree to that cluster, and vice-versa. The membership degree of an object $\mathbf{y}^i$ to a cluster $\mathbf{c}_j$ can be represented by μ_{ij} ($\mu_{ij} \in \mathbf{U}$), $\forall i = 1,\ldots, n$ e $\forall j = 1,..., c$ and indicates how representative (pertinent) the object is to this cluster in relation to the other ones [21]. The membership matrix **U** is iteratively updated by using, for example, the equation below [22]:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{ik}}\right)^{\frac{2}{m-1}}}. \quad (1)$$

where μ_{ij} is the membership degree of $\mathbf{y}^i$ in relation to $\mathbf{c}_j$; d_{ij} represents the distance (usually Euclidean) between $\mathbf{y}^i$ and $\mathbf{c}_j$; and d_{ik} is the distance between $\mathbf{y}^i$ and $\mathbf{c}_k$. The *fuzzification parameter* (m), known as *weighting exponent*, indicates the width of the hypersphere (circumference in bidimensional case) of the group, and it must be in the range (1, ∞), being commonly found in the interval [1.25, 2] [21]. Generally, fuzzy clustering algorithms use some objective function, which should be minimized throughout the iterations. The most commonly used function is:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} \mu_{ij}^m \, {d_{ij}}^2. \quad (2)$$

3 FcPSC: The Proposal of a Fuzzy Constructive Particle Swarm Clustering Algorithm

The cPSC algorithm uses concepts from the immune system to dynamically determine the number of particles in the swarm. The *growing* step of the swarm is evaluated every two iterations and is based on the response of B-cells to antigenic attacks, according to the *Clonal Selection Principle* [23]. If the affinity between the particle (B-cell that recognizes a higher number of antigens) and the presented object (antigen for which the B-cell had higher affinity with in two iterations) is greater than the affinity threshold (ε), then this particle (B-cell) is considered sufficiently stimulated to be duplicated (cloned).

How should the affinity threshold ε be defined so as to trigger the clonal process? If the number of particles added in the swarm is very different from the number of classes in the dataset, the algorithm effectiveness may decrease. In the cPSC, ε is empirically defined for each dataset and has a significant influence in guiding the *growth of the swarm*. We could empirically define ($\varepsilon = 10^{-3}$), i.e., particle and object have *high* affinity if their distance is less than or equal to 10^{-3}. Such threshold can be reasonable in clustering analysis, but affinity indeed depends on the arrangements of objects within the dataset to be grouped. The membership of the objects to groups can be useful to quantify their similarity and, thus, eliminate the need of prior knowledge about the dataset in order to define an affinity threshold in the *growing step* of the cPSC algorithm.

This section presents an extension of the cPSC to be applied to fuzzy partitions, called FcPSC. In the FcPSC, the affinity threshold ε was replaced by an empirical membership value equals to 0.7, which represents a *high* affinity between a particle and an object. Thus, the membership value is always the same for any dataset. The Pseudocode of the FcPSC is presented as follows.

Pseudocode 1: FcPSC: Fuzzy Constructive Particle Swarm Clustering

```
1.   Procedure [X, U] = FCPSC(Y, v_max, ω, m, 0.7)
2.   // Y: input dataset
3.   initialize x//particles' position
4.   initialize v//particles' velocity
5.   initialize U//membership matrix
```

```
6.   t=1
7.   while stopping criterion is not met
8.      for i = 1 to n //each object
9.         μ_ij = Arg_j max(U_i) ∀j = 1, ..., c
10.        if μ_ij > pμ_ij
11.           p_j^i = x_j
12.           pμ_ij = μ_ij
13.        end if
14.        if μ_ij > gμ^i
15.           g^i = x_j
16.           gμ^i = μ_ij
17.        end if
18.        v_j(t+1) = ω * v_j(t) + φ_1 ⊗ (p_j^i(t) − x_j(t)) + φ_2 ⊗ (g^i(t) − x_j(t)) + φ_3 ⊗ (y^i − x_j(t))
19.        v_j ∈ [−v_max, v_max]
20.        x_j(t+1) = x_j(t) + v_j(t+1)
21.        x_j ∈ [0,1]
22.        Update the membership function: Eq.(1)
23.     end for
24.     if mod(t,2)==0
25.        Eliminate particles from the swarm if necessary
26.        Test the stopping criterion: Eq.(3)
27.        Clone particles if necessary
28.     end if
29.     t = t + 1
30.     ω = 0.95*ω
31.  end while
32.  end Procedure
```

FcPSC receives, as input values, a dataset to be grouped ($\mathbf{Y}$), the parameter v_{max} responsible for controlling the winner particle's velocity, the inertia moment (ω) that acts over the winner particle's velocity memory, the fuzzifier parameter m and the fixed membership value 0.7 for controlling the growth of the swarm. Line 3 randomly initializes one particle in the Euclidean vector space [0,1], as well as its velocity in Line 4. Line 5 initializes the membership matrix ($\mathbf{U}$) using Eq. (1). $\mathbf{p}_j^i$ is the vector containing the best position of particle $\mathbf{x}_j$ in relation to the input pattern $\mathbf{y}^i$, and $\mathbf{p}\mu_j^i$ is its respective membership degree; and $\mathbf{g}_i$ is the vector containing the best position of all particles in relation to the input pattern $\mathbf{y}^i$ and $\mathbf{g}\mu_i$ its respective membership degree. Line 9 determines particle $\mathbf{x}_j$ (*winner* particle) with highest membership degree to the input pattern $\mathbf{y}^i$. This membership value is compared with the one that represents the best particle's position in relation to object $\mathbf{y}^i$ so far (Line 10). The same occurs to the particle closest to object $\mathbf{y}^i$ so far (Line 14). At every two iterations the algorithm evaluates the necessity to *eliminate* particles (Line 25), *stop* the algorithm (Line 26) and *clone* particles (Line 27). Line 27 evaluates the necessity to clone a particle and here the affinity threshold (0.7) is used. The particle that most *won* in two iterations is selected as candidate to be cloned. If the membership degree of the object with higher membership degree to this particle in two iterations is greater than 0.7, then a particle is generated and positioned in the middle point between its origin and the object.

4 Performance Assessment

To evaluate the performance of the proposed algorithm, the FCM [17], FPSC [15], FaiNet [16], FcPSC and cPSC [9] algorithms were implemented in Matlab® and applied to six datasets from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets.html): Ecoli, Iris, Pima Indians Diabetes, Yeast, Ruspini and Glass Identification. For the FCM and FPSC algorithms, the number of prototypes used was equal to the number of classes present in the respective databases. The FaiNet, FcPSC and cPSC algorithms start with one particle for every dataset. For all fuzzy algorithms, the fuzzyfication parameter (m) is equal to two. The parametric settings used in the FPSC, FcPSC and cPSC algorithms are described as follows. Parameters $\boldsymbol{\varphi}_1$, $\boldsymbol{\varphi}_2$ and $\boldsymbol{\varphi}_3$ were randomly chosen in the interval (0,1). The *inertia moment* (ω) has an initial value of 0.90, with an iterative decay of 95% at each iteration until the value 0.01 is obtained. The position and velocity of the particles are also controlled, ranging from [0,1] to [–0.1,0.1], respectively [15]. In FaiNet, the values of σ_d (*death threshold*) and σ_s (*suppression threshold*) are set equal to 0.5 [16]. The stopping criterion of the fuzzy algorithms is either a maximum of 200 iterations or a minimal variation in the cost J_m between two consecutive iterations, where J_m is given by Eq. (2):

$$|J_m(t)|-|J_m(t+1)| \leqslant 10^{-3}. \tag{3}$$

The stopping criterion of the crisp cPSC is given by a stabilization of the trajectory of particles or 200 iterations.

In this paper it was used the Entropy (E) and Purity (P) measures to evaluate the performance of the algorithms [24], and the Partition Coefficient (PC) and Partition Entropy Coefficient (PE) validity indices [25-28] to evaluate the fuzzy separation and fuzzy compactness between clusters, respectively. Besides, the Number of Iterations (I) required for convergence and the number of prototypes generated (NP) were also accounted for. The results are shown in Table 1 and the best absolute values of purity are highlighted. It was decided to stress the purity values because the homogeneity of the clusters found is the most important feature to be stressed.

As shown in Table 1, cPSC presented the best Purity and Entropy values for all datasets, when compared with the other algorithms. However, the number of prototypes and iterations for convergence are substantially greater than the remainder algorithms. All constructive algorithms (FaiNet, FcPSC and cPSC) generated at least 50% more prototypes than the number of existing classes, except FaiNet and FcPSC for the Ruspini dataset. For the Diabetes dataset, all algorithms were capable of finding the best solution possible. FcPSC was slower than the other algorithms on average, with the exception of cPSC, and presented better results of purity for the Ecoli and Glass datasets in relation to FCM, FaiNet and FPSC. The FcPSC algorithm presented the highest overlap for all datasets, except for the Diabetes dataset. Based on the PE values, FCM and FPSC presented hardest clusters than the FaiNet and FcPSC algorithms for all datasets.

Table 1. Mean ± standard deviation of the Entropy (*E*), Purity (*P*), Number of Iterations (*I*), Partition Coefficient (*PC*), Partition Entropy (*PE*), and Number of Prototypes (*NP*) for the FCM, FaiNet, FPSC, FcPSC and cPSC algorithms

Dataset	Measure	FCM	FaiNet	FPSC	FcPSC	cPSC
Ecoli	E	0.32±0.0	0.33±0.03	0.35±0.02	0.24±0.02	0.18±0.01
	P	0.79±0.0	0.77±0.03	0.77±0.01	0.83±0.02	0.89±0.01
	I	27.20±10.24	7.10±2.23	140.10±79.70	174.0±47.18	200.0±0.0
	PC	0.43±0.0	0.16±0.03	0.48±0.03	0.13±0.01	-
	PE	0.03±0.0	0.09±0.03	0.03±0.0	0.12±0.01	-
	NP	5.0±0.0	10.40±1.90	5.0±0.0	11.80±0.92	26.30±2.98
Iris	E	0.27±0.01	0.24±0.04	0.26±0.02	0.25±0.02	0.07±0.01
	P	0.89±0.0	0.89±0.03	0.89±0.01	0.88±0.02	0.96±0.01
	I	11.50±1.96	4.90±3.11	137.20±69.50	172.20±54.15	200.0±0.0
	PC	0.74±0.0	0.38±0.06	0.73±0.01	0.24±0.03	-
	PE	0.0±0.0	0.05±0.02	0.0±0.0	0.10±0.02	-
	NP	3.0±0.0	5.70±0.48	3.0±0.0	6.50±0.85	21.50±3.57
Diabetes	E	0.0±0.0	0.02±0.03	0.0±0.0	0.0±0.0	0.0±0.0
	P	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0
	I	10.10±1.52	7.10±2.69	187.20±40.48	200.0±0.0	200.0±0.0
	PC	0.79±0.0	0.13±0.03	0.79±0.01	0.25±0.02	-
	PE	0.0±0.0	0.05±0.02	0.0±0.0	0.02±0.0	-
	NP	2.0±0.0	12.20±2.53	2.0±0.0	5.30±0.48	8.60±4.11
Yeast	E	0.15±0.0	0.11±0.06	0.05±0.04	0.13±0.03	0.03±0.01
	P	0.92±0.0	0.96±0.03	0.98±0.02	0.92±0.02	0.99±0.0
	I	23.30±3.27	11.10±14.04	164.20±65.52	191.20±20.40	200.0±0.0
	PC	0.60±0.0	0.20±0.07	0.67±0.02	0.19±0.03	-
	PE	0.01±0.0	0.12±0.05	0.01±0.0	0.11±0.03	-
	NP	4.0±0.0	10.20±2.62	4.0±0.0	8.40±1.71	16.80±4.92
Ruspini	E	0.0±0.0	0.07±0.06	0.05±0.10	0.0±0.0	0.0±0.0
	P	1.0±0.0	0.97±0.03	0.96±0.08	1.0±0.0	1.0±0.0
	I	9.80±1.23	2.40±0.70	33.60±36.59	162.60±61.45	99.20±21.59
	PC	0.86±0.0	0.65±0.07	0.82±0.09	0.38±0.03	-
	PE	0.03±0.0	0.06±0.03	0.04±0.02	0.13±0.03	-
	NP	4.0±0.0	4.20±0.42	4.0±0.0	5.0±0.47	10.50±1.35
Glass	E	0.58±0.01	0.54±0.04	0.56±0.02	0.50±0.03	0.26±0.02
	P	0.55±0.01	0.54±0.02	0.54±0.02	0.58±0.03	0.78±0.02
	I	33.60±6.33	5.30±2.11	147.80±58.47	160.40±64.82	200.0±0.0
	PC	0.43±0.01	0.18±0.06	0.47±0.05	0.18±0.03	-
	PE	0.04±0.01	0.17±0.05	0.04±0.01	0.12±0.04	-
	NP	6.0±0.0	12.30±1.77	6.0±0.0	9.20±1.99	44.50±4.01

A Shapiro-Wilk Test [29] was used to determine whether the behavior presented by the algorithms (Table 1) had a normal distribution, based on the Purity measure. Assuming a confidence level equals to 0.95, the test of normality revealed that the null hypothesis (H_0) should be rejected and, thus, a nonparametric test should be used to assess the statistical significance of performances. To determine if the difference in performance among the evaluated algorithms is significant, we used the Friedman test [30][31], a nonparametric method analogous to the parametric ANOVA (Analysis of Variance) [32]. The Friedman test is based on the ranking of the results obtained for each sample (database) i to all k algorithms. The value of the degrees of freedom is

obtained by k–1, being $k = 5$, so there are 4 degrees of freedom. Thus, according to the χ^2 table [33], the critical value of probability for $\alpha = 5\%$ is 9.49. As the χ^2 calculated (10.20) exceeds the critical value, the null hypothesis H_0 is rejected. In other words, the difference in performance between the algorithms is statistically significant for the databases tested and the crisp cPSC presents best purity values, though the number of particles generated is higher than for the other ones.

5 Conclusions and Future Works

This paper presented a fuzzy version of the crisp cPSC algorithm, which is called FcPSC. In the *swarm growing* step, FcPSC replaces the affinity threshold (ε) by a fixed and unique value for any dataset, which is represented by a membership degree threshold. The proposed algorithm was applied to six datasets from the literature and its performance was compared with its crisp version, cPSC, and a fuzzy particle swarm algorithm (FPSC), besides a classical algorithm from the literature (FCM) and an immunofuzzy algorithm (FaiNet). The results showed that the FcPSC is competitive with the algorithms evaluated.

Further investigations include dynamically detecting the membership degree threshold used in the *swarm growing* step and a parametric sensitive analysis of the cPSC algorithm for the affinity threshold ε, as well as to its fuzzy version. To further decrease the number of prototypes it will be used a *Clonal Suppression Process*, similar to the one applied in the FaiNet algorithm.

Acknowledgment. The authors thank FAPESP, CNPq, Mackenzie University and Mackpesquisa for the financial support.

References

1. Chang-ming, Z., Guo-chang, G., Hai-bo, L., Jing, S., Hualong, Y.: Segmentation of Ultrasound Image Based on Cluster Ensemble. In: IEEE International Symposium on Knowledge Acquisition and Modeling, pp. 418–421 (2008)
2. Bonanno, L., Marino, S., Bramanti, A., Bramanti, P., Lanzafame, P.: Cluster Analysis boosted watershed segmentation of neurological image. In: 4th International Congress on Image and Signal Processing (CISP), pp. 1223–1226 (2011)
3. Chang, Y., Kim, M., Raghavan, V.V.: Construction of query concepts based on feature clustering of documents, vol. 9, pp. 231–248. Springer, Netherlands (2006)
4. Lau, R.Y.K.: Context sensitive text mining and belief revision for adaptive information retrieval. In: IEEE/WIC International Conference on Web Intelligence, pp. 256–262 (2003)
5. Jiang, D., Pei, J., Zhang, A.: An interactive approach to mining gene expression data. IEEE Transactions on Knowledge and Data Engineering 17, 1363–1378 (2005)
6. Yoo, I., Alafaireet, P., Marinov, M., Pena-Hernandez, K., Gopidi, R., Chang, J.-F., Hua, L.: Data Mining in Healthcare and Biomedicine: A Survey of the Literature. J. of Medical Systems, 1--18 (2011)

7. Döring, C., Lesot, M.-J., Kruse, R.: Data analysis with fuzzy clustering methods. Computational Statistics & Data Analysis 51, 192–214 (2006)
8. Omran, M.G.H., Salman, A., Engelbrecht, A.P.: Dynamic Clustering Using Particle Swarm Optimization with Application in Image Segmentation. Pattern Analysis and Applications 8, 332–344 (2006)
9. Prior, A.K.F., de Castro, L.N.: Um algoritmo de Enxame Construtivo para Agrupamento de Dados. In: Congresso Brasileiro de Automática (CBA), pp. 3300–3307 (2010)
10. de Casto, L.N., Von Zuben, F.J.: An evolutionary immune network for data clustering. In: 6th IEEE Brazilian Symposium on Neural Networks, pp. 84–89 (2000)
11. van der Merwe, D.W., Engelbrecht, A.P.: Data clustering using particle swarm optimization. In: IEEE Congress on Evolutionary Computation, pp. 215–220 (2003)
12. Runkler, T.A.: Ant colony optimization of clustering models. International J. of Intelligent Systems 20, 1233–1251 (2005)
13. Cohen, S.C.M., de Castro, L.N.: Data Clustering with Particle Swarms. In: IEEE World Congress on Computational Intelligence, pp. 6256–6262 (2006)
14. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. J. of Cybernetics 3, 32–57 (1973)
15. Szabo, A., de Castro, L.N., Delgado, M.R.: The proposal of a fuzzy clustering algorithm based on particle swarm. In: IEEE 3th World Congress on Nature and Biologically Inspired Computing, pp. 459–465 (2011)
16. Szabo, A., de Castro, L.N., Delgado, M.R.: FaiNet: An Immune Algorithm for Fuzzy Clustering. In: IEEE World Congress on Computational Intelligence (to appear)
17. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
18. Bellman, R.E., Kalaba, R.E., Zadeh, L.A.: Abstraction and Pattern Classification, pp. 1–7. RAND Corporation, Santa Monica (1964)
19. Ruspini, E.H.: A new approach to clustering. Information and Control 15, 22–32 (1969)
20. Davis, R.H., Economou, C.E.: A review of fuzzy clustering methods. Advances in Engineering Software 6, 189–191 (1984)
21. Cox, E.: Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration. Morgan Kaufmann, San Francisco (2005)
22. de Oliveira, J.V., Pedrycz, W.: Advances in Fuzzy Clustering and its Applications. John Wiley & Sons Ltd, England (2007)
23. Burnet, F.M.: The Clonal Selection of Acquired Immunity, pp. 1–209. Cambridge University Press, London (1959)
24. Amigó, E., Gonzalo, J., Artiles, J.: A comparison of Extrinsic Clustering Evaluation Metrics based on Formal Constraints. J. Information Retrieval 12, 461–486 (2009)
25. Bezdek, J.: Cluster validity with fuzzy sets. J. of Cybernetics 3, 58–73 (1974a)
26. Bezdek, J.: Numerical taxonomy with fuzzy sets. J. of Mathematical Biology 1, 57–71 (1974b)
27. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 841–847 (1991)
28. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. J. of Intelligence Information System. 17, 107–145 (2011)
29. Shapiro, S.S., Wilk, M.B.: An Analysis of Variance Test for Normality. Biometrika 52, 591–611 (1965)

30. Friedman, M.: The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. J. of the American Statistical Association 32, 675–701 (1937)
31. Derrac, J., Garcia, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1, 3–18 (2011)
32. Fisher, R.A.: Statistical methods and scientific inference, 2nd edn. Hafner Publishing Co., New York (1959)
33. Freedman, D., Pisani, R., Purves, R.: Statistics, 3rd edn., pp. A-107 . W.W. Norton & Company, New York (1998)

Self-Organizing Polynomial Neural Networks Based on Matrix Inversion and Differential Evolution

Lorena G.N. Tablada and Mêuser J.S. Valença

Polytechnic School of Pernambuco, University of Pernambuco,
Recife, Pernambuco, Brazil
{lgnt,meuser,mestrado}@ecomp.poli.br

Abstract. Although Artificial Neural Networks (ANNs) have been extensively used to solve forecasting problems, defining their architectures has commonly been a very difficult task. Self-Organizing Polynomial Neural Networks can be used to alleviate this problem. However, it causes an increase in the computational cost and the addition of other parameters. This first drawback can be mitigated by using a matrix inversion technique as training algorithm, while the second, by using Differential Evolution. The method developed in this study combines those techniques in order to simultaneously search for the best parameters, the network architecture and weights. Finally, one can observe that in most databases the proposed method outperformed the Backpropagation, the most commonly used training algorithm in ANNs.

Keywords: Self-Organizing Neural Networks, Matrix Inversion, Differential Evolution.

1 Introduction

One of the major problems regarding Artificial Neural Networks (ANNs) is related to the definition of their architectures. The amount of hidden layers and neurons located in each layer are difficult parameters to set mainly when the method used is the trial and error. Considering that a certain amount of simulations is required for each analysed architecture, the high computational cost caused by the lack of knowledge of the optimal architecture is well known.

In order to overcome ANN architecture definition problem, the Self-Organizing Polynomial Neural Network (SOPNN) can be used since it is able to self-organize by adding, testing and removing neurons and layers to its architecture. However, the utilization of SOPNN leads to an increase in the computational cost and other parameters, such as the amount of input variables in each neuron and the polynomial order.

While the first SOPNN drawback can be mitigated by using a matrix inversion technique known as Single Value Decomposition (SVD), the second one is alleviated by using the Differential Evolution (DE) algorithm. The combination of these algorithms provides a fast and automatic definition of its architecture.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 399–406, 2012.

In this paper we compare the performance of the presented model to a ANN using Backpropagation (BP) as training algorithm. The remainder of this paper is organized as follows: The SOPNN, Matrix Inversion and Differential Evolution algorithms are briefly described in sections 2, 3 and 4 respectively. In the next section the experimental setup is explained followed by the Results section. In the next section the conclusions are given and, finally, the references are in the last section.

2 Self-Organizing Polynomial Neural Network

The Group Method of Data Handling (GMDH) was first introduced by Ivakhnenko [1] in the late 1960's as a method for identifying non-linear relations among input and output variables. Since the mid-1970's it has been extensively used for prediction and modeling of complex nonlinear processes.

The main characteristics of GMDH are that it is self-organizing and provides an automated selection of the important input variables without using prior information about the input-output variables relationship. On the other hand it has some drawbacks, such as its tendency to generate complex polynomials to solve relatively simple problems and, if there are less than three input variables, it does not generate a highly versatile structure.

To minimize the problems associated with GMDH, SOPNNs were introduced by Oh [2],[3], *et al.* Those networks are highly flexible since each neuron can have a different amount of inputs, unlike common ANNs. Therefore, SOPNN can explore a different type of polynomial, as explained below. The SOPNN algorithm steps can be described as follows:

- Determine the inputs variables of the system according to the chosen database;
- Create the training, cross-validation and testing data sets;
- Choose its structure: the amount of input variables and the type of polynomial can be fixed or vary in each neuron;
- Determine the amount of input variables and the type of polynomial of the neuron. In this study the outputs of the individual neurons are expressed as second order regression equations. In particular, when two inputs are combined in each neuron, according to table 1, there is the following relation:

$$y = A + BX_i + CX_j + DX_i^2 + EX_j^2 + FX_iX_j \tag{1}$$

where A, B, C, D, E e F are parameters, while X_i and X_j denote two inputs and y is the output of the model. The just given example can be represented as in figure 1, where X_i and X_j are neurons combined in order to form the neuron X_k input, whose output is y, according formula 1. The outputs obtained in each one of these neurons are combined in order to obtain an even higher order polynomial.

Table 1. Different types of polynomials in neurons

Type of polynomial	Amount of inputs		
	1	2	3
1	Linear	Bilinear	Trilinear
2	Quadratic	Biquadratic	Triquadratic
3	Modified Quadratic	Modified Biquadratic	Modified Triquadratic

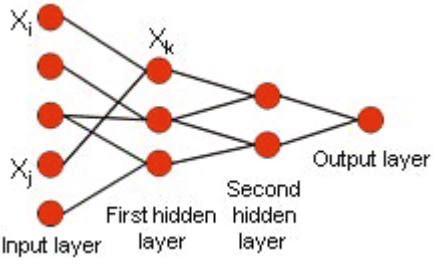


Fig. 1. Representation of a SOPNN

The formula 1 parameters A, B, C, D, E e F can be calculated through different ways. Among them, using a new ANN, which can be seen in figure 2. In this figure, $K_1 = 1$, $K_2 = X_i$, $K_3 = X_j$, $K_4 = X_i^2$, $K_5 = X_j^2$, $K_6 = X_i X_j$.

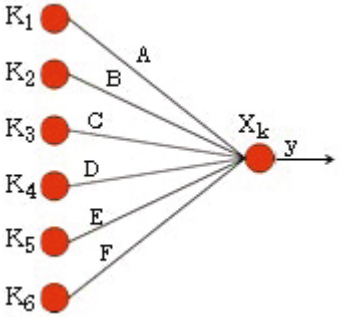


Fig. 2. ANN able to calculate bilinear function

- Estimate the errors of the neurons;
- Select the neurons with the best predictive capabilities;
- Check the stopping criterion.

It is well known that the use of a ANN like the one seen in figure 2 in order to calculate all neurons outputs would cause a extremely high computation cost. However, since this ANN has no hidden layers, matrix inversion can be used as a quick and efficient training technique.

3 Matrix Inversion

The Moore-Penrose Pseudo-inverse (M-PP) is a general way to find a solution to the following system of linear equations:

$$AX = B \tag{2}$$

where $A \in \mathbb{R}^{m \times n}, X \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{m}$. Moore and Penrose demonstrated that there is a general solution to this type of system of the form:

$$X = A^{+}B \tag{3}$$

where A^{+} is the M-PP of matrix A. They proved that this is the unique matrix which satisfies the following properties: $AA^{+}A = A$, $A^{+}AA^{+} = A^{+}$, $(AA^{+})^{T} = AA^{+}$ and $(A^{+}A)^{T} = A^{+}A$. The M-PP has the following properties if A is full rank:

- case $m = n$, $A^{+} = A^{-1}$;
- case $m < n$, $A^{+} = A^{T}(AA^{T})^{-1}$;
- case $m > n$, $A^{+} = (A^{T}A)^{-1}A^{T}$.

If A is not full rank, these formulas can not be used. More generally, they are best computed using SVD, whose greatest advantage is its calculation speed. Formally, the SVD of a matrix $A_{m \times n}$ is a factorization of the form:

$$A = U\Sigma V^{T} \tag{4}$$

where $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ and they are orthogonal. Σ is a $m \times n$ diagonal matrix having the form:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \sigma_p & 0 \end{bmatrix} \tag{5}$$

where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p \geq 0$ and $p = \min(m, n)$. Using SVD, the Moore-Penrose pseudo-inverse of A is given by:

$$A^{+} = V\Sigma^{-1}U^{T} \tag{6}$$

Applying the matrix inversion to the problem proposed in this study, for the first layer, A would be the input matrix of the data set; X would be the weights matrix; and B would be the output matrix of the data set. For the other layers, A would be the output of the previous layer. Thereby, applying equation 3, it is known that the weights matrix will be obtained by calculating the M-PP of the input matrix and multiplying this result by the output matrix.

4 Differential Evolution

Differential Evolution is a type of Evolutionary Algorithm. Therefore, it is based on Darwin's Theory of Evolution (1859), whose main mechanism is Natural Selection. In this mechanism, the disadvantaged individuals would gradually die, leaving only the advantaged ones, whose offspring would inherit that advantage and pass it on to their offspring. This process favours the generation of increasingly tailored individuals.

Originally proposed by Storn e Price [4], DE is a method based on a population, set of individuals composed by genes, that optimizes a problem by iteratively trying to improve a candidate solution, also known as fitness function, with regard to a given measure of quality. DE needs a few parameters to set, it is simple but effective and well known as a very fast algorithm. In DE some operations are executed with the individuals but according to [5] the mutation is the most important one.

The algorithm steps can be described as follows:

- Initialization: Initialize k sub-populations containing individuals randomly generated and different from each other;
- Opposition: For each sub-population A, create other sub-population composed by its opposed individuals.
- First selection: A first selection is executed by comparing the fitness functions of an individual to its opposed individual.
- Mutation: For each selected individual, called *base*, a new individual P_m is generated through the weighted sum of *base* and the subtraction of two other different individuals *r1* and *r2*, all of them belonging to generation G:

$$P_{m,G} = P_{base,G} + F(P_{r1,G} - P_{r2,G}) \tag{7}$$

- Crossing: Each P_{base} is crossed with its mutated vector creating a new candidate solution P_c;
- Selection: Based on their fitness functions, P_{base} or P_c will be selected to be part of the next generation.;
- If the stopping criteria is not satisfied, go to opposition step.

There is also another operation which can be executed on individuals: migration. It prevents the algorithm being trapped in a local minima since the subpopulations share information at regular periods.

In this study, the DE algorithm is used to choose the last undefined parameter: which neurons to combine in order to form the other neurons inputs. Thereby, each individual gene represents one specific neuron. The greatest genes values indicate what neurons to combine.

5 Experimental Setup

In order to evaluate the proposed model, eight well known databases were used: Abalone, which represents the age of abalone from physical measurements; Auto

Price, which represents the cars prices according to attributes such as fuel type and length; California Housing, which represents the variables used during the 1990s Census in California; Computer Activity, which is a collection of computer systems activity measurements; Delta Ailerons, which represents the task of controlling the ailerons of a F16 aircraft; Delta Elevators, which is also obtained from the task of controlling the elevators of a F16 aircraft, although the target variable and attributes are different from the previous database; Machine CPU, which represents the relative CPU performance and Servo, which represents a simulation of a servo system involving a servo amplifier, a motor, a lead screw/nut, and a sliding carriage of some sort; most of them are available in [6]. For assessing the performance of the model, the Root Mean Square Error (RMSE) was used. It is given by the following equation:

$$RMSE = \sqrt{\frac{1}{mn}\sum_{1}^{m}\sum_{1}^{n}(d_{e,s} - o_{e,s})^2} \tag{8}$$

where m is the amount of examples, n is the amount of outputs, d is the desired output, o, the obtained output, e refers to the current example and s, the current output.

In these experiments, the SOPNN neurons were set up with two entries and type two. We performed 30 trials for each analysed database and the RMSE and standard deviation were calculated. All databases were preprocessed by using a normalization procedure and they were divided into training, validation and testing sets, 50%, 25% and 25% of the database, respectively.

6 Results

Table 2 shows the average RMSE and standard deviation for both algorithms. The best results are marked in bold. It also shows the Wilcoxon test p-values for each database. Since some results are quite similar, the boxplot chart for RMSE was depicted and it can be seen in figure 3.

The software R [7] was used in order to perform the statistical analysis of the experiments for 95% of confidence interval based on Wilcoxon test. The following conclusions can be drawn from the test and analysis of the databases RMSEs and standard deviations: the results were only similar for the Delta Elevators database, where the SOPNN outperformed the BP algorithm. For the other databases, the SOPNN model outperformed the BP algorithm for the Abalone, Auto Price, Delta Ailerons, Machine CPU and Servo databases. The BP algorithm achieved the best results for the California Housing and Computer Activity databases. Thus, one can notice that the SOPNN achieved the best results in 62.5% of the databases used in the statistical tests.

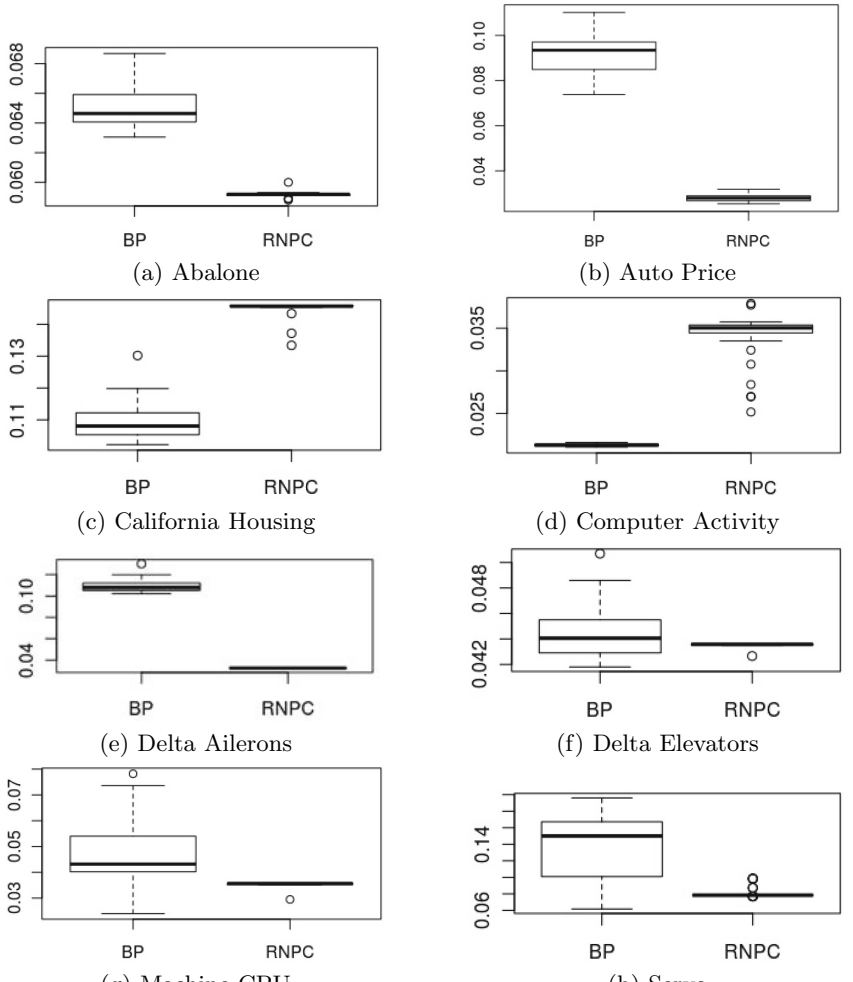


Fig. 3. Boxplot chart of the fitness for the SOPNN and BP for the databases: a) Abalone, b) Auto Price, c) California Housing, d) Computer Activity, e) Delta Ailerons, f) Delta Elevators, g) Machine CPU i) Servo

Table 2. Results obtained using the BP and SOPNN and Wilcoxon test p-values

Database	Backpropagation	SOPNN	P-values
Abalone	0.0625 ± 0.0043	**0.0593 ± 0.0001**	$1.579e^{-11}$
Auto Price	0.0919 ± 0.0079	**0.0279 ± 0.0015**	$2.2e^{-16}$
California Housing	**0.1096 ± 0.0058**	0.1459 ± 0.0026	$1.755e^{-11}$
Computer Activity	**0.0213 ± 0.0001**	0.0339 ± 0.0030	$2.928e^{-11}$
Delta Ailerons	0.0330 ± 0.001	**0.0323 ± 0.0001**	$1.649e^{-11}$
Delta Elevators	0.0445 ± 0.0022	0.0432 ± 0.0001	0.3158
Machine CPU	0.0468 ± 0.0145	**0.0349 ± 0.0011**	$4.426e^{-5}$
Servo	0.1345 ± 0.0392	**0.0810 ± 0.0063**	$1.088e^{-6}$

7 Conclusions and Future Works

This paper presented a model whose aims are the optimization of an ANN architecture definition, parameters set up and computational cost. The proposed model combines the SOPNN, Matrix Inversion and DE qualities. The model and a BP ANN were assessed using the RMSE. Thus, one can argue that, through the statistical analysis of the results, the proposed model obtained the best results in the major part of the data sets. On average, all the SOPNN results were obtained in less than 1 minute. This study can be concluded arguing that the results were quite satisfactory. The model efficiency stimulates the continuation of further research in the same field.

References

1. Ivakhnenko, A.G.: Polynomial theory of complex systems. IEEE Trans. on Systems, Man and Cybernetics 1, 364–378 (1971)
2. Oh, S.K., Pedrycz, W.: The design of self-organizing polynomial neural networks. Information Science 141, 237–258 (2002)
3. Oh, S.K., Pedrycz, W., Park, B.J.: Polynomial neural networks architecture: analysis and design. Computers and Electrical Engineering 29, 703–725 (2003)
4. Storn, R., Price, K.: Differential evolution–a simple efficient adaptive scheme for global optimization over continuous spaces. Technical Report 95-012, Int. Compt. Sci. Inst., Berkeley, CA (1995)
5. Zarth, A.M.F.: Otimização evolucionária multimodal de redes neurais artificiais com evolução diferencial. Master's thesis of Federal University of Pernambuco (2010)
6. Frank, A., Asuncion, A.: Uci machine learning repository
7. Chambers, J.: The r project for statistical computing

A Dilation-Erosion-Linear Perceptron for Bovespa Index Prediction

Ricardo de A. Araújo[1,2], Adriano L.I. Oliveira[1], and Silvio R.L. Meira[1]

[1] Informatics Center, Federal University of Pernambuco, Recife, PE, Brazil
[2] Informatics Department, Federal Institute of Sertão Pernambuco, Ouricuri, PE, Brazil
{raa,alio,srlm}@cin.ufpe.br, ricardo.araujo@ifsertao-pe.edu.br

Abstract. In this work we present the dilation-erosion-linear perceptron (DELP) for financial prediction. It is composed of morphological operators under context of lattice theory and a linear operator. A gradient-based method is presented to design the proposed DELP (learning process). Also, it is included an automatic phase fix procedure to adjust time phase distortions observed in financial phenomena. Furthermore, an experimental analysis is conducted with the proposed model using the Bovespa Index, where five well-known performance metrics and an evaluation function are used to assess the prediction performance.

Keywords: Hybrid Perceptrons, Mathematical Morphology, Lattice Theory, Gradient-based Learning, Financial Prediction.

1 Introduction

Linear and nonlinear statistical approaches have been proposed in the literature to solve the financial prediction problems [1]. However, these approaches have a drawback for the development of automatic prediction systems, because they have the need of a problem specialist to validate their predictions [1]. Alternatively, artificial neural networks (ANNs) [2,3] have been applied in the attempt to overcome such drawback. However, due to many complex features frequently present in financial phenomena, such as irregularities, volatility, trends, and noise, a limitation arises from all these models for financial prediction and is known as the random walk dilemma (RWD). It has been reported in the literature [2–6]. In this context, predictions generated by arbitrary models have a characteristic one step ahead delay with respect to the actual values [2–6]. This behavior has led some researchers to argue that financial phenomena are unpredictable [4,5].

In this way, this paper presents the dilation-erosion-linear perceptron (DELP) to overcome the RWD in the financial prediction problem. The proposed model is composed of morphological operators under context of lattice theory and a linear operator. The proposed learning process of the DELP employs a gradient-based method based on ideas from the back-propagation (BP) algorithm and using a systematic approach to overcome the problem of non-differentiability

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 407–415, 2012.

of morphological operations, based on ideas from Pessoa and Maragos [7] and Sousa [8]. Also, we have included into learning process of the DELP a procedure to overcome the RWD, called the automatic phase fix procedure (APFP) [2,3,6], which is a correction step that is geared toward eliminating the one step ahead delay that occur in financial phenomena prediction. Furthermore, an experimental analysis is conducted with the DELP using the Bovespa Index. Five metrics are used to assess the prediction performance. An evaluation function (EF) is further used as global prediction performance indicator. The obtained results are discussed and compared to results found using the random walk model [4,5].

2 The Time Series Prediction

A time series is a sequence of observations about a given phenomenon observed in a discrete or continuous space. In this work a time series will be considered time discrete and equidistant, and formally defined by

$$\mathbf{x} = \{x_t \in \mathbb{R} \mid t = 1, 2, \ldots, N\}, \tag{1}$$

where t is the temporal index, which is called time and defines the granularity of observations of a given phenomenon, and N is the number of observations.

The aim of prediction techniques applied to a given time series is to provide a mechanism that allows, with certain accuracy, the prediction of the future values of $\mathbf{x}$, given by x_{t+h}, $h = 1, 2, \ldots, H$, where h represents the prediction horizon of H steps ahead. These techniques try to identify certain regular patterns present in the data set, creating a model capable of generating the next temporal patterns, where, in this context, a most relevant factor for an accurate prediction performance is the correct choice of the past window, or the time lags, considered for the representation of a given time series.

In mathematical sense, the relationship which involves time series historical data defines a n-dimensional phase space, where n is the minimum dimension capable of representing such relationship. Therefore, a n-dimensional phase space can be built so that it is possible to unfold its corresponding time series. Takens [9] proved that if n is sufficiently large, such phase space is homeomorphic to the phase space that generates the series. The Takens' Theorem [9] is the theoretical justification of the phase space reconstruction using time lags.

2.1 The Random Walk Dilemma

A naive prediction strategy is to define the last observation of a time series as the best prediction of its next future value ($x_{t+1} = x_t$). This kind of model is known as the random walk (RW) model [4], which is defined by

$$x_t = x_{t-1} + z_t, \tag{2}$$

where x_t is the current observation, x_{t-1} is the immediate observation before x_t, and z_t is a noise term with a Gaussian distribution of zero mean and constant

standard deviation. The model above clearly implies that, as the information set consists of past time series data, the future data is unpredictable. Therefore, on average, the value x_{t-1} is indeed the best prediction of value x_t, and proof of this statement is given in [3,6].

It is possible to verify that the use of an arbitrary model to make predictions have an intrinsic limitation, since the generated predictions have a characteristic one step ahead delay regarding the original time series values, in which this behavior is common in the finance and economics and is called random walk dilemma or random walk hypothesis [4]. Therefore, in these conditions, to escape of the random walk dilemma is a hard task [3,6].

3 The Dilation-Erosion-Linear Perceptron

The proposed dilation-erosion-linear perceptron (DELP) consists of a linear combination of a nonlinear operators (dilation and erosion operators) and a liner operator (finite impulse response). Next we present the definition and the proposed training algorithm to design the DELP.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ a real-valued input signal inside an n-point moving window and let y the model output. Then, the DELP is defined as a morphological-linear system with local transformation rule $\mathbf{x} \to y$, given by

$$y = \lambda\alpha + (1-\lambda)\beta, \qquad \lambda \in [0,1], \tag{3}$$

where

$$\beta = \mathbf{x} \cdot \mathbf{p}^T = x_1 p_1 + x_2 p_2 + \ldots + x_n p_n, \tag{4}$$

and

$$\alpha = \theta\varphi + (1-\theta)\omega, \qquad \theta \in [0,1], \tag{5}$$

in which

$$\varphi = \delta_{\mathbf{a}}(\mathbf{x}) = \bigvee_{i=1}^{n} (x_i + a_i), \tag{6}$$

and

$$\omega = \varepsilon_{\mathbf{b}}(\mathbf{x}) = \bigwedge_{i=1}^{n} (x_i +' b_i), \tag{7}$$

where term n denotes the dimensionality of the input signal ($\mathbf{x}$), terms $\lambda, \theta \in \mathbb{R}$ and $\mathbf{a}, \mathbf{b}, \mathbf{p} \in \mathbb{R}^n$. The vector $\mathbf{p} \in \mathbb{R}^n$ represents the linear operator weights. The term β represents the output of the linear operator. The term α represents the linear combination of the morphological operators of dilation and erosion (the mixture term is defined by θ). The terms φ and ω represent the output of morphological operators of dilation and erosion, respectively. The vectors $\mathbf{a}$ and $\mathbf{b}$ represent the structuring elements (weights) of the dilation ($\delta_{\mathbf{a}}(\mathbf{x})$) and erosion ($\varepsilon_{\mathbf{b}}(\mathbf{x})$) operators employed into the nonlinear module of the DELP. Terms $\bigvee$ and $\bigwedge$ represent the supremum and the infimum operations. Note that the output y is given by a linear combination of the linear operator and

another linear combination of morphological operators of dilation and erosion (the mixture term is defined by λ). The main differences between "$+'$" and "$+$" are given by the following rules:

$$(-\infty) + (+\infty) = (+\infty) + (-\infty) = -\infty, \tag{8}$$

and

$$(-\infty) +' (+\infty) = (+\infty) +' (-\infty) = +\infty. \tag{9}$$

3.1 Learning Process

The design of the DELP model requires the adjustment of the parameters $\mathbf{a}, \mathbf{b}, \mathbf{p} \in \mathbb{R}^n$ and $\lambda, \theta \in \mathbb{R}$. Therefore, the weight vector $\mathbf{w} \in \mathbb{R}^{3n+2}$ of the DELP model is given by

$$\mathbf{w} = (\mathbf{a}, \mathbf{b}, \mathbf{p}, \lambda, \theta). \tag{10}$$

During the proposed learning process, all parameters of the DELP model are iteratively adjusted according to an error criterium until de convergence. Therefore, it is necessary to define an objective function $J(\mathbf{w})$ to be minimized during the learning process, and given by

$$J(\mathbf{w}) = \sum_{m=1}^{M} e^2(m), \tag{11}$$

in which M represents the input patterns amount in the learning process and $e(m)$ represents the instantaneous error for the m-th input pattern, and given by

$$e(m) = t(m) - y(m), \tag{12}$$

where $t(m)$ and $y(m)$ are the target and the model output, respectively.

The learning process updates the weight vector $\mathbf{w}$ based on the gradient steepest descent method. The adjustment of vector $\mathbf{w}$ for the m-th input training pattern is given by the following iterative formula:

$$\mathbf{w}(i+1) = \mathbf{w}(i) - \mu \nabla J(\mathbf{w}), \tag{13}$$

where $\mu > 0$ and $i \in \{1, 2, \ldots\}$. Term $\nabla J(\mathbf{w})$ is the gradient, which is given by

$$\nabla J(\mathbf{w}) = \frac{\partial J}{\partial \mathbf{w}} = \left(\frac{\partial J}{\partial \mathbf{a}}, \frac{\partial J}{\partial \mathbf{b}}, \frac{\partial J}{\partial \mathbf{p}}, \frac{\partial J}{\partial \lambda}, \frac{\partial J}{\partial \theta} \right), \tag{14}$$

in which

$$\frac{\partial J}{\partial \mathbf{w}} = -2e(m) \left(\frac{\partial y}{\partial \mathbf{a}}, \frac{\partial y}{\partial \mathbf{b}}, \frac{\partial y}{\partial \mathbf{p}}, \frac{\partial y}{\partial \lambda}, \frac{\partial y}{\partial \theta} \right), \tag{15}$$

Note that the existence of the gradient of J with respect to $\mathbf{w}$ depends on the existence of the gradients $\frac{\partial y}{\partial \mathbf{a}}$, $\frac{\partial y}{\partial \mathbf{b}}$, $\frac{\partial y}{\partial \mathbf{p}}$, $\frac{\partial y}{\partial \lambda}$ and $\frac{\partial y}{\partial \theta}$. Next, we present the formulas to calculate them.

The term $\frac{\partial y}{\partial \lambda}$ is given by

$$\frac{\partial y}{\partial \lambda} = \alpha - \beta. \tag{16}$$

The term $\frac{\partial y}{\partial \mathbf{p}}$ is given by

$$\frac{\partial y}{\partial \mathbf{p}} = \frac{\partial y}{\partial \beta}\frac{\partial \beta}{\partial \mathbf{p}}, \tag{17}$$

in which

$$\frac{\partial y}{\partial \beta} = 1 - \lambda, \tag{18}$$

and

$$\frac{\partial \beta}{\partial \mathbf{p}} = \mathbf{x}, \tag{19}$$

where $\mathbf{x}$ represents the input signal (m-th input training pattern).

The term $\frac{\partial y}{\partial \theta}$ is given by

$$\frac{\partial y}{\partial \theta} = \frac{\partial y}{\partial \alpha}\frac{\partial \alpha}{\partial \theta}, \tag{20}$$

in which

$$\frac{\partial y}{\partial \alpha} = \lambda, \tag{21}$$

and

$$\frac{\partial \alpha}{\partial \theta} = \varphi - \omega. \tag{22}$$

Terms $\frac{\partial y}{\partial \mathbf{a}}$ and $\frac{\partial y}{\partial \mathbf{b}}$ are estimated using the concept of smoothed rank indicator vector [7,8] (because dilation and erosion operators can be seen as particular cases of the rank function), where we choose the smoothed unit sample function $Q_\sigma(\mathbf{x}) = [q_\sigma(x_1), q_\sigma(x_2), \ldots, q_\sigma(x_n)]$, in which

$$q_\sigma(x_i) = sech^2\left(\frac{x}{\sigma}\right), \forall\, i = 1, \ldots, n\,. \tag{23}$$

Note that the choice of the scale factor σ directly affect the estimation and interpolation of the gradients $\frac{\partial y}{\partial \mathbf{a}}$ and $\frac{\partial y}{\partial \mathbf{b}}$. However, the learning process of the DELP model even works with $\sigma \to 0$, since in this particular case, the gradient will be given in terms of the usual rank indicator vector [7,8].

Therefore, the term $\frac{\partial y}{\partial \mathbf{a}}$ is given by

$$\frac{\partial y}{\partial \mathbf{a}} = \frac{\partial y}{\partial \alpha}\frac{\partial \alpha}{\partial \varphi}\frac{\partial \varphi}{\partial \mathbf{a}} = \lambda\frac{\partial \alpha}{\partial \varphi}\frac{\partial \varphi}{\partial \mathbf{a}}, \tag{24}$$

in which

$$\frac{\partial \alpha}{\partial \varphi} = \theta, \tag{25}$$

and

$$\frac{\partial \varphi}{\partial \mathbf{a}} = \frac{Q_\sigma\left(\varphi \cdot \mathbf{1} - (\mathbf{x} + \mathbf{a})\right)}{Q_\sigma\left(\varphi \cdot \mathbf{1} - (\mathbf{x} + \mathbf{a})\right) \cdot \mathbf{1}^T}. \tag{26}$$

As the same way, the term $\frac{\partial y}{\partial \mathbf{b}}$ is given by

$$\frac{\partial y}{\partial \mathbf{b}} = \frac{\partial y}{\partial \alpha}\frac{\partial \alpha}{\partial \omega}\frac{\partial \omega}{\partial \mathbf{b}} = \lambda \frac{\partial \alpha}{\partial \omega}\frac{\partial \omega}{\partial \mathbf{b}}, \tag{27}$$

in which

$$\frac{\partial \alpha}{\partial \omega} = 1 - \theta, \tag{28}$$

and

$$\frac{\partial \omega}{\partial \mathbf{b}} = \frac{Q_\sigma\left(\omega \cdot \mathbf{1} - (\mathbf{x} +' \mathbf{b})\right)}{Q_\sigma\left(\omega \cdot \mathbf{1} - (\mathbf{x} +' \mathbf{b})\right) \cdot \mathbf{1}^T}. \tag{29}$$

Furthermore, in order to automatically overcome the random walk dilemma in financial time series prediction problem, we have included an automatic phase fix procedure (APFP) [3,6] in the proposed learning process of the DELP model. Figure 1 presents the APFP.

According to the Figure 1, in the first step an input pattern $\mathbf{x}$ is presented to DELP generating the output y_1. The first output y_1 is used to rebuild the input pattern in the second step. This reconstructed pattern is presented to the same DELP generating the second output y_2, which is the phase fixed prediction.

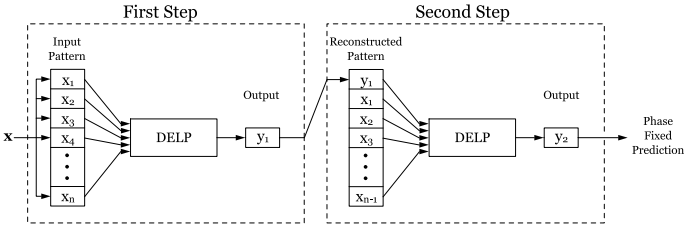


Fig. 1. Automatic phase fix procedure

4 Simulations and Experimental Results

The Bovespa Index time series was used as a test bed for evaluation of the proposed DELP model. This series was normalized to lie within the range $[0, 1]$ and divided in three sets according to Prechelt [10]: training set (50%), validation set (25%) and test set (25%).

For the experiments, the entries of the DELP weight, vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{p}$, are randomly initialized within the range $[-1, 1]$. The initial DELP mixture coefficients, λ and θ, are randomly chosen in the interval $[0, 1]$. Based on exhaustive experiments to determine the best learning rate (μ) and the scale factor (σ), we use $\mu = 0.01$ and $\sigma = 1.5$. It is worth mentioning that three stop conditions are used into the learning process [10]: i) The maximum epoch number equals to 10^4; ii) The decrease in the training error process training (Pt) of the cost

function equals to 10^{-6}; iii) The increase in the validation error or generalization loss (Gl) of the cost function equals to 5%.

In order to establish a performance study, results with the random walk (RW) model [4,5], which represents the results generated by classical prediction models, are employed in our comparative analysis, where we investigate the same time series under the same conditions. Additionally, we have used five well-known evaluation metrics formally defined in [3,6] to assess the prediction performance: mean square error (MSE), mean absolute percentage error (MAPE), u of theil statistic (UTS), prediction of change in direction (POCID) and average relative variance (ARV). Also, we use an evaluation function (EF) defined in [6] to serve as a global prediction performance indicator.

4.1 Bovespa Index Series

The *Bolsa de Valores do Estado de São Paulo* (Bovespa) Index is the most important average indicator of the stock prices from Brazilian market. The Bovespa series corresponds to daily records of Bovespa Index from 2008/04/07 to 2012/04/13. For the Bovespa series prediction (with one step ahead of prediction horizon – $H = 1$), we use the time lags (2-151 – note that here $n = 150$) to create the input patterns. The Table 1 shows the obtained results with RW and DELP models for all evaluation metrics.

Table 1. Obtained results with RW and DELP model for Bovespa series (test set)

	Evaluation Metrics					
Models	MSE	MAPE	UTS	ARV	POCID	EF
RW	4.2931e-004	2.4109e-002	1.0000e-000	4.0110e-002	51.42	24.9427
DELP	5.2968e-006	2.8898e-003	1.2298e-002	4.9487e-004	100.00	98.4714

It is possible to verify, according to Table 1, that in all experiments the POCID metric greater than 50%, indicating that the DELP model has much better performance than a "coin-tossing" experiment. The obtained UTS metric value ($\simeq$1.23e-002) indicates that the DELP model was able to overcome the random walk dilemma. Note that the MAPE metric value ($\simeq$2.89e-003) is very small, that is, without high percentage deviations. According to ARV metric value ($\simeq$4.95e-004), we can see a much better performance of the proposed model regarding a naive prediction model. Also, we can verify a small value of MSE metric ($\simeq$5.30e-006), which means that the predictions are too close to real values. The EF metric value ($\simeq$98.5) shows that the DELP have good global prediction performance. Therefore, we can see that the proposed DELP model overcame, for all evaluation metrics and for the evaluation function, the RW model in this work. Finally, we present in the Figure 2 a comparative graphic between real (solid line) and predicted (dashed line) values generated by DELP and RW models for the last twenty points of the Bovespa series test set. Note that the predicted values are very close to the real values of the Bovespa series, where the one step delay regarding the prediction values did not occur.

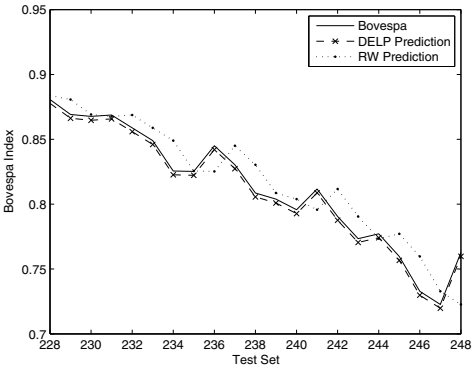


Fig. 2. Prediction results of Bovespa series (last twenty points of the test set): actual values (solid line) and predicted values (dashed line)

5 Conclusion

This paper presented the dilation-erosion-linear perceptron (DELP) to overcome the random walk dilemma in the financial prediction problem. The proposed model consists of nonlinear morphological operators under context of lattice theory and a linear operator. Also, we presented a learning process of the DELP employing a gradient-based method based on ideas from the back-propagation (BP) algorithm. Also, we have included into learning process of the DELP a procedure to overcome the RWD, which is a correction step that is geared toward eliminating the one step ahead delay that occur in financial phenomena prediction. The evaluation performance of the proposed DELP model regarding to random walk (RW) model was assessed in terms of five well-known performance measures and using the Bovespa Index series. In addition, an evaluation function served as a global indicator for the quality of solutions achieved by the investigated models.

The experimental results demonstrated a consistently better performance of the proposed DELP model, where we succeeded in to overcome the random walk dilemma in a particular financial prediction problem. It is possible to verify that our predictions have not any one step delay regarding real time series values. Further studies must be developed to better formalize and explain the properties of the DELP model and to determine its possible limitations with other time series with components such as trends, seasonalities, impulses, steps and other non-linearities. Further studies, in terms of risk and financial return, must be done in order to determine the additional economical benefits, for an investor, with the use of the DELP model in stock market applications. Finally, a particular theoretical study about the complexity of the DELP model must be done in order to establish a complete cost-performance evaluation of the DELP.

Acknowledgment. This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES), funded by CNPq and FACEPE, grants 573964/2008-4 and APQ-1037-1.03/08.

References

1. Clements, M.P., Franses, P.H., Swanson, N.R.: Forecasting economic and financial time-series with non-linear models. International Journal of Forecasting 20, 169–183 (2004)
2. de Araújo, R.A.: Swarm-based hybrid intelligent forecasting method for financial time series prediction. Learning and Nonlinear Models 5(2), 137–154 (2007)
3. Ferreira, T.A.E., Vasconcelos, G.C., Adeodato, P.J.L.: A new intelligent system methodology for time series forecasting with artificial neural networks. Neural Processing Letters 28, 113–129 (2008)
4. Sitte, R., Sitte, J.: Neural networks approach to the random walk dilemma of financial time series. Applied Intelligence 16(3), 163–171 (2002)
5. Malkiel, B.G.: A Random Walk Down Wall Street, Completely Revised and Updated Edition. W. W. Norton & Company (April 2003)
6. de Araújo, R.A., Ferreira, T.A.E.: An intelligent hybrid morphological-rank-linear method for financial time series prediction. Neurocomputing 72(10-12), 2507–2524 (2009)
7. Pessoa, L.F.C., Maragos, P.: Neural networks with hybrid morphological rank linear nodes: a unifying framework with applications to handwritten character recognition. Pattern Recognition 33, 945–960 (2000)
8. Sousa, R.P., Carvalho, J.M., Assis, F.M., Pessoa, L.F.C.: Designing translation invariant operations via neural network training. In: Proc. of the IEEE Intl. Conference on Image Processing, Vancouver, Canada (2000)
9. Takens, F.: Detecting strange attractor in turbulence. In: Dold, A., Eckmann, B. (eds.) Dynamical Systems and Turbulence. Lecture Notes in Mathematics, vol. 898, pp. 366–381. Springer, New York (1980)
10. Prechelt, L.: Proben1: A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94 (1994)

A Comparative Analysis of FSS with CMA-ES and S-PSO in Ill-Conditioned Problems

Anthony J. da C.C. Lins[1], Fernando B. Lima-Neto[1],
François Fages[2], and Carmelo J.A. Bastos-Filho[1]

[1] Polytechnic School of Engineering of University of Pernambuco
{ajccl,fbln,carmelofilho}@ecomp.poli.br
[2] Contraintes, INRIA Rocquencourt
francois.fages@inria.fr

Abstract. This paper presents a comparative analyzes between three search algorithms, named Fish School Search, Particle Swarm Optimization and Covariance Matrix Adaptation Evolution Strategy applied to ill-conditioned problems. We aim to demonstrate the effectiveness of the Fish School Search in the optimization processes when the objective function has ill-conditioned properties. We achieved good results for the Fish School Search and in some cases we obtained superior results when compared to the other algorithms.

Keywords: Fish School Search, Covariance matrix adaptation, Particle Swarm Optimization, Ill-conditioned problems, Invariance, Non-separable problems.

1 Introduction

Several real world problems are modeled through ill-conditioned functions due to the complexity of the relation between their components. An ill-conditioned function is a type of function that has variables which can generate great impact on the final results upon minimal adjustments. Hansen *et al.* [1] presented a comprehensive study on the performance of different search algorithms under different conditions of optimization when tackling problems with ill-conditioned characteristics. Hansen *et al.* investigated the performance of two stochastic search methods in ill-conditioned functions and non-separable problems for a set of benchmark functions with a different number of conditions. They assessed a particular version of the PSO (particle swarm optimization) [2] [3] and also an implementation of the CMA-ES (covariance matrix adaptation evolution strategy) algorithm [4] [5] [6] .

Recently, Bastos-Filho and Lima-Neto [7] [8] proposed a swarm intelligence technique for searching and optimization, called Fish School Search (FSS). FSS was inspired by the gregarious behavior of fish schools and it can tackle multimodal problems in high dimensional search spaces. In this paper, we assess the performance of the FSS algorithm when applied to problems with specific properties, such as ill-conditioning and non-separability. We also compare the

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 416–422, 2012.

performance of the FSS to the performance of two other approaches, S-PSO and CMA-ES.

The remainder of the paper is organized as follows: in section 2 we present an overview of the FSS algorithm. The concepts about ill-conditioned functions are described in section 3. The simulation setup, the benchmark functions and some other aspects are addressed in section 4. In section 4, we also present an analysis regarding a great number of parameter combinations for the simulated functions. In section V we give our conclusions.

2 FSS - Fish School Search

Fish School Search (FSS) is a bio-inspired algorithm for searching in high-dimensional and multimodal search spaces. The FSS functionality is based on a population of limited-memory individuals, called fish. FSS has two classes of operators [9], the feeding operator and the swimming operators. The fish school aims to achieve a collective goal that is to find food. Food is a metaphor for the fitness function, which measures the quality of the current solution. The pseudocode of the FSS algorithm can also be found in [9].

2.1 FSS Operators

Feeding Operator: The feeding operation is executed after the individual movement of the fish. Depending on the individual movement, the weight of the fish may increase or decrease, which indicates if this movement was successful or not. Bastos-Filho *et al.* [9] proposed to use an initial weight equal to 1, and it can vary up to a maximum scale value (W_{scale}). The weight is updated based on the current value of the weight summed to the normalized difference between the current fitness and the fitness at the new position considering the whole school. The weights of the fish are updated once in every FSS cycle by the feeding operator, according to equation (1):

$$W_i(t+1) = W_i(t) + \frac{\Delta f_i}{max(\Delta f)}, \tag{1}$$

where $W_i(t)$ is the weight of the fish i, Δf_i is the difference between the fitness value at the new position and the fitness value at the current position of each fish. $max(\Delta f)$ is the maximum value of Δf_i in the iteration considering the whole school.

Swimming Operator. In nature, animals react instinctively to environmental stimuli. For the fish, swimming is directly related to all significant individual and collective behaviors of the school, such as feeding, reproduction, escape from predators, move to safer positions of the aquarium or just stay grouped. In FSS, swimming is a vectorial composition of three movements: individual, collective-instinctive and collective-volitive. These movements are detailed below.

i. **Individual movement:** The individual movement operator is executed for all fish at each iteration of the algorithm. Each fish calculates randomly a new position within the search space in its neighborhood and evaluates the new position using the fitness function. If the fitness of the neighbor position is better, then the fish moves to the new position. Otherwise, the fish remains in the current position.

The neighbor position is calculated by adding to each component of the current position a random value generated by an uniform distribution in the interval [-1,1] multiplied by a step as shown in equation (2).

$$\boldsymbol{n}_i(t) = \boldsymbol{x}_i(t) + \boldsymbol{rand}(-1, 1)\, step_{ind}, \tag{2}$$

in which $\boldsymbol{x}_i(t)$ is the current position of the i^{th} fish, $\boldsymbol{n}_i(t)$ is the candidate position of the fish and $\boldsymbol{rand}()$ is a vector of random number generated in each iteration. $step_{ind}$ is given as a percentage of the search space and decays linearly with the iterations. The individual step size of the fish is updated according to the equation (3) in order to allow more exploration in the beginning and more exploitation in the end of the search process.

$$step_{ind}(t+1) = step_{ind}(t) - \frac{(step_{ind\,initial} - step_{ind\,final})}{iterations}, \tag{3}$$

in which $iterations$ is the total number of iterations, $step_{ind\,initial}$ and $step_{ind\,final}$ are the initial and final steps, respectively.

ii. **Collective-Instinctive movement:** After the individual movement of all fish, the algorithm calculates the weighted movement of the whole school for fish that performed the individual movement. The resultant direction ($\boldsymbol{I}(t)$) is more influenced by the fish that have highest weights. $\boldsymbol{I}(t)$ is calculated according to the equation (4). Then, all the fish update their positions by using equation (5).

$$\boldsymbol{I}(t) = \frac{\sum_{i=1}^{N} \Delta \boldsymbol{x}_i \Delta f_i}{\sum_{i=1}^{N} \Delta f_i}, \tag{4}$$

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{I}(t). \tag{5}$$

iii. **Collective-Volitive movement:** After the two former movements, the algorithm calculates if the weight of the whole school had increased. If it is the case, we have an indication that the search process had success in the iteration and the radius of the school should contract to allow more exploitation. Otherwise, the radius should increase in order to allow the school to get out of local minima and find better regions of the search space. This operator balances the exploration/exploitation trade-off.

The expansion or contraction of the school is applied as a small adjustment for each position of the fish with respect to the barycenter of the school. The calculation of the school barycenter is obtained by a weighted average position of all fish weighted by its respective weight, as shown in equation (6).

$$\boldsymbol{B}(t) = \frac{\sum_{i=1}^{N} \boldsymbol{x}_i W_i(t)}{\sum_{i=1}^{N} W_i(t)}. \tag{6}$$

The expansion or contraction of the radius of the school is calculated by comparing the weight of the school in the previous iteration and the current iteration. If the weight of the school had increased, the fish must update their positions according to (7). Otherwise, all fish must update their positions using equation (8).

$$\boldsymbol{x}(t+1) = \boldsymbol{x}(t) - step_{vol}\, rand(0,1)\, \frac{(\boldsymbol{x}(t) - \boldsymbol{B}(t))}{distance(\boldsymbol{x}(t), \boldsymbol{B}(t))}, \tag{7}$$

$$\boldsymbol{x}(t+1) = \boldsymbol{x}(t) + step_{vol}\, rand(0,1)\, \frac{(\boldsymbol{x}(t) - \boldsymbol{B}(t))}{distance(\boldsymbol{x}(t), \boldsymbol{B}(t))}, \tag{8}$$

in which $distance()$ is a function that calculates the Euclidean distance between the barycenter and the current position of the fish. $step_{vol}$ is the step size used to control the movement of fish to or from the barycenter. We used $step_{vol} = 2step_{ind}$.

3 Ill-Conditioned Functions

In [1], Hansen et al. investigate the behavior of PSO (particle swarm optimization) [2] [3] and CMA-ES (covariance matrix adaptation evolution strategy) [4] [5] [6] on ill-conditioned and non-separable objective functions.

An objective function is separable with respect to coordinate i if the optimal value for the ith coordinate does not depend on the values for the other coordinates. A separable optimization problem in dimension n can be solved by solving n optimization problems in one dimension. Separable problems are thus easy to solve with a linear complexity in the dimension of the problem, while non-separable problems are hard to solve and may involve an exponential complexity in the number of dimensions.

Furthermore, ill-conditioned functions correspond to situations where small differences in variables, *i.e.* in different directions in the search space, can generate very different results for the evaluation function, by several orders of magnitude. Ill-conditoned objective functions can hardly guide the search of optimal solutions and thus provide challenges for any optimization methods.

4 Comparing FSS to CMA-ES and S-PSO

4.1 Simulation Setup

In [1], a benchmark of hard non-separable ill-conditioned functions was tried with two well-established optimization methods, each of them with a specific

bio-inspired stochastic search: swarm intelligence (PSO) and evolutionary strategy (CMA-ES). In this section, we present the results obtained with FSS on this benchmark.

The objective functions are given in Table 1. The search space domain is $[-20, 80]^n$ in both cases. According to the original paper, in the Rosenbrock function the parameter α tunes the width of the bent ridge that guides to the global optimum. In the classical Rosenbrock function α equals 100. For smaller α the ridge becomes wider and the function becomes less difficult to solve. The α value will be vary between one and 10^8. Rastringin function was implemented and tested without any kind of modifications.

Table 1. Test functions rosenbrock and rastrigin and target function values

Function	f_{target}
$F_{Rosenbrock}(\boldsymbol{x}) = \sum_{i=1}^{n}[\alpha(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	10^{-9}
$F_{Rastrigin}(\boldsymbol{x}) = 10n + \sum_{i=1}^{n}[x_i^2 - 10cos(2\pi x_i)]$	10^{-9}

For the configuration setup and tests, we used all parameters according to the S-PSO presented in the original paper. We used 10 dimensions for both benchmark functions. For the Rosenbrock function, we used 16 fish. For the Rastrigin function, we used the following swarm sizes 10, 16, 30, 100, 300 and1000. The simulations were performed using 21 runs of the algorithm, for each configuration in both functions. In each of the executions of the algorithm, the evaluation function call was 10^7 times. The stopping criterion for each run is when the maximum number of function evaluations was reached. The FSS implementation was based on the *FSS-Vanilla* Java™version[1].

4.2 Simulation Results

Table 2 is a summary of the results obtained from the simulations using FSS for both benchmark functions. The table depicts the average and the (standard deviation) values for the fitness and the execution time for all simulations. We just achieved the target value for the optimization process in the Rastrigin function using 16 and 30 fish. One can observe that the target value (10^{-9}) was not reached for the Rastrigin function for the other configuration settings. We believe that 10 fish are not enough to provide the proper exploration of the entire search space, whereas too many fish can mitigate the exploitation capacity. We did not achieved the target value for the Rosenbrock function in any case.

[1] Published: http://www.fbln.pro.br/fss/versions.htm

Table 2. FSS simulations summary, considering fitness value (average and std deviation) and time, for each configuration

FSS				Fitness		Time(ms)	
Function	Dimensions	Size	α	Average	Std Dev	Average	Std Dev
Rosenbrock	10	16	1	2,76E-03	4,76E-04	3,11E+05	7,04E+02
			10	2,44E-02	4,88E-03	3,21E+05	4,64E+03
			100	2,67E-04	3,74765E-05	2,67E-04	3,74765E-05
			300	5,54E+00	3,94E-01	3,23E+05	1,21E+03
			1000	2,28E-04	3,51995E-05	2,28E-04	3,51995E-05
Rastrigin	10	10	-	3,05E-06	6,65E-07	2,35E+05	4,93E+02
		16	-	1,75514E-10	2,49508E-11	1,75514E-10	2,49508E-11
		30	-	8,38232E-11	1,80445E-11	8,38232E-11	1,80445E-11
		100	-	1,82707E-07	3,64708E-08	2,34E+06	2,06E+04
		300	-	4,56812E-08	7,84523E-09	7,09E+06	2,67E+05
		1000	-	9,56496E-09	1,37591E-09	2,38E+07	8,98E+04

Table 3 shows the results for the analysis of the effectiveness and (number of times the algorithm obtained the fitness target - maximum is 21 trials) considering the FSS, S-PSO and CMA-ES. Although the standard version of the FSS is not able to solve configurations tests, one can observe that the FSS can solve the problem with low number of fish, while the S-PSO and the CMA-ES need 300 particles and 1000 individuals, respectively, to solve the same problem.

Table 3. Analysis of the effectiveness (number of times the algorithm obtained the fitness target) considering the FSS, S-PSO and CMA-ES

Entities	10	16	30	100	300	1000
S-PSO	-	-	5%(1)	71%(15)	100%(21)	100%(21)
CMA-ES	-	-	-	24%(5)	76%(16)	100%(21)
FSS	-	100%(21)	100%(21)	-	-	-

5 Conclusions

In this paper we assessed the performance of the FSS for two well known ill-conditioned functions, Rastrigin and Rosenbrock. We compared the results with the results obtained by CMA-ES and S-PSO. The overall results were somehow counterintuitive, since FSS excelled for the apparently more ill-conditioned function (Rastrigin), producing excellent results for situations in which the two competing techniques not even could produce anything. And this could be due to the built-in set of mechanisms that endow FSS to perceive 'ill-conditions' of problems.

On the other hand, the topology of Rosenbrock function, with the global minimum inside a long parabolic shaped flat valley, makes it hard to spot the global minimum during the optimization process. The poor results obtained with FSS (for that particular function) are likely due to a premature convergence of the search process convergence in local minima. Though FSS has a mechanism to self-adjust exploration and exploitation modes, the topology of this apparently simple function does not allow FSS to escape from local minima. We hypothesize that the premature convergence of the FSS can be related to the linear decreasing of search steps for Individual and Volitive movement operators. In a future work, this kind of problem will be addressed.

Another point quite interesting is that FSS method is rather easy to adjust, as many parameters can be left without change. One reason for example is due to its self-control of exploration and exploitation modes.

Although counterintuitive, the increase of the number of fish may compromise the performance of the FSS method because not-directly it adds on the granularity of the search. This eventual artifact of acceleration of convergence is subject to change in the next release of FSS.

References

1. Hansen, N., Ros, R., Mauny, N., Schoenauer, M., Auger, A.: Impacts of invariance in search: When cma-es and pso face ill-conditioned and non-separable problems. Applied Soft Computing 11(8), 5755–5769 (2011)
2. Kennedy, J., Eberhart, R.: Particle swarm optimization, vol. 4, pp. 1942–1948 (1995)
3. Eberhart, R., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, pp. 84–88 (2000)
4. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation 9, 159–195 (2001)
5. Hansen, N.: The CMA evolution strategy: a comparing review. In: Lozano, J., Larranaga, P., Inza, I., Bengoetxea, E. (eds.) Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms, pp. 75–102. Springer (2006)
6. Hansen, N., Ostermeier, A.: Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 312–317 (May 1996)
7. Bastos-Filho, C.J.A., de Lima Neto, F.B., Lins, A.J.C.C., Nascimento, A.I.S., Lima, M.P.: A novel search algorithm based on fish school behavior. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2008, pp. 2646–2651 (2008)
8. Bastos Filho, C.J.A., de Lima Neto, F.B., Lins, A.J.C.C., Nascimento, A.I.S., Lima, M.P.: Fish School Search. In: Chiong, R. (ed.) Nature-Inspired Algorithms for Optimisation. SCI, vol. 193, pp. 261–277. Springer, Heidelberg (2009)
9. Bastos-Filho, C.J.A., de Lima-Neto, F.B., Sousa, M.F.C., Pontes, M.R., Madeiro, S.S.: On the influence of the swimming operators in the fish school search algorithm. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2009, pp. 5012–5017 (2009)

A Framework for Application of Tree-Structured Data Mining to Process Log Analysis

Dang Bach Bui[1,2], Fedja Hadzic[2], and Vidyasagar Potdar[1]

[1] School of Information Systems, Curtin University, Perth, Australia
[2] Department of Computing, Curtin University, Perth, Australia
dang.buibach@postgrad.curtin.edu.au,
{fedja.hadzic,v.potdar}@curtin.edu.au

Abstract. Many data mining and simulation based algorithms have been applied in the process mining field; nevertheless they mainly focus on the process discovery and conformance checking tasks. Even though the event logs are increasingly represented in semi-structured format using XML-based templates, commonly used XML mining techniques have not been explored. In this paper, we investigate the application of tree mining techniques and propose a general framework, within which a wider range of structure aware data mining techniques can be applied. Decision tree learning and frequent pattern mining are used as a case in point in the experiments on publicly available real dataset. The results indicate the promising properties of the proposed framework in adding to the available set of tools for process log analysis by enabling (i) direct data mining of tree-structured process logs (ii) extraction of informative knowledge patterns and (iii) frequent pattern mining at lower minimum support thresholds.

Keywords: process mining, frequent subtree mining, XML database, DSM.

1 Introduction and Related Works

A business process is a set of related activities following some logical order whose objective is to create a complete product or service for a customer or a market [1]. Business processes are now supported by computer systems and when a business process is executed it leaves traces in an event log (also called process log). From this log one can perform different types of analysis such as process discovery, conformance checking and process enhancement which are commonly called as process mining tasks [2]. Process discovery finds the model that best describes an event log. Many types of models are graphical-based and describe a variety of control flow constructs e.g. sequence, loop, choice, parallel, synchronization. Conformance checking tries to find any difference between the log and the model. If there are discrepancies, depending on whether the view of the model is descriptive or normative, one can conclude that either the model is wrong or the event log deviates from the desired model, respectively. Process enhancement analyzes other dimensions of the process log, e.g. actors and activities (work distribution), actors and actors (social network analysis), actor behaviors and decision mining [3], using a variety of data mining and

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 423–434, 2012.

statistical methods. Frequent pattern mining applied in the process mining area has been observed in [4] where a database of execution is examined to find the frequent sub-graph patterns that leads to a successful execution of a business process.

XML and XML based markup languages are now widely used in many domains due to flexibility, expressiveness, extensibility and portability nature. There is a recent momentum in representing event log in XML format. The first XML standard created for event log is *MXML* [5] and more recently, *XES* standard was proposed in [6]. Other attempts in using XML to store event logs are presented in [7] and [8]. Semi-structured documents such as XML are known for their ability to represent the contextual information among different data items in a domain specific way. Due to their hierarchical nature XML documents are commonly represented as rooted ordered labeled trees [9]. To enable the discovery of association between semi-structured data objects, the pre-requisite is to extract all frequent subtrees. This task corresponds to the frequent subtree mining (FSM) problem and is one of the most commonly utilized techniques for analyzing XML data. Many frequent subtree mining algorithms have been proposed to date but their application to analysis of business process data is limited due to the inherent complexity in data. To alleviate the complexity associated with mining complex structures and to enable a wider range of data analysis/mining techniques to be directly applied on tree-structured data, a structure-preserving flat data format of tree-structured data such as XML has been recently proposed in [10]. An interesting implication of the method is that the exact positions of nodes/attributes are taken into account during the knowledge discovery process. This property can be useful in process mining as events/activities are distinguished based on their context or exact occurrence within a trace of events. Using this technique as a basis, a general framework for encompassing a broad range of business process dimensions during the analysis phase is proposed. Frequent pattern mining and decision tree learning methods are used as case in point and applied to a publicly available real-world dataset.

2 Motivation and Scenario

Control flow/model discovery is currently the main focus within the process mining field. However, in many cases, the process may not be so well-structured resulting in "spaghetti-like" process models, as well as the representational bias where most existing approaches discover models that do not make much sense to the user [2]. This can hinder the analysis of the event log as well as introduce bias due to expectation driven analysis, currently largely done through simulation and visualization techniques. This motivates us to take a different approach to allow users to directly analyze the event log, without using the process model as a basis for analysis. For example, a user may be interested in detecting some specific instances of inconsistencies in process execution or generally discovering frequent associations among process aspects, (e.g. associations among business process activities, actors, exceptions, execution time etc.). To the best of our knowledge, no techniques capable of directly mining semi-structured event log have been explored in the process mining field. Even though *PROM* [2] can read the XES/MXML file directly but it still converts it into sequence/relational structure suitable for process mining algorithm. For example, the frequent subtree

mining methods are the basis for discovering interesting associations among tree-structured data objects in XML data, but their utilization in the process mining field is still to be explored. Similarly, methods capable of taking structural aspects into account during tasks such as clustering [11, 12] and classification [13] are yet to be explored in the business process mining research field.

A simplified example trace of a process log represented in XES format is displayed in Fig. 1, which was extracted from the publicly-available Dutch Hospital[1] process log dataset. Each trace contains an attribute *Diagnosis code* and *Age* (discretized in our example) group of the patient. Each event has an entity *concept:name* representing the name of the activity and an *org:group* showing the department where the activity is carried out. There can be additional entities e.g. *org:type* showing additional information of the activity. In the data mining field, an XML document is often modeled as a rooted ordered labeled tree and the strings/labels are mapped to integers for faster processing [9, 14]. On the left of Fig. 2, we provide an example string to integer mapping, and the *Trace 1* tree is the representative of the example trace from Fig. 1. In the tree mining field, pre-order string encoding is one of the ways used for efficient tree representation. A pre-order string encoding lists the nodes in the sequence of the pre-order traversal of a tree, and uses a special symbol (e.g. '*-1*') when backtracking up the tree. For example, the pre-order string encoding of the *Trace 1* in Fig. 2 is '*0 2 -1 4 -1 1 6 -1 11 -1 -1 1 7 -1 11 -1 -1 1 8 -1 12 -1 15 -1 -1*'.

```
<trace>
        <string key="Diagnosis code" value="M13">
        <int key="Age" value="21-65">
        <event>
                <string key="concept:name" value="Consult">
                <string key="org:group" value="Radio Therapy">
        </event>
        <event>
                <string key="concept:name" value="Administration">
                <string key="org:group" value="Radio Therapy">
        </event>
        <event>
                <string key="concept:name" value="Blood Test">
                <string key="org:group" value="General Lab">
                <string key="org:type" value="Hemoglobine">
        </event>
</trace>
<trace> ...
```

Fig. 1. One part of a simplified Hospital Process Log

Due to the tree-structured nature of the event log in XES, the activities/characteristics e.g. '*Consult*', '*Administration*', etc. are contextualized by the node '*event*'. The context of a set of activities/characteristics is important as it indicates their occurrence within particular stage of process execution, and they characterize that stage/event. By using different tree mining techniques, we can directly mine the process log without going through the process discovery step. For example,

[1] http://data.3tu.nl/repository/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54

applying frequent subtree mining at minimum support of three on the example tree database of Fig. 2 would discover a set of frequent subtrees that occurred in three traces. An example of such a frequent subtree is shown in Fig. 3. As one can see, a subtree pattern preserves the structural characteristics of the data and the activities/characteristics of a particular event are grouped together. However, the resulting subtree pattern does not clearly show the position of its subtrees '*1 6 -1 13 -1*' and '*1 9 -1 14 -1*' within the overall trace, and in reality there could be other events that occur in between the events, as is the case in *Trace 2* and *Trace 3* from Fig. 2. A certain event within a trace may need to be distinguished based on their occurrence within the workflow as a whole. This may be the case during conformance checking to a desired business process model, where if certain actions are performed at the different phase during process execution they have different implications. A method proposed in [10, 12] distinguishes tree-structured knowledge patterns based on their occurrence within a general database structure model. As such it forms the basis of the proposed framework explained next.

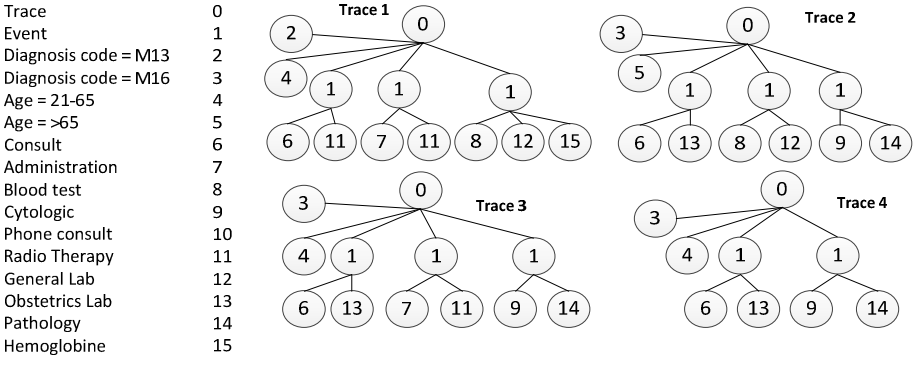


Fig. 2. Part of hospital log trace represented in trees

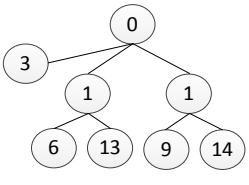


Fig. 3. An example frequent subtree of the hospital log trace in Fig. 2

3 Proposed Framework

The proposed conceptual framework is shown in Fig. 4 that consists of four main phases, pre-processing, database structure model (*DSM*) extraction and tree to flat representation [10], knowledge discovery and interpretation. Depending on the specific purpose of the process mining task, different pre-processing techniques could be

used e.g. grouping/removal, discretization, filtering etc. If the MXML/XES data is not ready, extract transform and load methods can be used [2]. This semi-structured file is modeled as a set of rooted ordered labeled trees and represented in a pre-order string encoding [14]. For definitions of tree concepts and overview of tree mining algorithms, please refer to [9, 14].

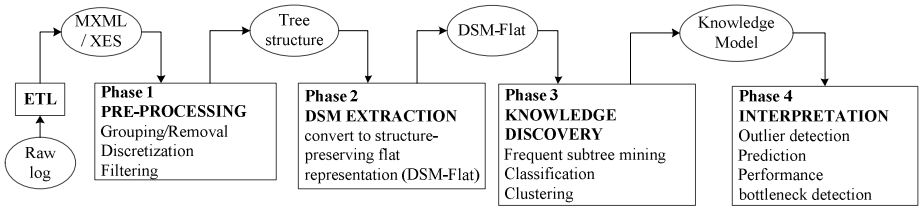


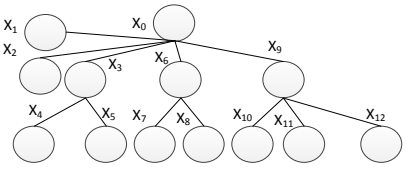
Fig. 4. Proposed framework

In phase 2 of the framework, using a technique described in [10] tree-structured data is converted into a flat data structure format (henceforth referred as table) so that both structural and attribute-value information is preserved. The approach starts by first extracting a database structure model (DSM) [10, 12] of which each tree instance is a valid subtree. This DSM contains the most general structure where every instance from the tree database can be matched to. The DSM tree of the tree database from Fig. 2 is shown in Fig. 5a. The pre-order string encoding of DSM will become the first row of the table where nodes x_i (i corresponds to the pre-order position of the node in the tree) and backtracks b_j (j corresponds to the backtrack number) are used as the attribute names. The remaining rows of the table are filled by matching the string encoding of the tree instances to the DSM. For each record, when a label is encountered, it is placed to the matching column under the matching node x_i in the DSM structure. When a backtrack ('*-1*') is encountered, a valued '*1*' is placed to the matching backtrack b_j. Remaining entries are assigned a value of '*0*' (non-existence). The flat representation of the example trace is shown in Fig. 5b.

To indicate structural characteristics of the knowledge patterns discovered, they can be re-mapped to the DSM. Hence, structural complexity is avoided and structural characteristics of the data are preserved. In this particular application the backtrack attributes and corresponding columns can be omitted from the flat representation, as they are kept in the DSM. For pseudo code of the conversion process and illustrative examples, please refer to [10, 12]. This conversion process enables the application of frequent pattern mining, clustering, classification and prediction techniques originally developed for vectorial data directly to tree structured process data (knowledge discovery phase in Fig. 4). For example, the frequent subtree mining can be done by converting the table into itemset format where each item is a combination of the node value and its corresponding position. Fig. 5c shows one frequent itemset at minimum support set to 3 and its corresponding subtree annotated with position value. The common characteristics among the three process instances is that they all share the same *Diagnosis code* ('*M16*') and the process starts with the '*Consult*' activity done in the '*Obstetrics Lab*' department (the contextual node of '*Consult*' is the node

'*Event*' at X_3 which flags the first event). The node '*Event*' at X_6 points out that there is another activity at the second event of three traces but they are not the same across the three traces. Furthermore, the activity '*Cytologic*' administered in '*Pathology*' department shown in Fig. 3 did not occur in DSM-based subtree, which indicates that these attributes were not frequent in the third event but were frequent when their occurrence in the second and third event was counted together.

Besides indicating the structural characteristics of the knowledge patterns discovered, in the interpretation phase of the framework, the patterns will be evaluated for their specific use in a given application. For example in outlier/exception detection, analysis and prediction, the different knowledge discovery techniques will be explored for their utilization for such purposes. For example, a low occurring frequent subtree pattern is suspect of being an outlier, and the difference to the more frequently occurring patterns reflecting the norm will be investigated. The instances that contain such rare patterns may correspond to an outlier. The classification methods can be useful for outlier prediction purposes. Hence, once the outlying instances are detected, the instances themselves will be labeled as outlying and others as norm and classification techniques are run to discover a classification model. This knowledge model can then be used to predict the outlying behavior, when a set of pre-conditions during business process execution path become true. In the next section we perform some preliminary experiments on a real-world data set to show the potential of the framework in adding to the available set of tools for process log analysis.

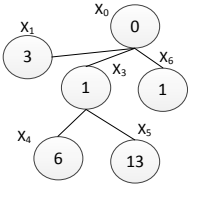


(a)

X_0	X_1	b_1	X_2	b_2	X_3	X_4	b_3	X_5	b_4	b_5	X_6	X_7	b_6	X_8	b_7	b_8	X_9	X_{10}	b_9	X_{11}	b_{10}	X_{12}	b_{11}	b_{12}
0	2	1	4	1	1	6	1	11	1	1	1	7	1	11	1	1	1	8	1	12	1	15	1	1
0	3	1	5	1	1	6	1	13	1	1	1	8	1	12	1	1	1	9	1	14	1	0	0	1
0	3	1	4	1	1	6	1	13	1	1	1	7	1	11	1	1	1	9	1	14	1	0	0	1
0	3	1	4	1	1	6	1	13	1	1	1	9	1	14	1	1	0	0	0	0	0	0	0	0

(b)

$X_0(0), X_1(3), X_3(1), X_4(6), X_5(13), X_6(1)$



(c)

Fig. 5. (a) DSM tree (b) Flat representation (c) One frequent closed itemsets at minimum support = 3 and its corresponding subtree

4 Experiments and Discussion

In the previous section we have described a general framework within which many data mining methods can be directly applied to tree-structured process logs. In this section we take frequent pattern mining and classification task as a case in point to demonstrate the applicability of the proposed approach and the usefulness of results in one domain. Frequent pattern mining is a pre-requisite and the most complex task that enable one to form association rules describing interesting associations between the data objects (e.g. associations occurring among process activities, actors, exceptions, execution time, etc.). In the first experiment we apply DSM based frequent pattern mining on the dataset to show the information content of the discovered patterns and their potential usefulness/actionability in the domain. This approach is compared with one traditional frequent subtree mining method for varying minimum support values. In the second experiment we focus on the classification task and use decision tree learning as a case in point. The classification methods are useful when the process log can be labeled with respect to a business need to build prediction model for a particular business aspect (e.g. duration, performance bottleneck, known cases of exceptions/fraud). The hospital dataset which is mentioned in Section 2 is used in our experiments. Each trace in the input file contains a set of attributes e.g. *diagnosis*, *diagnosis code*, *treatment*, *treatment code*, *start time*, *end time*, *age* etc. The dataset contains a large number of attributes; we mainly observe the attributes '*treatment code*' and the relationship between it and others. The preprocessing steps are (1) remove all attributes of the each trace except the '*treatment code*'; this attribute serves as a class for the classification purpose, (2) concatenate different treatment codes of each trace into one value (3) delete all traces but those belonging to the two most populous treatment codes ('*101*' and '*803*'), and (4) convert the XML file into a set of rooted ordered labeled trees. The structural properties of the resulting dataset are as follows: |transactions| = 506; avg. length of encoding = 639.4; max. tree size = 4778; avg. height of trees = 2; avg. fan-out of trees= 7.13; avg. size of trees = 320.2; max height of trees = 2; max. fan-out of trees = 598.

Frequent Closed/Maximal Pattern/Subtree Mining. We compare the time performance and patterns extracted from a frequent closed subtree mining algorithm *CMTreeMiner* [15] with the frequent closed itemset mining algorithm *LCM* [16] applied on the flat representation of the tree acquired using DSM method [10] (abbreviated as LCM-DSM). The purpose is to demonstrate that using the proposed approach lower support thresholds can be handled.

The experiments were run on Linux machine, Intel Xeon E5345 at 2.33 GHz, 8 GB RAM and 4MB Cache Open SUSE 10.2. The two tests are repeated with eight different support values. The number of frequent closed/maximal subtrees detected using LCM are displayed in Table 1. The CMTreeMiner fails to produce result at any other support less than 0.9, at which eight closed and one maximal subtree are detected. The difference in number of closed subtrees detected between the two approaches is due to the fact that using the proposed LCM-DSM approach further distinguishes the subtrees based on the exact position of the nodes within the DSM (as explained in

Section 3). The time performance of each algorithm is presented in Fig. 6. The DSM extraction and tree to flat conversion for different support were on average 2.6 and 0.3 seconds, respectively.

Table 1. Number of closed/maximal subtrees using LCM-DSM

Support	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**
Closed	2	5	15	28	60	151	36630	4984965
Maximal	1	2	4	3	5	13	5555	257424

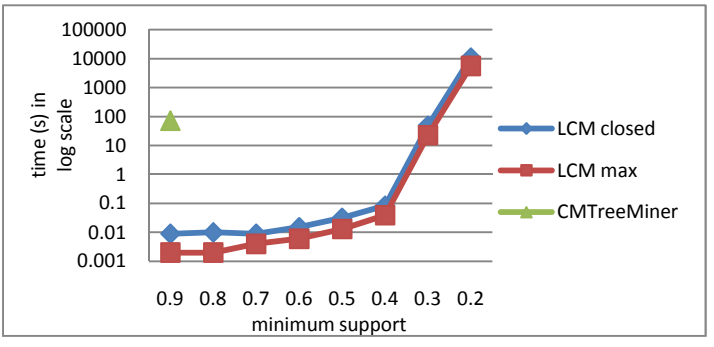


Fig. 6. Time performance

The closed subtree detected using CMTreeMiner at 90% support is shown in Fig. 7. This subtree occurs in 493 instances, and hence the event '*administratief tarief*' and the administered department '*Obstetrics & Gynaecology*' occur in 493 from the total of 506 process instances. Note that, no closed or maximal subtrees contained the nodes keeping the treatment code values '*101*' and '*803*' at support = 90% because they occur in 98 and 408 instances, respectively. Further, for the frequent closed subtree from Fig. 7, it is not certain whether any other event(s) occurred between detected events among the instances and/or whether the additional events differed.

On the other hand the DSM approach can detect subtree patterns at every support level tested. This confirms the capability of the DSM approach in reducing the complexity caused by the structural properties inherent in data, as stated in [10, 12]. Fig. 8 shows a closed frequent subtree pattern containing the node '*treatment code = 101*' extracted from the pattern set when the support is 0.5. This subree is representative for 269 process instances of the '*treatment code = 101*' class in the event log. Knowing that each event contains 7 attributes, we can infer that the event nodes at X_2, X_{10} and X_{18} representing the first three events of the 269 process instances. To find a subtree pattern reflecting the class '*treatment code = 803*' a lower support of at most 18% would need to be used due to this class occurring in only 98 instances. The closed frequent subtree shown in Fig. 9 (shown partially due to space limitation) characterizes 93 process instances with treatment code '*803*', out of a total of 98 instances of that

class. It is noticeable that the subtrees found are more informative than traditional subtrees as the position of each node is preserved, thereby indicating the context of an event or action, and its exact occurrence within a trace. The subtrees detected for the different treatment codes indicate the common process execution path for a particular treatment. However, only some specific activities and characteristics were detected to occur in all process executions, while there was a variation in activities/characteristics of other events. This indicates that there is in reality no global standard in the process execution for a particular treatment and many variations exist. To detect different variations or non-standard process executions, lower minimum support value would be used. At lower support thresholds more subtrees would be detected which reflect common activities/characteristics for subsets of process executions/traces of the treatments. Generally speaking, frequent subtree mining can be used to reveal common/different characteristics of process executions, which depending on the application domain can be useful for exception detection and handling, standardization/optimization of process executions, etc.

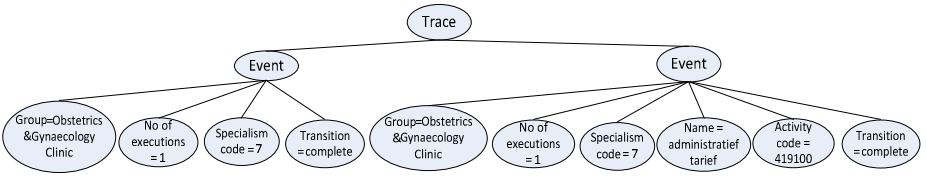


Fig. 7. A frequent closed subtree detected using CMTreeMiner

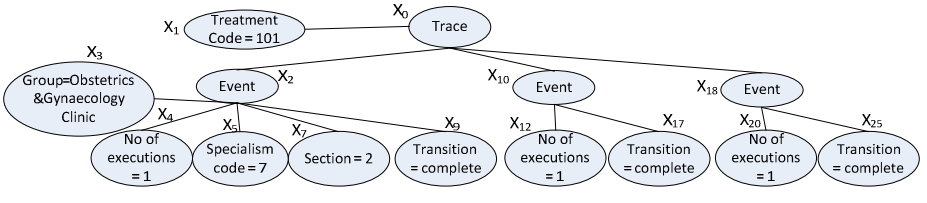


Fig. 8. A frequent closed subtree characterizing treatment code '101'

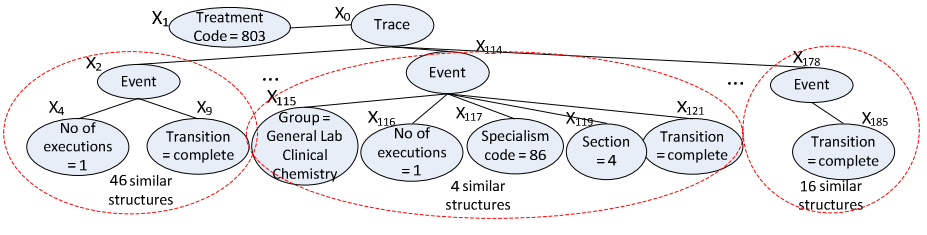


Fig. 9. A frequent closed subtree characterizing treatment code '803'

Decision Tree Learning. The aim is to find the distinguishing characteristics of the process instances having the two *treatment codes* '*101*' and '*803*'. We focus on the application of the C4.5 decision tree learning algorithm [17] and Weka software [18] is used, on the structure preserving flat representation of the tree database. Fig. 10 shows the decision tree learned, which took a total of 0.14 seconds to build, and had 97.4% accuracy evaluated using ten-fold cross validation. The rule characterizing the process instances with treatment code '*803*' is that "if X_{514} = '*event*' and X_{139} = '*group = General Lab Clinical Chemistry*' then the treatment code is '*803*'". This rule correctly classifies 93 process instances and misclassifies one. As the position of the data is incorporated in the result, we know that X_{514} keeps the values of the 65th event of each trace (the two possible values are either = '*event*' or '*NO'*) and X_{139} keeps the values of the second attribute of the 18th event of each trace (there are seven different possible values: '*group=General Lab Clinical Chemistry*', '*group=Radiology*', etc.). It can be observed that the majority of process instances with treatment code '*101*' (403 over 408 total instances) have less than 65 activities. On the other hand most process instances with treatment code '*803*' (93 over 98 total instances) have more than 65 activities. While in this domain, the classification task may not have extended use besides showing the distinguishing characteristics of treatment process executions, learning classification models from data where labels are specifically assigned (e.g. known cases of fraud or undesired outcome) can be extremely useful. The learned model can then be used to predict future cases that are likely to result in an undesired outcome and allow for timely intervention.

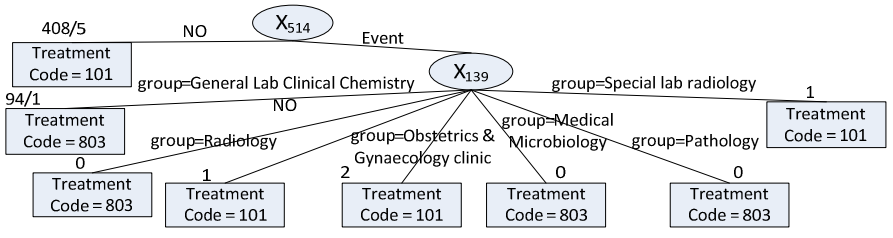


Fig. 10. C4.5 decision tree learned the flat representation of the tree database

5 Conclusion and Future Work

Traditional process mining mainly explores different applications of process models, for example, how they can be discovered/learned, enhanced and conformance checking is often performed. Providing that process logs are increasingly available in XML format, this paper introduces a framework that enables the direct application of a wide range of data mining/analysis methods to tree structured process logs. Two experiments were done on real dataset showing the promising capabilities of this framework. First experiment shows the tree-structured subtree patterns can be discovered at relatively low support thresholds, and the patterns learned are more informative as the position of the nodes (reflecting actions/events in traces) is incorporated in the results.

The second experiment shows how close discriminating characteristics can be directly detected from tree structured process log data. These results demonstrate the potential of the proposed framework to provide an alternative approach to process log analysis. Furthermore given that the proposed framework enables direct application of a wider range of data mining/analysis techniques to tree structured/semi-structured process logs, it expands the pool of available techniques for process log analysis. In the future work we will explore these techniques as well as clustering for their applications to exception detection/analysis/prediction, model conformance checking, and in general discovery of patterns encompassing broad aspects of processes.

References

1. Aguilar-Savén, R.S.: Business process modelling: Review and framework. International Journal of Production Economics 90, 129–149 (2004)
2. van der Aalst, W.M.P.: Process mining: discovery, conformance and enhancement of business processes. Springer, Heidelberg (2011)
3. Rozinat, A., van der Aalst, W.M.P.: Decision Mining in ProM. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 420–425. Springer, Heidelberg (2006)
4. Greco, G., Guzzo, A., Manco, G., Sacca, D.: Mining and Reasoning on Workflows. IEEE Trans. on Knowl. and Data Eng. 17, 519–534 (2005)
5. Günther, C.W., van der Aalst, W.M.P.: A Generic Import Framework for Process Event Logs. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 81–92. Springer, Heidelberg (2006)
6. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES, XESame, and ProM 6. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 60–75. Springer, Heidelberg (2011)
7. Kim, K.: A XML-Based Workflow Event Logging Mechanism for Workflow Mining. In: Shen, H.T., Li, J., Li, M., Ni, J., Wang, W. (eds.) APWeb Workshops 2006. LNCS, vol. 3842, pp. 132–136. Springer, Heidelberg (2006)
8. Gonçalves, M.A., Luo, M., Shen, R., Ali, M.F., Fox, E.A.: An XML Log Standard and Tool for Digital Library Logging Analysis. In: Agosti, M., Thanos, C. (eds.) ECDL 2002. LNCS, vol. 2458, pp. 129–143. Springer, Heidelberg (2002)
9. Hadzic, F., Tan, H., Dillon, T.S.: Mining of Data with Complex Structures. Springer, Heidelberg (2011)
10. Hadzic, F.: A Structure Preserving Flat Data Format Representation for Tree-Structured Data. In: Quality Issues, Measures of Interestingness and Evaluation of Data Mining Models Workshop (QIMIE 2011), Lyon (2011)
11. Kutty, S., Nayak, R., Li, Y.: XML Documents Clustering Using a Tensor Space Model. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part I. LNCS, vol. 6634, pp. 488–499. Springer, Heidelberg (2011)
12. Hadzic, F., Hecker, M., Tagarelli, A.: XML Document Clustering Using Structure-Preserving Flat Representation of XML Content and Structure. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part II. LNCS, vol. 7121, pp. 403–416. Springer, Heidelberg (2011)

13. Kim, H., Kim, S., Weninger, T., Han, J., Abdelzaher, T.: NDPMine: Efficiently Mining Discriminative Numerical Features for Pattern-Based Classification. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part II. LNCS (LNAI), vol. 6322, pp. 35–50. Springer, Heidelberg (2010)
14. Zaki, M.J.: Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications. IEEE Trans. on Knowl. and Data Eng. 17, 1021–1035 (2005)
15. Chi, Y., Yang, Y., Xia, Y., Muntz, R.R.: CMTreeMiner: Mining Both Closed and Maximal Frequent Subtrees. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 63–73. Springer, Heidelberg (2004)
16. Uno, T., Kiyomi, M., Arimura, H.: LCM ver. 2 Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets. In: Goethals, R.J.B., Zaki, B., M.J. (eds.) IEEE ICDM Workshop on Frequent Itemset Mining Implementations. CEUR-WS, Brighton, UK (2004)
17. Quinlan, J.R.: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
18. Holmes, G., Donkin, A., Witten, I.H.: WEKA: a machine learning workbench. In: Second Australian and New Zealand Conference on Intelligent Information Systems, pp. 357–361 (1994)

Face Detection under Illumination Variance Using Combined AdaBoost and Gradientfaces

Joao Paulo Magalhaes, Tsang Ing Ren, and George D.C. Cavalcanti

Center of Informatics, Federal University of Pernambuco, Brazil
{jpm,tir,gdcc}@cin.ufpe.br
http://www.cin.ufpe.br/~viisar

Abstract. Face detection is a research area in computer vision that has received much attention in recent years and a lot of commercial applications were developed using this technology. Even though several methods have been developed, there are still some particular situations that need improvements, especially those related to variations in illumination and face occlusions. In general, illumination problems are handled by using preprocessing, and model or training-based approaches. Here, we propose a face detection method that combines well-known AdaBoost with Gradientfaces technique following a model-based approach, which was not yet used for the face detection problem. We applied Gradientfaces before training an AdaBoost Haar-based cascade classifier to overcome the problem of strong variations in illumination. Quoted approaches were evaluated in two different datasets, containing first artificial and then real illumination problems. Experiments show that proposed method is stable when facing different lighting conditions, and better than others when dealing with strong and uncontrolled illumination problems.

Keywords: Face detection, Gradientfaces, AdaBoost.

1 Introduction

Face detection is the first stage of any computer vision face related technique, such as face recognition, face posing, facial expression analysis, and facial components detection - e.g. eyes, mouth and nose. Face detection can be embedded in a wide range of applications such as tracking systems, intelligent human-computer interaction, surveillance and security systems, content-based image retrieval, and video conferencing.

Over the past years, a variety of methods have been proposed to deal with face detection problem, in particular [1], [2], and [3] are of interest. However, most existing face detection methods has its performance strongly affected by occlusion, and variations in illumination, or pose, mainly if these methods were subject to real images. Phillips et al. argued that pose and illumination variations are two problems that cause a bottleneck for a practical face recognition system [4]. Thus, illumination variation is one of the most significant factor affecting the performance of real life computer vision applications and has received much attention in recent years [5].

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 435–442, 2012.

According to Bo Du et al. [6], model-based approaches are best suited in theory, since it faces some practical problems such as performance and other image constraints or assumptions. Due to the lack of model-based algorithms for face detection, we have combined classical AdaBoost classifier [2, 3] with Gradientfaces [5] in order to reach an illumination insensitive face detector. In our case, instead of using an illumination compensation method, we trained an AdaBoost Haar-based cascade classifier with the Gradientfaces of the input images. So the learning process occurs with illumination insensitive features. In the detection phase, the input images are processed using Gradientfaces prior to detection using trained AdaBoost classifier.

In our experiments, we evaluate different approaches dealing with lighting changes, including: the proposed method as a model-based approach; several preprocessing algorithms, such as gamma intensity correction and different histogram techniques; and a classifier trained with images of faces in different lighting conditions as a training-based approach.

The remaining sections of this paper are organized as follow: in Section 2, a review of the AdaBoost and Gradientfaces techniques is done — also the proposed method which combines both methods is presented; in Section 3, we describe the experiments and results, comparing the detection performance between the three presented approaches; finally, some discussion and conclusion are presented in Section 4.

2 Face Detection Using AdaBoost and Gradientfaces

We have developed a new face detection procedure that uses AdaBoost and Gradientfaces. The idea is to combine the best of both techniques: AdaBoost, as the state-of-art face detection method with interactive learning and simple characteristics that reaches fast processing; and Gradientfaces as a lighting insensitive measure to overcome illumination problems.

Combined AdaBoost and Gradientfaces reaches the same results as convencional AdaBoost in normal lighting conditions and its performance is better than AdaBoost in insufficient or degraded lighting conditions. The procedure is done in a straightforward manner, i.e. it's not necessary a prior knowledge of the image to be processed which demonstrate the lighting insensitive property of Gradientfaces.

In the begining of training phase, all images should be processed using Gradientfaces. Conventional AdaBoost training is then executed using preprocessed images as the input dataset. This way, extracted features are also insensitive to illumination changes. Note that it is not necessary to use illumination degraded images during training phase neither images should be preprocessed to lighting compensation. At runtime, images where faces should be found must also be processed using Gradientfaces before applying AdaBoost detection with obtained features. It is not necessary to process all sub-windows, only the entire image needs to be processed at the beginning of the scanning process to find faces.

2.1 AdaBoost Face Detection

Adaptive Boosting or AdaBoost is a machine learning method, formulated by Yoav Freund and Robert Schapire as described in [7] and [8]. It is used to boost the classification performance of a simple learning algorithm, creating in this way strong classifiers by combining a collection of weak classifiers.

AdaBoost was successfully introduced in computer vision for robust real-time object and face detection using a Haar-based cascade of classifiers by Viola and Jones [2], [3]. Viola and Jones proposals are based on three main ideas: an image representation called integral images which allows very fast computation of the rectangle features; AdaBoost learning of visual features to select a small number of critical features from a very large set; and a cascade of classifiers which allows a coarse to fine search, increasing the performance of the detection.

AdaBoost learning is then used to select key weak classifiers from the set of possible weak classifiers. The final strong classifier takes the form of a perceptron, a weighted combination of weak classifiers followed by a threshold [2], [3].

Cascade of Classifiers. Increased detection performance is achieved through a cascade of classifiers which radically reduce computation time. The stages of the cascade are constructed by training classifiers using AdaBoost. The key insight in this procedure is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances [3].

Within any image a huge number of sub-windows are negative. The structure of the cascade reflects this fact, since it attempts to reject as many negatives as possible at the earliest possible stage. This cascade structure is called attentional cascade and it is based on a coarse-to-fine grained strategy. It is responsible for dramatically increase the speed of the detector by focusing efforts on the promising regions of the image.

2.2 Gradientfaces

Gradientfaces is an illumination insensitive measure extraction method initially proposed by Zhang et al. [5] and applied for face recognition under varying lighting. The method transforms a face image into the gradient domain, this operation generates the Gradientface. Theoretical analysis shows that Gradientfaces is insensitive to illumination and robust to different illumination, including uncontrolled natural lighting conditions [5].

The advantage of the proposed method can be summarized as follows: Gradientfaces is insensitive to illumination changes, including uncontrolled (natural) lighting conditions; Gradientfaces is extracted from the gradient domain, being able to reveal underlying inherent structure of image data; the method is able to apply directly to any single face image neither does it require any prior information on 3-D face shape and many training samples [5].

Based on image reflectance model, a face image $I(x, y)$ is regard as a product $I(x, y) = R(x, y)L(x, y)$, where $R(x, y)$ is the reflectance and $L(x, y)$ is the

Fig. 1. Example of Gradientfaces extracted from [5] showing in the first line five images of one subject from Yale B database, each of them under varying angle lighting sources, and the corresponding Gradientfaces in the second row

luminance at each point *(x, y)* [9]. In order to extract illumination insensitive measure from gradient domain, according to the illumination model, given an arbitrary image *I(x, y)*, Gradientface can be computed as the ratio of y-gradient to x-gradient of *I(x, y)*. That is:

$$G = \arctan\left(\frac{I_{\partial y}}{I_{\partial x}}\right), G \in [0, 2\pi), \tag{1}$$

where $I_{\partial x}$ and $I_{\partial y}$ are the gradient of image *I(x, y)* in the x and y directions, respectively. Arctangent function is used because the ratio of y-gradient to x-gradient of image might be infinitude derived by zero value of x-gradient. Thus, it can not be directly used as the illumination insensitive measure. Fig. 1 shows the original images under different illumination angle sources and the corresponding Gradientfaces.

3 Experiments and Results

We have performed two experiments to confirm our hypothesis using (a) an artificial illumination degraded face dataset and (b) a face dataset with real illumination problems. Our goal was first to validate the proposed method and them to prove the hypothesis in images that contain real illumination problems, while comparing different approaches to deal with lighting problems.

To obtain a correct comparison, the classical AdaBoost from [3] was implemented and trained in equal condition with the same public database used by Viola and Jones. Trainings were done with equal minimum detection and maximum false positive acceptable rates per layer and the same target overall false positive rate, using practical acceptance rates and obtaining approximately equal rates for all validation and test datasets. To confirm this, tests were done using trained classifiers with more than 154,000 images containing no faces and 160 faces were detected, false alarm rate of approximately 1/1,000. With these results, we can

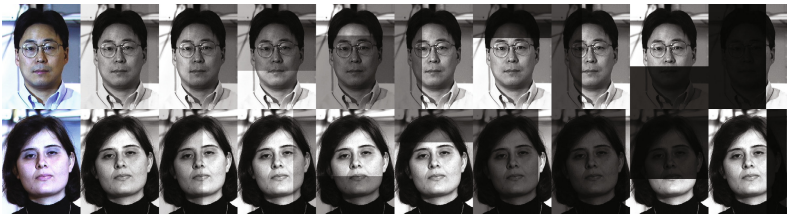


Fig. 2. Examples of two subjects (first and second rows) with vertical or horizontal illumination problems artificially and randomly introduced. First column shows original images extracted from PIE database. Second column shows original images with shadows that degrade 10% the original lighting. Third column shows original images with shadows of 20% of degradation and so on until the last column with shadow of 90% of degradation.

isolate false positive rates and they will not be considered. Two different training were done: directly using face images with lighting problems, generating a basic training-based face detector and using Gradientfaces of the same images, generating a model-based classifier, our combined AdaBoost and Gradientfaces (MB-GF). The basic training was tested in five configurations, without preprocessing input images, that is, only a training-based classifier (TB), and using preprocessing-based techniques: gama intensity correction (PB-GA) with factor of 0.5; histogram equalization (PB-HE); Contrast-limited Adaptive Histogram Equalization (PB-AH), known as CLAHE; and histogram adjustment (PB-HA) — using the intensity values that represent the bottom 1% and the top 1% of the range as the adjustment limits.

The first tests were done using a database of 4,916 faces with shadows artificially introduced. In these images, shadow regions were made horizontally or vertically, with locations and sizes randomly chosen. We have generated nine sets of 4,916 images containing all original images with problems from 10% to 90% in the illumination level. A degradation of 10% means pixels in the shadow area had their values reduced by 10%. And so on, until an image with degradation of 90% meaning pixels in the shadow area had their values reduced by 90%. E.g. if we apply a 100% degradation, the pixels under shadow area would be all equal to zero and a degradation of 0% would not change pixel values in the shadow area. Fig. 2 shows examples of the resulting images from artificial test base.

Fig. 3 shows comparative results between the techniques. As can be seen, in normal conditions, that is, with no lighting degradation, most of techniques obtained approximately the same result, only the adaptive histogram preprocessing-based technique (PB-AH) have caused a diminish in detection rate. When the degradation is increased, we can observe different behavior of the studied approaches. Training based approach (TB) has its performance linearly affected by lighting degradation, going from more than 90% without lighting degradation to less than 30% in the higher degradation level. Preprocessing-based techniques showed a varying behavior. The performance of (PB-AH), (PB-HA), and

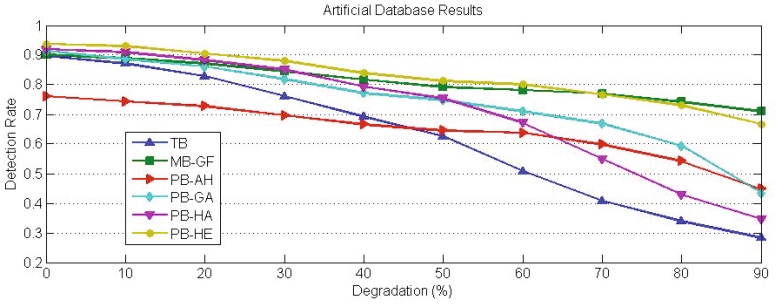


Fig. 3. Results comparing classical AdaBoost and the proposed AdaBoost with Gradientfaces

(PB-GA) is strongly affected with lighting degradation levels greater than 50%. In the other hand, the simple histogram equalization (PB-HE) and the proposed (MB-GF) present good and stable results in all levels for this type of degradation. The overall difference between all techniques increases significantly from 0.59% with degradation of 10% (excluding (PB-AH) result) to 42.68% with degradation level of 90%. The difference in performance is accentuated especially when degradation exceeds 50%. We've considered these results expressive even with an artificial test base.

As initial experiments showed promising results, we conduct further experiments this time using images with real lighting problems. Images were obtained from the PIE and Yale databases. In the total, 315 images of 15 subjects have been collected from the PIE database, each containing a person with the face approximately frontal and with lighting problems varying from side shadows to excessive frontal lighting. A total of 45 images of 15 subjects have been collected from Yale database, containing 3 images of each subject, one with normal frontal illumination and 2 with side (left and right) lighting effects. Fig. 4 shows some examples of images from these datasets.

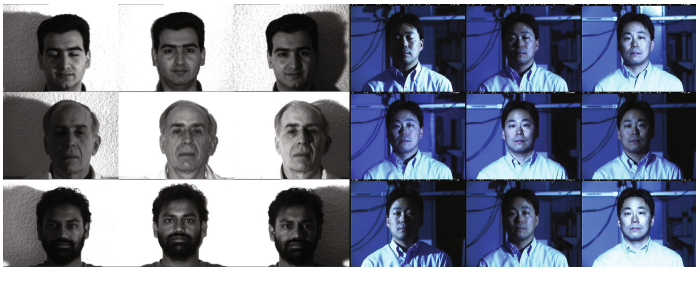


Fig. 4. Examples of images with real illumination problems from Yale and PIE databases. On the left column, it is shown 3 people from the Yale database, each of them with left, centered and right lighting. On the right column, it is shown a person from PIE database with great lighting variability.

Table 1 shows the results obtained in these datasets. As can be seen, proposed method (MB-GF) is better than others when applied in both Yale and PIE databases. It can be observed also that all other methods present unstable results, with the training-based technique (TB) being the worst in both. Histogram-equalization (PB-HE) continues presenting stable results, even being inferior to the best ones. Other techniques present a wide variation when subjected to variations in lighting even with gama intensity correction achieving a good result in Yale base. Proposed method remained stable and achieved the best results, superior to 80%. In general, Yale results are better than PIE results because PIE base has more uncontrolled lighting variations and complex background.

Table 1. Comparison results on Yale and PIE databases

Method	PIE	Yale
TB	51.91%	61.36%
MB-GF	**80.57%**	**84.09%**
PB-AH	53,50%	68.18%
PB-GA	58.59%	**84.09%**
PB-HA	70.38%	59.09%
PB-HE	71.65%	77.27%

4 Conclusion and Discussion

We have described a new face detection method for images that present illumination problems. The procedure combines classical Adaboost with Gradientefaces technique. Experiments were performed in images with artificial and real illumination problems, testing different approaches dealing with illumination changes. Results have shown training-based approach presenting poor results, preprocessing approach presenting varying results, and model based approach showing more stable and better results. Proposed method is more robust when dealing with uncontrolled lighting conditions and it is proved to be superior compared with other approaches.

An important characteristic of this method is that Gradientfaces also works well when modeling problems of excessive lighting. This can be observed in the results obtained from the PIE database, since it contains both images with shadows and images containing faces with excessive lighting. Although this was not investigated in detail, since here we focused our attention in solving the problem where face images are occluded by shadows.

As future work, we can study the insertion of Gradientfaces into the rectangle features, excluding the need to process normal images before AdaBoost training and detection. And, based on test results, the reasonable behavior of the technique for cases of up to 90% of lighting degradation, it can still be considered its use to model the problem of face detection considering occlusions of the face.

References

1. Roth, D., hsuan Yang, M., Ahuja, N.: A snow-based face detector. In: Advances in Neural Information Processing Systems, vol. 12, pp. 855–861. MIT Press (2000)
2. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. I–511–I–518 (2001)
3. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57, 137–154 (2004)
4. Phillips, P., Moon, H., Rizvi, S., Rauss, P.: The feret evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)
5. Zhang, T., Tang, Y.Y., Fang, B., Shang, Z., Liu, X.: Face recognition under varying illumination using gradientfaces. Trans. Img. Proc. 18(11), 2599–2606 (2009)
6. Du, B., Shan, S., Qing, L., Gao, W.: Empirical comparisons of several preprocessing methods for illumination insensitive face recognition. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005), pp. ii/981 – ii/984 (March 2005)
7. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: European Conference on Computational Learning Theory, pp. 23–37 (1995)
8. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting (1997)
9. Horn, B.K.P.: Robot Vision. MIT electrical engineering and computer science series. MIT Press, Cambridge (1986)

Iris Segmentation and Recognition Using 2D Log-Gabor Filters

Carlos A.C.M. Bastos*, Tsang Ing Ren, and George D.C. Cavalcanti

Center of Informatics, Federal University of Pernambuco – CIn/UFPE
Recife, PE, Brazil
{cacmb,tir,gdcc}@cin.ufpe.br
www.cin.ufpe.br/~viisar

Abstract. This paper describes an analysis on the parameters used to construct 2D log-Gabor filters to encode iris patterns. An iris recognition system, composed by segmentation, normalization, encoding and matching is also described. The segmentation module combines the Pulling & Pushing and Active Contour Model and the Circular Hough Transform to find the inner and the outter boundaries of the iris. The experiments were performed using the CASIA v.1 iris database and the results are analyzed using ROC curves. They showed that 2D log-Gabor filters are also an effective alternative to encode the features present on iris patterns.

1 Introduction

This work presents an analysis of the parameters used to construct 2D log-Gabor filters to encode iris patterns. Despite previous use of this filter by Du [2] and Yao et al. [3] their work only presents the recognition rates without make any consideration about the values of the parameters used to build the filter. The 2D log-Gabor is a complex filter composed by four parameters. Choosing the best parameter combination to some application (like the iris recognition) is a challenging task due to the variety of possibilities that can be created.

This work also describes an iris recognition system composed by segmentation, normalization, and encoding and matching, as described in the following sections. The CASIA v.1 [9] iris image database is used for the experiments. The CASIA database contains a total of 756 images (grayscale, 8-bit), obtained from seven samples for each of the 108 individuals. The image acquisition is divided in two sessions, and the images are obtained using a near infrared illumination.

The remainder of this paper is organized as follows: Section 2 presents the combined Pulling & Pushing and Active Contour Model [4] and the Circular Hough Transform for the segmentation procedure. In Section 3 the normalization process based on the rubber sheet model is described. Section 4 is dedicated to the encoding and matching procedure. Experimental results are shown in Section 5 and Section 6 some concluding remarks and ideas for future works are presented.

* This work was partially supported by FACEPE and CAPES.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 443–450, 2012.

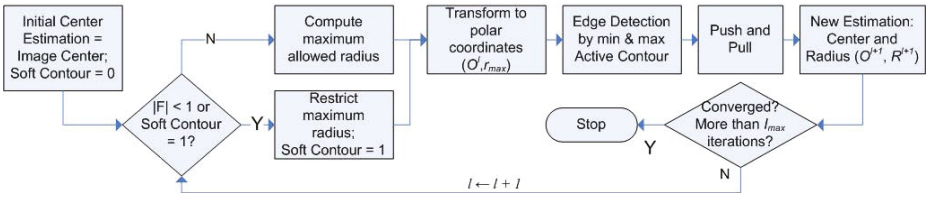


Fig. 1. Flowchart of the PP–AC method

2 Pupil and Iris Segmentation

The iris region, shaped as a circular ring, can be approximated by two circles: one for the iris/sclera boundary and another for the iris/pupil, which is inside the first region. However, it is not always possible to assume that the two circles are concentric. Another difficult, in precisely segment the iris from the complete image, lies in the fact that the eyelids and eyelashes normally produce occlusion on the upper and lower part of the iris. Also, specular reflections can corrupt the iris texture. A successful segmentation technique must identify and correct these issues.

The segmentation procedure combines two approachs: one to determine the pupil/iris borders and another to determine iris/sclera borders. For the first case the Pulling & Pushing and Active Contour Model is used. In the second case the borders are found using the Circular Hough Transform.

The Pulling & Pushing & Active Contour Model [4] is a method that combines a spring-force driven iterative scheme with active contour models. It uses a combination of N mass-less springs joined at a common point O', to estimate the center $O_p(x_p,y_p)$ and radius R_p parameters. The PP–AC flowchart is shown in Figure 1. For more details about PP–AC method, please refer to [4].

The segmentation procedure for the iris/sclera boundary uses a circular Hough transform to evaluate the center coordinates (x_c, y_c) and radius (r) of the iris. First, a histogram equalization pre-processing is used in the identification of the iris/sclera boundary. An edge map is then generated by using the Canny edge detector modified to output only the vertical gradient. At the edge map image we applied a grouping algorithm (8-connected) to remove noise segments from the image. This step is necessary due to the significant increase in the number of edge segments in the image obtained after the histogram equalization procedure. These procedures reduce the number of edge points and, consequently, the computational cost due to the use of the circular Hough transform.

Another strategy to reduce the computational cost of this technique is to restrict the range of the radius. For CASIA v1 database the values for the iris radius range from 80 to 140 pixels. Using the procedures described above, 719 (95.10%) images were found to be correctly (both pupil and iris) segmented.

The eyelid detection is obtained by performing a linear Hough transform (using only horizontal gradient) to localize a line which best fits the iris/sclera upper (or lower) area. Once this line is found, a second horizontal line, which

intersects with the first at the iris edge that is closest to the pupil, is defined. This method is applied for both upper and lower eyelids and allows maximum isolation of the occluded region [5]. Eyelashes are detected by a threshold technique. Eyelids, eyelashes and reflections are undesirable in the iris images template and they should be detected in order to not corrupt the iris patterns, which are used in the matching stage.

3 Normalization

Normalization is the stage where the previously detected iris region is transformed to have fixed dimensions so that two images of the same iris, even if they are acquired on different conditions, have the discriminating features in the same spatial location. Therefore comparisons can be made in a consistent manner. The applied normalization process is based on Daugman's rubber sheet model [1], which deals with contractions and dilations of the pupil to create a uniform representation in polar coordinates (r, θ) with r between $[0, 1]$ and θ between $[0, 2\pi]$. The transformation from the Cartesian coordinates to polar coordinates is performed through the relationship [1]: $I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta)$, with: $x(r,\theta) = (1-r)x_p(\theta) + rx_I(\theta)$; $\quad y(r,\theta) = (1-r)y_p(\theta) + ry_I(\theta)$, where $I(x, y)$ is the region of the iris image in Cartesian coordinates, $I(r,\theta)$ is the image in polar coordinates, (x_p, y_p) and (x_I, y_I) are the pupil and iris boundary points, respectively, at θ direction. The pupil center is considered the reference point for the coordinate system.

From the pupil center until the iris boundary, radial lines are drawn. The number of data points in each radial line defines the radial resolution. Similarly, the number of lines defines the angular resolution. In this work, we used 20 radial points and 240 angular lines, leading to a rectangular representation of 20 x 240 pixels size.

4 Encoding and Matching

Encoding is the stage that extracts the iris characteristics, to generate a compact representation called template. In order to create distinct representation only the most discriminating characteristics should be encoded, in this way an accurate recognition rate can be obtained. There is a correspondent matching metric related to the encoding method. This measure should inform the similarity degree between two representations. From this metric two intervals are expected: one for comparing images of the same eye (intra-class comparisons) and another for images of different eyes (inter-class comparisons). In order to obtain decisions with a high degree of confidence, those intervals must be as distinct and separate as possible.

Traditionally Gabor filters are chosen to encode iris patterns because they offer the best simultaneous localization of spatial and frequency information. However it is not possible to generate such filters with an arbitrary wide bandwidth, without taking in consideration the effects of the DC component, present in

the even-symmetric filter. Alternatively, log-Gabor filters, by definition, have no DC component and can be generated with an arbitrary bandwidth which can be optimized to produce filters with small spatial dimensions [6]. Even though some authors have shown results of iris recognition system based on 2D Log-Gabor filters [2,3], the results are very limited and not clearly concerning several parameter used. Here we proposed a more thoughtful analysis of this method.

Field [7] suggested that natural images are better encoded by filters that have Gaussian transfer functions when viewed on the logarithm frequency scale. Such filter has two important characteristics: no DC component and transfer function extended by a large range in frequency, which makes possible the use of large bandwidth to capture information.

Due to the limitation of log function at origin, Log-Gabor filters are built in the frequency domain. In a polar coordinate system, 2D log-Gabor filters can be separated in two components: radial and angular, and it is defined by the equation: $G(\omega,\theta) = \exp\left\{-\frac{[\log(\omega/\omega_0)]^2}{2\cdot[\log(\sigma_\omega)]^2}\right\}\cdot\exp\left\{-\frac{(\theta-\theta_0)^2}{2\cdot\sigma_\theta^2}\right\}$, where ω_0 is the central frequency, θ_0 is the orientation angle, σ_r and σ_θ are the radial and angular bandwidth, respectively. Kovesi [6] stated that the values 0.74, 0.55 and 0.41 for σ_r would produce filters bandwidth with one, two and three octaves, respectively.

To encode an iris pattern, only phase information is extracted, projecting the filtered phasor over the complex plane formed by the filter. In this process two bits per projection are acquired, and it is represented by: $h_{\{\mathrm{Re,Im}\}} = sgn_{\{\mathrm{Re,Im}\}}\left\{\int_\rho\int_\theta I(r,\theta) * lg(r,\theta)drd\theta\right\}$, where $I(r,\theta)$ is the image in polar coordinates, $lg(r,\theta)$ is the inverse Fourier transform of the filter.

In this process 9600 bits (240x20x2) are encoded. An equal number of bits are assigned to the noise mask, which mark areas such as eyelids, eyelashes, occlusions and reflections detected in the segmentation stage. Those areas should not be considered when matching iris patterns.

The comparison between two templates is performed by calculating the Hamming distance of the templates. The Hamming distance is given by: $HD = \frac{\|(templateA \oplus templateB) \cap maskA \cap maskB\|}{\|maskA \cap maskB\|}$, where $\{templateA, templateB\}$ are the phase bits of two iris and $\{maskA, maskB\}$ are the corresponding noise masks. This metric defines the degree of dissimilarity between two bit strings. The denominator counts the number of bits that are effectively used to the match. Translation and scale invariance are obtained during the normalization stage. Rotational invariance is then achieved by shifting the bits of a template to both left and right directions. This procedure is suggested by Daugman [1] and corrects misalignments present in the normalized pattern.

5 Experimental Results

For each correctly segmented image, a template and its corresponding noise mask are generated using a 2D Log-Gabor filter. Then, the encoded templates are compared 2 by 2. These comparisons are divided into two groups (one for intra-class comparisons and another to inter-class comparisons) and consist in

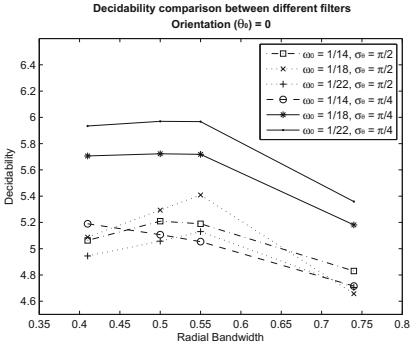


Fig. 2. Decidability variation as a function of the angular bandwidth using only the filters that have $\theta_0 = 0$

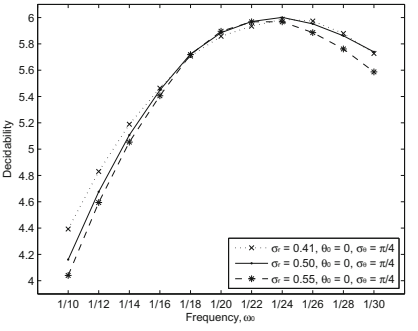


Fig. 3. Decidability variation as a function of the central frequency for the second investigation phase

calculating the Hamming distance between the templates and $\pm$ 8 shifts. At each shift, a new Hamming distance is calculated and only the lowest value is selected. With the correctly segmented images, 2067 comparisons belongs to the intra-class group and 256054 comparisons belongs to the inter-class group.

The decidability is the criterion used to estimate the performance of the encoding scheme and thus to choose the best combination of parameters for the log-2D Gabor filter. This metric shows the degree of separation between the two classes (as intra an inter-class groups) and it is computed using: $d' = \left(|\mu_s - \mu_d| \, / \sqrt{\frac{\sigma_s^2 - \sigma_d^2}{2}} \right)$ where μ_s and μ_d are the mean values each distribution and σ_s, σ_d its standard deviation.

The best parameter determination was split in three steps. First we selected some basic parameters, and from them, we generated multiple combinations. In the initial investigation we include three values for the center frequency (ω_0): 1/14; 1/18; 1/22; four points for the radial bandwidth (σ_r): 0.41; 0.50; 0.55; 0.74; four values for the orientation (θ_0): $-\pi/4$; 0; $\pi/4$; $\pi/2$; and two values for the angular bandwidth (σ_θ): $\pi/4$; $\pi/2$. The combination of all parameters generates a total of 96 different filters. The values chosen for σ_r corresponds to a bandwidth of one, two, little more than two and three octaves. They were selected to verify the ability of the 2D log-Gabor filter in to capture information from a wide range in frequency.

The overall analysis of the first 96 filters showed that the best orientation is $\theta_0 = 0$ radians. This indicates that the regions of the frequency spectrum that possess orientation near zero contain the most discriminating information. Another important point is that the combinations wich used radial bandwidth of one octave (0.74) have obtained, in most cases, lower debidalibity values.

The combinations that showed the best results in terms of radial bandwidth were those that captured information in two and little more than two octaves

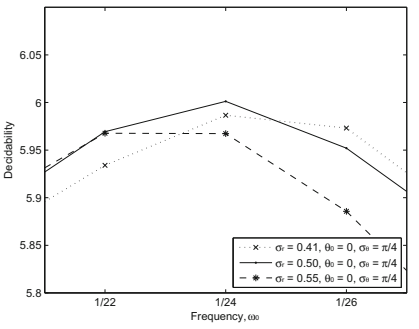


Fig. 4. Zoom of the region from Figure 3

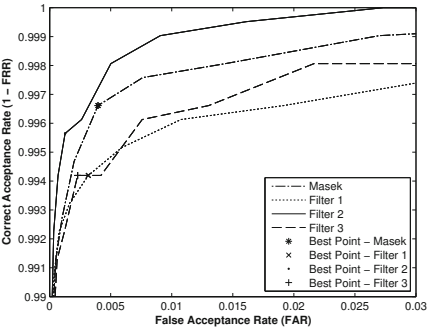


Fig. 5. ROC curve showing the performance comparison among different filters

(σ_r = 0.55 and 0.50). Those results can be confirmed by observing Figure 2, which shows the variation of decidability as a function of the radial bandwidth. This graph was constructed using only the combinations of filters that had $\theta_0 = 0$, being grouped by the center frequency and radial bandwidth. It is also possible to note that $\sigma_\theta = \pi/4$ is the best value for the angular bandwidth.

With the definition of the parameters $\theta_0 = 0$ and $\sigma_\theta = \pi/4$ the second investigation was initiated. It was the search for the best values for the radial bandwidth and for the center frequency. Due to inferior results obtained by the combinations that used $\sigma_r = 0.74$, this value was not used. So it continued only with the values 0.41, 0.50 and 0.55, for frequencies ranging between 1/10 and 1/30. This led to thirty-three new filter combinations to be analysed. These results can be viewed in Figure 3 that shows the variation of decidability due to changes in center frequency, grouping the curves by the radial bandwidth. Through this Figure it can be seen that the peaks of each curve occur close to the other curves. To help visualization, Figure 4 presents a zoom of this region.

Based on the values of the decidability, three filters were selected (one for each radial bandwidth) to do the performance evaluation of the entire iris recognition system. The values of decidability of the selected filters as well as their parameters are displayed in Table 1.

The three filters are then compared with the encoding method proposed by Masek [5]: an 1D log-Gabor filter. The values $\omega_0 = 1/18$ and $\sigma_r = 0.50$, used to construct this filter represents the best combination of parameters found by Masek for the Casia v1 database. In tests conducted in our work this filter has reached a decidability $d' = 5.5484$.

The Table 1 shows the values of decidability, mean and standard deviation of intra and inter classes for the four selected filters. The three log-2D Gabor filters present mean and standard deviation values close to each other. The greatest separation between the means of intra and inter groups occurs for the Filter 2 and the smaller separation occurs for the filter proposed by Masek. However, Masek's Filter has the lowest standard deviation in inter-class, indicating that their data is more condensed than those generated through the filter log-Gabor

Table 1. Parameters, decidability, mean and standard deviation values of intra and inter groups for the filters used in evaluation

Filter	Parameters	d'	μ_{Intra}	σ_{intra}	μ_{Inter}	σ_{inter}
Masek	$\omega_0 = 1/18; \sigma_r = 0,50$	5,5484	0,2915	0,0433	0,4701	0,0140
Filter 1	$\omega_0 = 1/24; \sigma_r = 0,41; \theta_0 = 0; \sigma_\theta = \pi/4$	5,9866	0,2563	0,0412	0,4490	0,0194
Filter 2	$\omega_0 = 1/24; \sigma_r = 0,50; \theta_0 = 0; \sigma_\theta = \pi/4$	6,0012	0,2455	0,0433	0,4479	0,0200
Filter 3	$\omega_0 = 1/22; \sigma_r = 0,55; \theta_0 = 0; \sigma_\theta = \pi/4$	5,9677	0,2484	0,0443	0,4505	0,0182

Table 2. CAR, FAR, EER, and threshold values for the operating points highlighted in Figure 5

Filter	Threshold	CAR (%)	FAR(%)	EER(%)
Masek	0,425	99,6613	0,3964	0,3964
Filter 1	0,395	99,4194	0,3167	0,5921
Filter 2	0,385	99,5646	0,1261	0,3870
Filter 3	0,385	99,4194	0,2296	0,5806

2D. This feature influences the values of the False Acceptance Rate and Correct Acceptance Rate.

The third step investigates the hability of each filter in encoding the iris patterns. This evaluation was performed by comparing the recognition accuracy with the aid of the ROC curve. The values of this curve are determined by calculating the rates of genuine correctly accepted (CAR) and imposters incorrectly accepted (FAR) for each threshold value chosen. Figure 5 presents the ROC curve for the four filters. Nothe that the best operating points appear highlighted and they were determined by the smaller Euclidean distance in relation to the ideal point (CAR = 1 and FAR = 0).

By examining Figure 5 it is possible to note that the filter proposed by Masek and Filter 2 have better performance because its curves are closer to the ideal value. Table 2 shows in more detail the comparison between the four filters, highlighting the rates of correctly accepted users and imposters incorrectly accepted computed in its best operating point. The Table 2 also show the threshold values and, in the last column, presents the equal error rate (EER).

A careful observation of the FAR and EER rates for the filter proposed by Masek reveals that the best operating point of the system is equal to the point where there is the EER. As for the second filter the operating point is below the point of EER. Filters 1 and 3 despite having the same CAR, they differ in FAR because of the slightest separation between the means achieved by Filter 1 (see Table 1).

Despite some filters present a greater decidability, this fact does not guarantee that such filters will necessarily have a better recognition rate. Filters 1 and 3 have decidability larger than the filter proposed by Masek, yet it has the highest correct acceptance rate. However, for most security systems is more important to have a low false acceptance rate at the expense of a high correct acceptance rate, which makes Filter 2 the best combination to encode iris patterns, in this work.

6 Conclusions

This work presented an analysis of the parameters used to construct 2D log-Gabor filters to encode iris patterns. The combined segmentation technique showed good results archieving more than 95% of correct segmentation.

For the encoding process 2D Log-Gabor filters are used and 129 filters were evaluated to identify the best parameter combination. First we have defined a set of base parameter values from wich 96 combinations were created. As a result it was possible to identify the best values for orientation θ_0 and for the angular bandwidth σ_θ. Altought, to identify the best parameters for the center frequency and for the radial bandwidth 33 new filters were generated. The best three filters, according to the decidability value, were selected and then compared with a 1D log-Gabor filter to evaluate the reconition performance.

The 2D log-Gabor filter proved to be more efficiency in capturing global characteristics, resulting in a system with lower false acceptance rates. Therefore, 2D Log-Gabor filter is an effective alternative to encode the features present on the iris textures. The CASIA v1 database was used to verify the proposed method, and very high recognition rates were obtained. Even though, several researchers do not consider the CASIA v.1 as an interesting database to validate iris recognition system due to the pre-processing of the images [8]. However this pre-processing have affected only the pupil area, leaving the iris texture pattern without artificial modification, enabling texture analysis be performed.

References

1. Daugman, J.: How Iris Recognition Works. IEEE Trans. Circuits and Syst. for Video Tech. 14(1), 21–30 (2004)
2. Du, Y.: Using 2D-Log Gabor Spatial Filters for Iris Recognition. In: Proceedings of SPIE, vol. 6202, pp. 62020: F1–62020: F8 (2006)
3. Yao, P., Li, J., Ye, X., Zhuang, Z., Li, B.: Iris Recognition Algorithm Using Modified Log-Gabor Filters. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 4, pp. 461–464 (2006)
4. Carlos, A.C.M., Bastos, T.I., Ren, G.D., Cavalvanti, C.: A combined Pulling & Pushing and Active Contour model for pupil segmentation. In: IEEE Int. Conf. on Acoustics, Speech and Signal Proc., pp. 850–853 (2010)
5. Masek, L.: Recognition of human iris patterns for biometric identification, The University of Western Australia (2003)
6. Kovesi, P.: Image Features From Phase Congruency. Videre: A Journal of Computer Vision Research 1(3) (1999)
7. Field, D.: Relations between the statistics of natural images and the response properties of cortical cells. Journal of the Optical Society of America 4(12), 2379–2394 (1987)
8. Phillips, P.J., Bowyer, K.W., Flynn, P.J.: Comments on the CASIA Version 1.0 Iris Data Set. IEEE Transaction on Pattern Analysis and Machine Intelligence 29(10) (2007)
9. CASIA-Iris V1 (2009), http://www.cbsr.ia.ac.cn/IrisDatabase

A Hybrid GMM Speaker Verification System for Mobile Devices in Variable Environments

Tsang Ing Ren, George D.C. Cavalcanti,
Dimas Gabriel, and Hector N.B. Pinheiro

Center of Informatics, Federal University of Pernambuco, Brazil
{tir,gdcc,dg,hnbp}@cin.ufpe.br
http://www.cin.ufpe.br/~viisar

Abstract. This paper proposes a hybrid GMM speaker verification system for mobile devices in variable environments. A neural network based on backprobagation learning and a kNN algorithm were used to determine the probabilities in the GMM models to verify the speaker from a voice signal. A Voice Activity Detection (VDA) algorithm was also used to improve the voice identification ratio. The proposed method was tested using a database obtained from several different mobile devices in used in different environments, showing the robustness of the system.

Keywords: Speaker verification, Gaussian mixture model, mobile devices.

1 Introduction

In the last few years, the way in which the information is transmitted has changed. From the advance of the Internet to portable devices and the presence of technology in all places, security has become a priority to ensure that personal data and private places can only be accessed by authorized people. Shopping with a credit card, accessing restricted areas or resources, security transactions and automobile voice lock are just some examples in which it is necessary to verify the identity of the user.

Speaker recognition is the process of automatically recognizing who is speaking by using speaker-specific information included in the voice waves [6]. Traditionally, recognition can be classified into speaker identification and speaker verification. Identification is the process of determining from which of the registered speakers a given utterance comes. Verification is the process of accepting or rejecting the identity claim of a speaker. Another classification used is in respect of the utterance spoken by the speaker: the recognition can be dependent or independent of a specific text.

One of the most important advantages of voice biometrics is that the voice signal can be captured, with quality, by almost any device that contains a microphone. This means that the authentication process can be conducted in any place with, for example, a mobile phone or device. Thus, the system must deal with uncontrolled and unknown environments. Mobile devices are often used in

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 451–458, 2012.

highly variable acoustic environments such as quiet offices, busy cafeterias, or loud street intersection.

This paper proposes a text-independent Automatic Speaker Verification (ASV) system applied to a highly variable environments and with different capture devices. In our experiments, a classical GMM recognition system and two hybrid GMM-based system applied with a Voice Activity Detection (VAD) algorithm were evaluated.

In the next section, the theoretical background of the used techniques to develop the systems are discussed. This includes the VAD algorithm, Gaussian Mixture Models (GMM), Multilayer Backpropagation Perceptron Neural Networks (MLP-NN) and an Nearest Neighbors techniques (kNN). In Section 3, the MIT Mobile Device Speaker Verification Corpus (MIT-MDSVC) are described. These corpus are designed to represent the scenario in which voices were captured using small handheld devices varying both the environment and the microphones. Section 4 presents the experiments and the results of the proposed system using the MIT-MDSVC corpora. Finally, conclusions and future directions are given in Section 5.

2 Theoretical Background

2.1 Voice Activity Detection

Rabiner and Sambur [5] proposed an algorithm to voice activity detection that is based on measurements of energy and zero crossing rate. The main idea of this algorithm is the following: i) divide the speech signal into frames; ii) calculate the energy and the zero crossing rate for each frame; iii) compare with pre-established thresholds, iv) define the endpoints of each word. These thresholds define the characteristic values of the parameters for silence or background noise. The start of the recording interval is considered to have only silence or noise signal. Initially the energy values of each frame are compared with the thresholds and candidates qualified to be the endpoints of the word are defined. Next, the zero crossing rate is analyzed in an interval of S frames from the points candidate to be the start and the end of the voice signal to improve the location of the words that begin or end with fricative consonants.

Gaussian Mixture Models. Gaussian Mixture Models (GMMs) have been widely used in density modeling and clustering. In speaker recognition, they are used to model the distribution of feature vectors of speaker utterances. Generally, the system has one GMM for each speaker. GMMs are considered to have a universal approximation ability because they can closely model any density function provided that it contain enough mixture components [8].

Let $X = \{x_1, x_2, ..., x_N\}$ be a data set of N vectors x_n, each having a d-dimensional feature vector extracted from a speaker utterances. Since the distribution of these vectors is unknown, a Gaussian mixture can be used to approximately model this distribution. The GMM is a weighted sum of M mixture components of multivariate Gaussians, each one represented by

$$N(\mathbf{x}_n; \mu; \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}_n - \mu)^T \Sigma^{-1} (\mathbf{x}_n - \mu)}, \tag{1}$$

where μ is the mean vector, and Σ is the covariance matrix. The GMM can be represented as

$$p(\mathbf{x}) = \sum_{m=1}^{M} w_m N(\mathbf{x}; \mu_m, \Sigma_m), \tag{2}$$

having the component weight defined as $\sum_{m=1}^{M} w_m = 1, w_m > 0$.

In the training phase, the GMM parameters are estimated such that they best match the distribution of the training vectors. The most popular estimation method is the maximum likelihood (ML) estimation, generally executed with the Expectation-Maximization (EM) algorithm [1]. For a sequence of training vectors X, the current likelihood of the GMM is

$$p(X|\lambda) = \prod_{n=1}^{N} p(\mathbf{x}_n|\lambda), \tag{3}$$

where λ is the parameter model, μ and Σ, of the Gaussian distribution. The aim of ML estimation is to obtain a new parameter model $\bar{\lambda}$ such that $p(X|\bar{\lambda}) > p(X|\lambda)$. Maximizing this function directly is not easy and is computationally expensive, hence an auxiliary function Q is used

$$Q(\lambda, \bar{\lambda}) = \sum_{n=1}^{N} p(i|\mathbf{x}_n, \lambda) log[\bar{p}_i N(\mathbf{x}_n, \bar{\mu}_i, \bar{\Sigma}_i)], \tag{4}$$

where $p(i|\mathbf{x}_n, \lambda)$ is the posteriori probability for the component $i, i = 1, ..., w$ and satisfies

$$p(i|\mathbf{x}_n, \lambda) = \frac{p_i N(\mathbf{x}_n, \mu_i, \Sigma_i)}{\sum_{k=1}^{w} P_k N(\mathbf{x}_n, \mu_k, \Sigma_k)}. \tag{5}$$

The maximization of the Q function is performed using the EM algorithm. The basics of this algorithm is: if $Q(\lambda, \bar{\lambda}) \geq Q(\lambda, \lambda)$ then $p(X|\bar{\lambda}) \geq p(X|\lambda)$ [6].

EM is an iterative algorithm that consist of two steps: the expectation step and the maximization step. The aim of the expectation step is to find the value of the likelihood, $p(X|\lambda)$, assuming the current values of λ for the GMM. In the maximization step, the values of λ (μ, Σ and the component weight, W) for all Gaussian component i in the GMM are updated to $\bar{\lambda}$ such that

$$\bar{\mu}_i = \frac{\sum_{n=1}^{N} p(i|\mathbf{x}_n, \lambda) \mathbf{x}_n}{\sum_{n=1}^{N} p(i|\mathbf{x}_n, \lambda)}, \tag{6}$$

$$\bar{\Sigma}_i = \frac{\sum_{n=1}^{N} p(i|\mathbf{x}_n, \lambda)(\mathbf{x}_n - \bar{\mu_i})(\mathbf{x}_n - \bar{\mu_i})'}{\sum_{n=1}^{N} p(i|\mathbf{x}_n, \lambda)}, \tag{7}$$

$$\bar{W}_i = \frac{1}{N} \sum_{n=1}^{N} p(i|\mathbf{x}_n, \lambda). \tag{8}$$

The process of training a GMM is described in four steps as follows:

Step 1: Generate the first posteriori probability $p(i|\mathbf{x}_n, \lambda)$ following equation (5) and using random values for λ;
Step 2: Compute a new $\bar{\lambda}$ by updating the values of μ, Σ and W using equations (6), (7) and (8), respectively;
Step 3: Update the posteriori probability $p(i|\mathbf{x}_n, \lambda)$ using the new values of λ computed in Step 2;
Step 4: Compute the auxiliary Q function following equation (4). Stop the training process if the increase in the value of Q at the current iteration relative to the value of Q at the previous iteration is below a chosen threshold, otherwise go to the next iteration starting at Step 2.

Multilayer Perceptron Neural Networks. Multilayer Perceptrons (MLP) are neural networks with one or more hidden layers [4]. Each layer is composed of set of neurons. Two neurons belonging to adjacent layers are connected by directed arrows. This connection has a weight that ponders its degree of importance.

One method to adjust the weights is the Backpropagation algorithm, which is a supervised learning algorithm by error correction. The Backpropagation algorithm is a generalization of the least squares algorithm and it aims to minimize the root mean squared error between the current network output and the desired output through gradient technique [3]. It consists of:

Step 1 - Propagation: After presenting the input pattern, the response of an unit is propagated as input to the next units in the next layer until the output layer, where the network response is obtained and the error is calculated.
Step 2 - Backpropagation: The synaptic weights are updated based on the difference between the output of the network and the desired output.

2.2 k-Nearest Neighbor

The k-Nearest Neighbor algorithm or kNN is a supervised lazy learning algorithm. The general idea of this algorithm is to find k labeled neighbors closest to the query element. After that, the class of the query element is equal to the most frequent class among the k neighbors. Given a database with labeled elements and a new element to be classified, the following steps are executed by the kNN algorithm:

Step 1: Calculate the distance from the element to be classified to each element in the reference database. Several similarity and dissimilarity measures can be used to calculate the distance measurement [2];

Step 2: Identify the k neighbors that have the lowest distance to the new element;

Step 3: Identify the most frequent class among the k elements identified in the previous step and classify the new element as belonging to that class.

3 The MIT Mobile Device Speaker Verification Corpus

The MIT Device Speaker Verification is a corpus that was collected by smalls mobile devices using multiply microphones in multiply environments. This corpus, which is publicly available by MIT [7], has the objective to aid research on speaker verification for mobile devices in variable environments. The data was collected by prototypes of mobile devices provided by Intel. In order to simulate the various scenarios encountered by real-world speech verification systems, the collected speech data consisted of two unique sets: a set of enrolled users and a different set of dedicated impostors. For the enrolled user set, speech data was collected over the course of two different twenty minute sessions (one for training and one for evaluation) that occurred on separate days. For the impostor set, users participated in a single twenty minute session.

Within each data collection session, the user recited a list of name and ice cream flavor phrases which were displayed on the device. In total, 12 different lists were created for enrolled users while 7 lists were created for impostors. Enrolled users recited two phrase lists which were almost identical, differing only in the location of the ice cream flavor phrases on the lists. The first phrase list was read in the enrolled user's initial data collection session, while the second list phrase was used in the subsequent follow-up session. In total, each session yielded 54 speech samples per user. This yielded 5,184 examples from enrolled users (2,592 per session) and 2,700 impostor examples from users not in the enrollment set. Within the enrolled set of 48 speakers, 22 were female while 26 were male. For the impostor set of 40 speakers, 17 were female while 23 were male.

4 Experimental Results

4.1 Preprocessing and Feature Extraction

The first task of our systems is the preprocessing of the signal using the VAD algorithm described in Section 2. We apply the VAD algorithm in order to remove the non-speech portion present at the beginning and at the end of most samples of the dataset. The VAD algorithm removes a considerate portion of the speech signal at the beginning and at the end of the sample. The main advantage is that the number of feature vectors generated that contains features that does not express the user voice is reduced, decreasing the intra-variability of the speaker data. Another advantage is that, with the number of feature vectors reduced, the processing time to develop the GMM and to evaluate the test database is also reduced. At the MIT corpus, around 45% of the signal does not contain voice.

After applying the VAD algorithm, we extracted the features vector using the Mel-Frequency Cepstral Coefficients (MFCC) technique. For each sample on the dataset a 20 ms hamming window with a 10 ms overlap was applied and the 13 first coefficients of the mel scale were extracted.

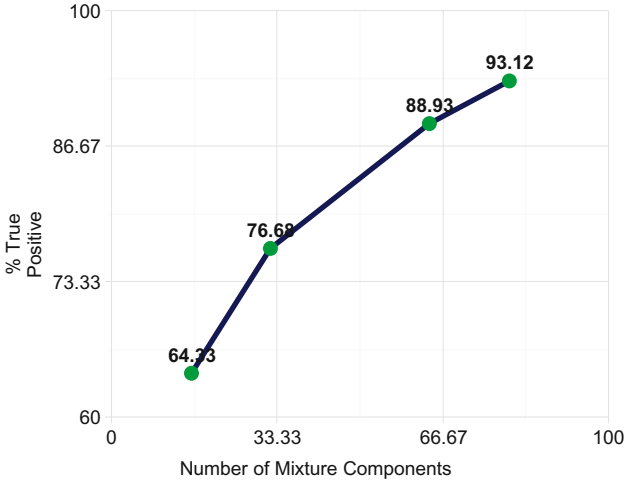


Fig. 1. True positive performance as a function of the number of component densities per speaker model

4.2 GMM, GMM-MLP and GMM-KNN Evaluation Experiments

To test our proposed systems, we perform experiments in the MIT Mobile Device Speaker Verification Corpus. All the dataset were used to test three different systems: a simple GMM system, one combining Gaussian Mixture Models with Multilayer Perceptron Neural Networks (GMM-MLP) and other combining Gaussian Mixture Models with k-Nearest Neighbor (GMM-kNN). The feature vectors used to model all the GMMs are the same for all systems. We used vectors extracted from 40 of the 54 speech samples from each record session to train the GMMs, given a total of 80 samples per speaker. The remainder, 28 samples, were used for testing. For all the proposed GMM based systems, the GMM provides a probability related to the tested speech and based on the value of this probability, the system make a decision of acceptance/rejection for that user.

The first experiment compares the performance of the GMM with respect to the VAD algorithm. MFCC vectors from all speakers samples with and without the application of the VAD algorithm were extracted. Next, two different types of GMMs with 80 mixture components for all speakers were built and verification tests over this models were performed. Table 1 shows the results of this tests in respect to the GMM verification performance as true and false positive rates. The false positive rate was obtained by testing with the impostors speakers.

The GMM+VAD outperforms the GMM system showing a 5.77% higher rate for true positive and an improvement of 3.54% in the false positive rate. From this results, we observe that the frames removed from the VAD algorithm improve the system in terms of the verification performance. It is known that, if the voice signal length and, as consequence, the number of feature vectors are reduced, the system can perform the process of training and verification in less time.

Table 1. The comparison between GMM and GMM+VAD systems

System	True Positive	False Positive
GMM	87.35%	13.41%
GMM+VAD	93.12%	9.87%

To investigate the verification performance of the GMM with respect to the number of component densities per model, we conduct an experiment using speaker models with 16, 32, 64 and 80 components. Fig 1 presents the result for this experiment, as the percent of true positive performance versus the number of Gaussian components. From the results, we can observe that the performance rate increases with the number of components. With 80 components a performance of more them 90 % was obtained, indicating that with this number of components the speakers can be modeled adequately. Tests with a higher number of GMM components were not performed because of hardware limitation.

In the next experiment, two hybrid systems were tested, GMM-MLP and GMM-KNN. In the GMM-MLP, an MLP was trained to learn the probabilities of each speaker and the neural network was used to make the acceptance/rejection task. The training data for both systems was divided in two classes: the real speaker class and the impostor speaker class. Each of the two classes had 80 samples. The speaker class has the probabilities for all the samples used for training the GMM, i.e. a verification test is done in the GMM over their own training data and, with this, creating an space of speaker probabilities. For the impostors class, 80 random samples were collected from speakers on all the MIT dataset and tested the samples over the speaker GMM, creating an space of impostor probabilities. This process was carried out for all speakers. Based on the previous results from the experiments using the VAD and Gaussian components, a GMM with 80 components was chosen and the samples processed by the VAD algorithm. All MLPs used in this experiment had 2 hidden layers, each of them with 10 neurons, and they were trained with a back-propagation learning algorithm using the sigmoid logistic function and a learning rate equals to 0.001. For the kNN space, a parameter k set to 6 was used which showed a good performance. Table 2 presents the results in terms of false and true positive rates (%) for both systems (GMM-MLP and GMM-kNN). As can be observed, in the GMM-MLP method the true positive rate decreases when compared to the GMM-only method. However, the GMM-kNN system obtained better results, with an increase of 1.2% in the true positive rate and a improvement of 3.64% in the false positive.

Table 2. Experiments result in terms of true and false positive (%) for GMM-MLP and GMM-KNN systems

System	True Positive	False Positive
GMM-MLP	91.17%	8.23%
GMM-KNN	94.26%	6.23%

5 Conclusion

Presently there are many systems developed for speaker verification and several models based on GMM models since they have proved to be one of the most efficient method for voice recognition and identification. Here, we proposed a hybrid system that combined an neural network and kNN algorithm to improve the performance of the traditional Gaussian Mixture Model. Also, we used a database of voice signals obtained from different mobile devices and recorded in different real life situations to test the method. The verification system was performed independent of the speaker text. The proposed hybrid methods showed a better performance when compared to the traditional GMM model. The results showed that the speaker verification system can performs well in an noisy environment.

References

1. Duda, R., Hart, P., Stork, D.: Pattern classification. Citeseer (2001)
2. Jain, A., Dubes, R.: Algorithms for clustering data. Prentice Hall (1988)
3. Knight, K.: Connectionist ideas and algorithms. Communications of the ACM 33(11), 74 (1990)
4. Lippmann, R.P.: An introduction to computing with neural nets. ARIEL 209, 115–245 (1987)
5. Rabiner, L.R., Sambur, M.R.: An algorithm for determining the endpoints of isolated utterances. Bell Syst. Tech. J. 54(2), 297–315 (1975)
6. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital Signal Processing 10(1-3), 19–41 (2000)
7. Woo, R., Park, A., Hazen, T.: The MIT Mobile Device Speaker Verification Corpus: Data collection and preliminary experiments. In: Proc. of Odyssey, The Speaker & Language Recognition Workshop (2006)
8. Zeng, J., Xie, L., Liu, Z.Q.: Type-2 fuzzy Gaussian mixture models. Pattern Recognition 41(12), 3636–3643 (2008)

A Neural Network Based Approach for GPCR Protein Prediction Using Pattern Discovery

Tsang Ing Ren[1], George D.C. Calvalcanti[1],
Francisco Nascimento Junior[1], and Gabriela Espadas[2]

[1] Center of Informatics, Federal University of Pernambuco, Brazil
[2] Atlanta International School, Atlanta, Georgia, USA
{tir,gdcc,fnj}@cin.ufpe.br,
gabiespadas@live.com
http://www.cin.ufpe.br/~viisar

Abstract. Artificial neural networks have been applied, as solutions, in several different problems within the field of bioinformatics. Similarly, pattern discovery algorithms have also been used to uncover hidden motifs in protein sequences, which have further contributed to understanding the problem of classification of different protein sequences. G-protein coupled receptors (GPCRs) represent one of the largest protein families in the Human Genome. Most of these receptors are major targets for drug discovery and development; therefore, they are of interest to the pharmaceutical industry. The technique used in this article combines both: neural network and pattern discovery methods to develop a protein prediction procedure in relation to its functional class, more specifically, to predict the GPCR protein. Vilo [2] proposed an algorithm to extract patterns of regular expressions from known GPCR protein sequences. Our contribution in this article is to combine these patterns as features for a neural network. We select patterns through the PCA (Principal Component Analysis) procedure and produce a learning machine for the prediction of the GPCR super class.

Keywords: Protein prediction, Neural Networks, Pattern Discovery.

1 Introduction

G protein-coupled receptors (GPCRs), is also known as seven transmembrane receptors or 7TM receptors because of its structural characteristic. They represent one of the largest protein families in the human genome. Also, they are a major target for drug discovery and development because approximately 50% of these receptors appear to be of relevance to the pharmaceutical industry seen as 40% to 60% of the current drugs on the market target GPCRs. An important aspect of their function is the coupling specificity with members of G-protein families. A GPCR can interact with one or more G-proteins; a problem presented is the prediction of the coupling specificity of GPCRs to the G-protein family class. Several prediction methods have been developed to successfully accomplish this task [1][4-7]. However, we are interested

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 459–466, 2012.

in solving a different problem, the prediction of a GPCR protein from an unknown class of protein. This procedure is of interest to identify GPCR proteins that are not yet annotated.

Based on the patterns of regular expressions found by Moller et al. [1], we develop a neural network procedure to predict if a protein belongs to a GPCR protein class. The main objective of Moller's el al. work was to device a method to predict the coupling specificity class of a GPCR protein to be of either $G_{i/o}$, G_s or $G_{q/11}$ class. Similarly, we would like to develop a method to predict if, whether or not, an unknown protein sequence belongs to the GPCR protein class. This problem is of relevance since the discovery of new GPCR proteins is of great interest to the pharmaceutical and biotech industry.

A class of statistical learning algorithms, Support Vector Machines (SVMs), presented by Vapnik became popular within the machine-learning community during the 1990s [11]. When SVMs are applied to the simplest learning problem, two-class pattern recognition, the learning machine shows a series of labeled examples from two categories and is trained to distinguish between them. With the use of an SVM [13], Karchin et al. successfully developed a classifying system for G-protein coupled receptors. Nascimento et al. [13] have also shown a similar method as proposed here for GPRC prediction; however, we have applied another type of neural network.

2 Proposed Method

A series of 40 patterns of regular expressions [1] were obtained to characterize the coupling receptors for each of the three coupling class using the SPEXS patterns search program [2]. A classification system was constructed to successfully predict the GPCR coupling classes based on derived pairs and triplets of those patterns. The classification was obtained by simple measurements of specificity and sensitivity of the patterns to the classification of best accuracy.

The success of Moller's et al. classification scheme, based on 120 patterns and the high values of the specificity of each pattern suggest that these patterns are appropriate for not just detecting coupling specificity classes of GPCR proteins but can also be used to characterize a general GPCR protein from a different type of protein. Using the 120 patterns as features for the GPCR protein prediction method, we are making an implicit assumption that the patterns well qualify a GPCR protein. Our approach is to use the total number of patterns as signature for a classifier. We first test our assumption using a simple chi-square statistic, to evaluate how well the patterns will perform in a first classification attempt; secondly, we use a more sophisticated classifier system, namely a neural network based on statistical learning.

Table 1 shows the list of all patterns of regular expressions found by the SPEXS search program [2]. These patterns of regular expressions were used to classify the G-protein family, each column represents the specific coupling that the patterns best represents.

Table 1. List of Patterns found that best represent a specific coupling

Number	G_{io}	$G_{q/11}$	G_s
1	[ILV]...SG.{0,10}R	T..[RK].{0,10}S..T	A[ILV].{1,5}Y..[ILV].T
2	N..R.{1,4}R	A.{3,6}V[ILV][RK]	A.{1,5}RY....T
3	Y.A.{1,8}A[ILV]	P..[AGS]T.{0,10}S	I....RY.{1,10}R
4	A[ILV].{2,5}RT	[AGS][ILV][ILV][RK].{2,10}S	I....RY.{4,6}T
5	N..[RK]..R	S[FWY].{1,11}Q[ILV]	LR.{1,9}T...[ILV]
6	K.[RK].{0,10}K.[ILV]	[AGS].{0,3}S..T[ILV]	RS.{3,13}C[AGS]
7	V...[RK]....R	S...L.{2,9}TL	[ILV].[FWY]H.{1,3}I
8	[RK]...[CM][RK]	[RK]F....K	F.{1,4}Y....T
9	V[RK].{1,10}SG	[AGS].[ILV].{0,10}K.F	I....RY.{4,4}T
10	K.[RK].{1,4}L[RK]	[AGS].S.[RK].{0,10}F	I....R[FWY]
11	[FWY][ILV]..V.{2,10}R	S...L.{1,10}T[ILV]	I....RY....T
12	Y.[RK].[RK].{0,9}T	[RK].T.{0,10}Q[AGS]	I....RY
13	[ILV].A[AGS].{1,4}R	[AGS]...L.{1,10}TL	[FWY].A.{2,6}Y..[ILV]
14	FR....[RK].{0,3}L	[AGS][ILV][ILV][RK]	I.[AGS].{1,10}S...R
15	DRY.[AGS].{3,6}A	A.{0,10}V[ILV][RK]	[ILV].[FWY]H.{3,12}T
16	F[RK]....K.{1,7}C	[AGS].{0,3}V[ILV][RK]	L..H.[ILV]
17	A....[ILV].{1,8}RT	F.{0,10}Y...[RK]	[ILV].[FWY]H.[ILV]
18	[RK]....R.{0,9}EK	[CM].[FWY].{3,12}P	[ILV].[FWY]H.I
19	[RK]R.{0,3}TR	S.[AGS].{3,13}TL	A....[RK][RK]I
20	KA.{3,6}T	V[AGS].{0,10}S.[AGS].[ILV]	[AGS].{0,10}L..H.[ILV]
21	DR.{4,11}H...[AGS]	Y....[RK]P.{2,10}A	[ILV].[FWY]H.{3,10}T
22	R....K.{0,8}T[AGS]	[ILV].....A.T	[FWY]H.I.{0,3}T
23	[RK][FWY][ILV].{2,5}V	S...L.{1,11}Y	S.{5,12}S.L.[RK]
24	N.{2,5}R.[FWY]	A.{3,12}V[ILV][RK]	S.{5,9}S.L.[RK]
25	Y.[AGS].{1,8}A[ILV]	[AGS].{2,5}V[ILV][RK]	Q.{0,9}S.L.[RK]
26	N..[RK].{1,4}R	[FWY].{4,7}KP	A.{1,5}RY..[ILV].T
27	[ED].{0,3}N..[RK]	R.[RK].{0,10}K[AGS][AGS]	F.{1,10}A...H
28	Y.{2,5}I..[AGS]	[ILV]A.{2,4}S.[ILV]	[ILV]..H.[ILV].{1,3}T
29	N..[RK].{1,11}R	[AGS].[ILV].{2,10}L.[FWY]	[FWY]H.I.{0,10}V
30	[RK].R.{2,12}K[RK]	[AGS][FWY]..[FWY]	A..[FWY].{0,3}H
31	[ILV]...SG	S.S.{1,11}L.S	I....[RK]Y.{4,6}T
32	[AGS][RK]..[ED].{0,10}R	[ILV].L.{6,11}A.T	A.{1,5}R[FWY]....T
33	[FWY].A.{1,9}A[ILV]	K.{0,3}N.P	A.{2,6}Y..[ILV].T
34	R[FWY].[AGS][ILV].{0,7}A[ILV]	[ILV].L.{6,10}A.T	A..[FWY].{0,8}H
35	[ILV].R....V	[RK][FWY]....K	[AGS].{1,5}RY....T
36	[RK]Y.[AGS].{3,5}A	[AGS].S.[RK].{2,10}F	I....[RK]Y.{1,10}R
37	[ILV]...SG.{0,8}E	[ILV].{3,6}S.Q	R[FWY]H.{5,14}R
38	[FWY].[AGS][ILV]..A	C.[FWY].{2,11}K	[RK]S.{3,13}C[AGS]
39	[RK]..[RK].{0,3}R[ILV]	C.[FWY].{2,12}K	[RK].[ILV].C.R
40	[ED]A.{0,3}E	S....[RK]A.{3,10}S	[RK].[ILV].C.[RK]

2.1 Characteristic Matrix

Characteristic matrix will be used as input for the classifier. This matrix is created using String Matching algorithm between regular expressions and sequences, thus

obtaining the number of occurrences for each pattern per sequence expressed in a matrix n x p, where n is number of sequences and p represents the number of patterns.

2.2 Chi-Square Test

Chi-Square is a statistical procedure used to evaluate the efficiency of the fit from a known distribution compared to a dissimilar distribution [8]. The equation is defined as:

$$\chi^2 = \sum_{i=1}^{p} \frac{(M_i - m_i)^2}{m_i} \tag{1}$$

where, M_i is to equal the average value of the curve we want to compare and m_i is equal to the average value obtained from the known data. The index "i" represents all 120 different patterns; therefore, the smaller the value of the test, the closer curve is to the known distribution. We will generate two distributions, one that characterizes the GPCR protein and another that characterize all other proteins (non-GPCR proteins), so that we can compare the distribution of the sequence that we are interested in predicting to both known distribution that characterize either GPCR or non-GPCR protein.

The main objective as to the use of the Chi-Square Test is to verify if the frequency of a specific observed event in a sample has strayed from expected frequency. In our case, we would want to consider if the analysis of frequency of matches of pattern set which differentiate the sequences of GPCR class against sequences of Non-GPCR class.

2.3 Artificial Neural Network

We used a back-propagation neural network [9] as a classifier, testing several different learning algorithms and architecture. We chose the resilient back-propagation (RPROP) [10] as the final algorithm for our experiments due to its excellence in performance. Figure 1 shows a typical architecture of back-propagation network, the standard multi-layer network with gradient descent algorithm which minimizes cost function.

The RPROP performs a local adaptation according to the behavior of the error and it takes into account the sign of the partial derivative patterns. If there was a sign change with respect to the last iteration, the update value is decreased by a factor of η; however, if the last iteration produced the same sign then the update value is decreased by a factor of η^+, making its convergence faster than the standard back-propagation algorithm.

2.4 GPCR and Non-GPCR

Our objective is to answer the question: Is an unknown sequence a GPCR protein? We analyzed the pattern of co-occurrence from the data of protein sequences that we know to be GPCR and compare them to sequences that we know are non-GPCRs. A difference between total number of patterns for GPCR and non-GPCR sequences emerged from this analysis and we want to test the possibility to use these patterns to

derive a distinction from the class function protein sequence. An analysis for a group of GPCR's and non-GPCR is presented.

3 Experiments

In this experiment, 778 sequences of GPCR from GPCRDB were used along with 2565 Non-GPCR, the same set used in [12]. Hence, characteristics matrix size 3343 x 120 containing the number of occurrences of each pattern in each sequence was created.

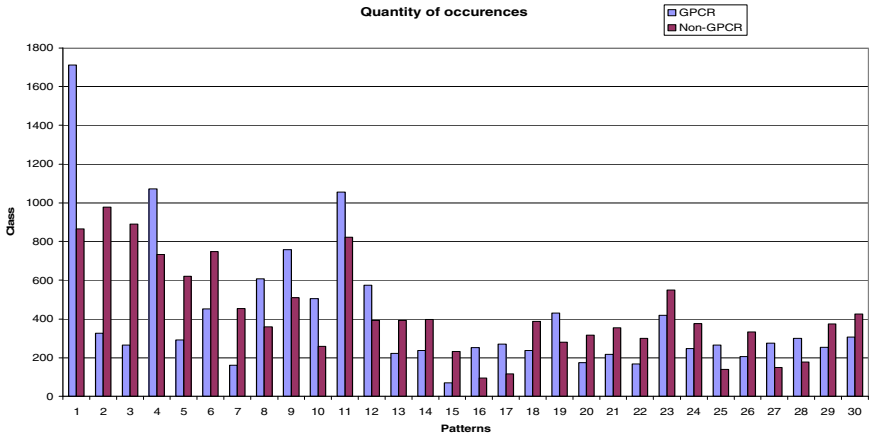


Fig. 2.

Distribution Plot. Using a set of GPCR and Non-GPCR protein sequences, we generated a distribution plot based on the 120 patterns suggested above. These plots provide some insights on the performance of these patterns as features to characterize GPCR from Non-GPCR sequences. Figure 2 shows a histogram plot representing the number of occurrences while using only 30 patterns.

The variation in the distribution gives an idea of how a classifier can perform. These plots show a significant difference between GPCR and non-GPCR proteins. The next step is to apply a simple chi-test statistics to verify if this difference will be sufficient to make the recognition possible.

3.1 Results

Chi-Square Test

The average number of occurrences of each individual pattern was used to generate the distribution of GPCR and non-GPCR sequences. The null hypothesis, the frequency of GPCR not is different from the frequency of non-GPCR. Therefore, the alternative hypothesis defends the possibility of such a difference. Using a matrix of

occurrences, it was calculated a mean and standard deviation for both subsets. Applying the chi-square equation shown in the previous section the following result were obtained:

The results imply that the null hypothesis was incorrect, concluding that frequencies of GPCR and Non-GPCR are, statistically, different. So, it is possible to use such frequencies like characteristics for classifying.

Table 2. Statistics from Samples

Samples	Mean	Standard Deviation
GPCR	219,00	224,94
Non-GPCR	208,90	211,49
Value of X2 = 47,63		
Critic value = 3,841 (5% of significance)		

Neural Networks

A more elaborate way to predict the protein class was tested using artificial neural networks; as well as, the resilient back-propagation learning rule. For this experiment, we used 439 GPCR and 156 non-GPCR sequences leaving 10 sequences of each class for testing. The architecture of the net consists of an input layer with 120 elements, 1 hidden layer with 60 elements and an output layer with 2 elements. The transfer function used was tan-sigmoid and for the training phase, we set the output to be equal to [1 –1] for GPCR and [-1 1] for non-GPCR. For the test results, we considered that a sequence belong to a GPCR class if the output of the ANN were in the range [1.2>O1>0.7 –1.2<O2<-0.7] and for a non-GPCR, if the output were [–1.2<O1<-0.7 1.2>O2>0.7] ;the sequence is considered indeterminate otherwise. Below, we present a summary of the best result:

Pattern choice: To check if the 120 patterns used were indeed proper for the classification between GPCR and non-GPCR sequences, we tested the procedure using the same number of non-GPCR sequences and 400 GPCR sequences as learning sets and 43 as test set. The result of this test was: 100% recognition for the GPCR class.

In the current setup of the neural network, the input does not specify the relative location in the sequence where the patterns occur. This specification would give an additional level of information that can help the process of classification. To verify this claim, we analyzed the occurrence of patterns in relation to the relative position (100 compartments) for a set of GPCR and non-GPCR sequences. The values were normalized so that we could compare proteins with different length. A different pattern of occurrence between the two classes emerged, suggesting that another level of information can be extracted from this modification. In a practical setup we developed these ideas using 10 sequence locations yielding 1200 input nodes for the neural net.

Testing the non-GPCR class, just one sequence, was classified as a GPCR, after testing for transmembrane prediction, we verify that the sequence has in fact 7 transmembrane, and it was correctly classified as GPCR, even though in our set it belonged incorrectly to a non-GPCR sequence. Thus again confirming the validity of

this method as shown in the table with the overall result of this test. For learning, we used 430 GPCR and 9215 non-GPCR sequences, in which the NN was able to correctly learn all of the non-GPCR and 392 GPCR. The number of false positives is significantly low, implying that most of the sequences that this procedure classifies as being GPCR have a high probability of being accurate.

IPI Database

Once the NN was trained, the next step was to try to obtain GPCR sequences that were not cataloged in our internal BUZZ database. We applied the procedure in the IPI database, 55747 sequences. From those we obtained 1021 sequences that the system classified as GPCR, a reasonable number for the total number of GPCR that the database must contain. After comparing the results to the BUZZ database, 538 sequences were still present and still leaving 446 sequences. Those 446 sequences have the possibility of being GPCR proteins but were neither found present in the BUZZ or public GPCRDB databases. A validation procedure to analyze these sequences is necessary to confirm the sequence, mainly consisting of checking for the presence of 7 transmembrane regions. However, to be able to have full confidence that the sequences are GPCR, "wet" approached are in fact needed.

True positive	65%	True negative	99.5%
False negative	30%	False positive	0%
Indeterminate	5%	Indeterminate	0.5%

4 Conclusion

A procedure for identifying possible GPCR protein based on patterns of regular expressions and a neural network classification scheme was proposed. This procedure which basically allows checking for the coupling specificity of a GPCR. And differently from other techniques that mainly look for patterns in the transmembrane region, the proposed method is not restricted to a specific region on the protein sequence, thus the same procedure can be used to identify others classes or sub-classes of proteins. The constraint of the method relies on the selected known sequences where the patterns are obtained and the NN is trained.

The result of the IPI database search provide greater proof that the method can be applied to a larger search and is able to retrieve a reasonable number of identified sequences. Since it correctly retrieved most of the sequences in BUZZ and 92 correct sequences which, identified by the public GPCRDB database, were not included in BUZZ.

Another point of interest is the ability to assign the coupling specificity GPCR belongs to. Different from the original Moller et al. article, the method proposed uses a

NN as a classifier that also uncovers multiple couplings. Protein functional prediction, in general, is a very challenging field, where no single technique has been proven to work without fail. Structural based methods are often used to improve the prediction of the method. Even though several techniques from the literature apply just the primary sequence information, this will prove limited if one is looking for a close to 100% prediction. The method described here will also be limited in scope. Due to it's very general, in the context that it does not make use of (biological) information specific to the protein class; on the other hand, it can be easily applied to classify different types of proteins.

References

[1] Moller, S., Vilo, J., Croning, D.R.: Prediction of coupling specificity of G protein coupled receptors to their G proteins. Bioinformatics 17, S174–S181 (2001)
[2] Vilo, J.: Discovering Frequent Patterns from Strings. Department of Computer Science, University of Helsinki (1998)
[3] Wang, J.T.L., Shapiro, B.A., Shasha, D.: Pattern Discovery in Biomolecular Data: Tools, Techniques, and Applications. Oxford University Press, USA (1999)
[4] Cao, J., Panetta, R., Yue, S., Steyaert, A., Young-Bellido, M., Ahmad, S.: A naive Bayes model to predict coupling between seven transmembrane domain receptors and G-proteins. Bioinformatics 19, 234–240 (2003)
[5] Sgourakis, N.G., Bagos, P.G., Hamodrakas, S.J.: Prediction of the coupling specificity of GPCRs to four families of G-proteins using hidden Markov models and artificial neural networks. Bioinformatics 21, 4101–4106 (2005)
[6] Sreekumar, K.R., Huang, Y., Pausch, M.H., Gulukota, K.: Predicting GPCR–G-protein coupling using hidden Markov models. Bioinformatics 20, 3490–3499 (2004)
[7] Baldi, P., Brunak, S.: Bioinformatics: The Machine Learning Approach. The MIT Press, Cambrigde (1998)
[8] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipies in C++: The Art of Scientific Computing. Cambridge University Press (2002)
[9] Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall (1998)
[10] Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the Rprop algorithm. In: Proceedings of ICNN, San Francisco (1993)
[11] Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers. HP Laboratories (March 16, 2004)
[12] Karchin, R., Karplus, K., Haussler, D.: Classifying G-protein coupled receptors with support vector machines. Bioinformatics 18, 147–159 (2002)
[13] Nascimento, F., Ren, T.I., Cavalcanti, G.D.C.: A SVM for GPCR Protein Prediction Using Pattern Discovery. In: Proceedings of HIS (2008)

Real-Time Head Pose Estimation for Mobile Devices

Euclides N. Arcoverde Neto[1,2,3], Rafael M. Barreto[1,2], Rafael M. Duarte[1,2], Joao Paulo Magalhaes[1], Carlos A.C.M. Bastos[1,2], Tsang Ing Ren[1], and George D.C. Cavalcanti[1]

[1] Center of Informatics, Federal University of Pernambuco, Brazil
{enan,rmb,rmd,jpm,cacmb,tir,gdcc}@cin.ufpe.br
[2] C.E.S.A.R – Recife Center for Advanced Studies and Systems
[3] Faculdade Boa Viagem, Brazil
http://www.cin.ufpe.br/~viisar

Abstract. Mobile devices are an essential equipment in modern life. Nowadays, its presence is so widespread that almost everyone has a mobile phone, smartphone or tablet device. There are several different ways to interact with those equipments, such as the use of the keypad or the touchscreen. Here, we propose a real-time head pose estimation technique based on the secondary video camera as a means of interaction between the user and the device. The proposed technique is composed of several computer vision methods specially optimized to be able to operate in a restrict environment and a head pose estimation based on the calculations of the roll, yaw and pitch movements. Experiments were conducted based on 363 videos of 27 different people in a varied environment (illumination and background).

Keywords: Head pose estimation, adaboost, mobile devices.

1 Introduction

The interpretation of audio and visual signals is essential to understanding the communication process in human-human interaction [1]. In recent years, human-computer intelligent interaction (HCII) has received great attention [2]. The objective of this area is to understand the patterns in human communication to produce a truly natural interaction between human and computers. Recently, advances in mobile technology enabled new ways of interaction with mobile devices, which can be called human-mobile interaction.

We propose a novel real-time method for head pose estimation based on facial features for human-mobile interaction. The proposed approach is robust and does not require any extra special hardware, since it used the images obtained from the frontal (or secondary) camera of mobile devices. The estimation of the head movements were obtained by comparing the facial features positions from the sequential frames of the video.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 467–474, 2012.

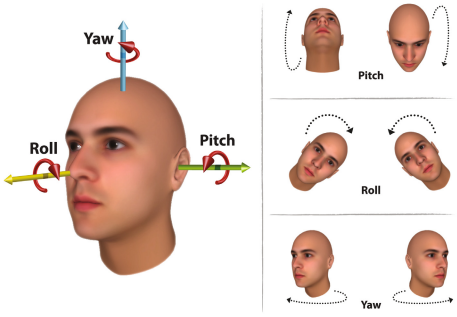


Fig. 1. Orientation of the head in terms of *pitch*, *roll*, and *yaw* movements describing the three degrees of freedom of a human head

The pose movements of the human head have three degrees of freedom (DOF), the orientation of the head is described in terms of *pitch*, *roll*, and *yaw* movements as shown in Fig. 1. The use of three DOF is the most complex approach in head pose estimation [3] since it is necessary to estimate the position of the head in a full 3D-orientation space.

The method proposed in this article begins by detecting a face, followed by the eyes and nose localization using a novel and simple approach. With the information of the eyes and nose positions, head pose can be estimated by comparing the position of the eyes and nose from one frame to the previous frame. Face detection is interspersed by periods of tracking in order to reduce the computing requirements, as performed by [5]. The approach proposed here uses the current most efficient and robust face detector method, the Viola and Jones general object detection algorithm [7,8]. The obtaining result is robust and accurate and a series of optimizations, similar to [6], were performed to reduce the high computational requirements of this algorithm. Finally, an estimation of the head pose was obtained using three set of equations developed to calculate roll, yaw and pitch movements. The remainder of this paper is structured as follows: section 2 describes the proposed method; experiments and results are presented in section 3 and the conclusions are given in section 4.

2 Real-Time Head Pose Estimation for Mobile Devices

An overview of the proposed head pose estimation system is summarized in Fig. 2. In the system: an image is captured from the frontal (or secondary) camera of the device; the face detection is performed on the captured image; if a face is detected the image is preprocessed using a illumination correction; eyes and nose are localized; and their positions are given as input for the tracker; with present and last positions the head pose estimation can be obtained. Tracking is performed until an error is detected by the control unit, when the system returns to search for a face. We describe the details of each module in the subsections below.

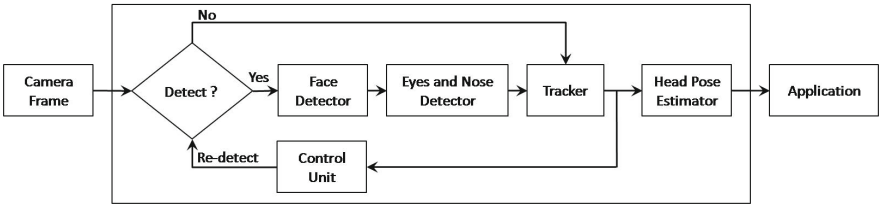


Fig. 2. Overview of the system

3 Face Detection

In order to extract facial features to be used during the tracking phase, a face must be located. Hence, a face detection algorithm is a crucial step in the proposed technique. Initially, algorithms based on skin color segmentation were considered because of their reduced computational cost, which is a requirement for implementation in mobile platforms. However, experiments using the mobile camera showed that this approach is not robust to noise, variations in environment, and ethnic differences. Therefore, it was not precise enough for the intended use. More accurate alternatives are available but at higher computational cost. One of these alternatives is the object detection algorithm proposed by Viola and Jones [8]. Despite its higher resource requirements, this algorithm represents the state-of-art in general object detection method (including face detection), both academically and commercially, and there are reports of success full implementations of real-time software versions of it on mobile platforms [6]. This led to the choice of the Viola and Jones face detector to compose the technique presented in this paper.

3.1 Optimizations

An optimized version of the Viola and Jones object detector was implemented using a frontal face training. The optimizations are described below:

Scaled Cascades before Algorithm Execution. The image size does not change during the head pose estimation flow. As a consequence, it is possible to scale all cascades before starting the algorithm execution.

Pre-calculation of Rectangles Coordinates. Every time a feature is evaluated, it is necessary to calculate the points for each rectangle. This can be avoided by pre-calculating these coordinates at the beginning and representing each rectangle as 4 pointers to the integral image buffer at that coordinates. The application of a profiling tool showed that approximately 70% of the detection execution time was due to the feature evaluation. This number dropped to about 50% after this optimization.

Reduced Search Space. The detector search space is determined by three parameters: the scale factor; the minimum face size; and the minimum step of the sliding window. The execution time is proportional to this space, thus, changing these parameters is an effective way to increase the algorithm performance. However, this type of optimization also affects the hit rates of the detector and must be performed carefully. It was found, experimentally, that the following parameters are a good trade-off between performance and accuracy for the intended use of the detector: 1.2 for the scale factor; $0.4 \min\{M, N\}$ for the minimum face size, where M and N are the image dimensions; and 4 as the minimum step.

3.2 Eyes and Nose Detection

In the problem of computer face detection and recognition, as well as facial expression analysis and recognition, eyes and nose are fundamental features, since they are stable and have a high degree of dissimilarity compared to its vicinity, the position of the eyes and nose are good reference point to monitor the movement of the face, which means face tracking. Considering that a face is already detected, we propose a method to locate the position of the eyes and nose using horizontal projections and the usage of templates defined by anthropometry of the face based on [4]. For the head pose estimation problem, it is important to notice that the exact location of the eyes is not of fundamental relevance but the relation between these points and the movement of the head that matter the most for a good tracking.

Peng et al. [4] proposed a technique for eyes detection using the horizontal and vertical projection and their gradient. Since the region of the eyes has more detail, which means more variation in the pixels values, compared to the rest of the face, the horizontal projection of the gradient presents a peak at the eyes level. Similarly, the vertical projection shows peaks at the edge of the face. The medial axis of the face can be localized using the vertical projection of the face. Because the pixels in the line with the nose are lighter, the vertical projection is maximal at this line. The output of the method produces a horizontal line at the eyes level and three vertical lines, two lines delimiting the face and a line passing vertically through the nose. Subsequently, the template matching technique is used with a standard template for the eyes resized from the region of interest as defined in the previous phase (the region between the vertical and horizontal lines). The goal of this step is to localize the iris of the eye.

We propose a modification of this method to minimize the processing time, since we are interested in a real time response on a mobile platform. A gradient operator in the x direction is applied on the luminance component of the image and the horizontal projection is calculated. The peak position of this projection defines a horizontal line at the height in which the eyes are located. Using a standard face template based on face anthropometry, the regions around the eyes are defined. The template considers the first eye is located at 30% of the total length of the face and the second at 70%. Considering the distance between the eyes is d_{eye}, the point representing the nose is determined as the point which is perpendicularly below the midpoint of the eyes ($d_{eye}/2$), having a distance

of $d_{eye} * 0.45$. We can define this position formally as: given that the eyes are represented by two points (x_1, y_1) and (x_2, y_2), the point representing the region of the nose is determined as $x_{nose} = (x_1 + x_2)/2$ and $y_{nose} = y_1 + (x_2 - x_1) * 0.45$. These estimated positions were obtained through empirical studies of several different persons. The objective is to define the area where the eyes should be located in order to perform tracking.

The transformation used is based on a log function. Its general form is defined as $y = c * \log(1 + x)$, with $0 \leq x \leq 255$ and $c = 1$. This transformation expands the values of dark input values in an image while compressing the higher-level values, which causes the output image to be brighter than the input.

The resulting values of the log function were pre-computed to avoid the time-consuming float point operations, resulting in an integer to integer conversion.

Although simple, this technique presented an accuracy greater than 90% when tested with 361 images.

3.3 Control Unit

This module was implemented to verify whether the tracking is consistent. It uses heuristics based on the eyes and nose coordinates in current and previous frames, distance between the eyes, and facial movements. The control unit sets two different flags to warn of tracking inconsistencies: if it detects that tracking is lost and a re-detection must be performed, it aborts and tries to perform a re-detection; in less severe cases, it simply recommends a re-detection and continues running normally, until facial movements provide a hint that the face is centered, when a re-detection is performed.

The first heuristics aims to detect situations where tracking has degraded to an unacceptable level and must be restarted. In our experiments, we noticed it happens often with occlusion and fast movements. The recommendation heuristics aims on preventing tracking degradation, which is expected with prolonged use.

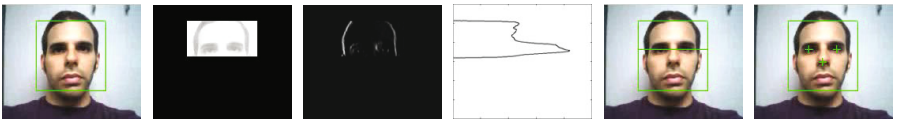
3.4 Face Pose Estimation

The estimation of the head movements is based on three tracking points. Considering pt_{eyeL} and pt_{eyeR} as the left and right eye reference points, pt_{nose} as the nose reference points, and their dashed versions as their updated coordinates, we have then the following equations to calculate facial movements:

$$roll = \arctan\left(\frac{pt_{eyeL}.y - pt_{eyeR}.y}{pt_{eyeL}.x - pt_{eyeR}.x}\right) \tag{1}$$

$$yaw = pt_{nose}.x' - pt_{nose}.x \tag{2}$$

$$pitch = pt_{nose}.y - pt_{nose}.y' \tag{3}$$



(a) Face detection. (b) Log transformation. (c) Gradient operator applied to x direction. (d) Horizontal histogram projection. (e) Eyes level. (f) Eyes and nose detected.

Fig. 3. Images showing example of the technique used to detect the eyes. (a) Initially it is assumed that a face has been detected; (b) The log transformed image is calculated for the upper half of the face and (c) then the gradient operator in x direction is applied; (d) The peak of the horizontal histogram is calculated and from that point, (e) the projection of eyes on the face is defined; (f) From the projection, template matching is used to determine the points representing the eyes and nose.

This equations estimates movements as displacements. Roll angles (equation 1) depend solely on the current eye location, and are estimated as the angle formed by a straight line passing through the eyes and the horizon line, which is a precise estimation for the inclination angle in the *roll* axis. On the other hand, *yaw* and *pitch* movements (equations 2 and 3) are calculated using a detection time reference point, and their value is the displacement (in pixels) between the reference and current points. This approach for yaw and pitch does not result in a inclination angle, but we have chosen the displacement because it is simpler to calculate and provides adequate data to be processed by applications. It does, however, require a frontal pose initialization to work properly.

4 Experiments and Results

Experiments were conducted to verify the performance of each sub-system and the system as a whole. A video database, composed of 363 videos with 27 persons was created. In general, the videos have low quality (due to the use of the secondary or video-call camera). They were recorded at 15 fps with 176x144 pixels resolution. The database contains a variety of movements, including individual movements (only *roll*, *yaw*, or *pitch*) and complete movements (without a pre-defined sequence). On average, each movement was recorded five times. Furthermore, the database contains different scenarios including: controlled environment (uniform neutral background, no accessories, and good lighting conditions); different lightning sources (fluorescent, tungsten and daylight), intensities (strong and weak) and directions (frontal, lateral, upward and downward); facial accessories (spectacles, sunglasses, baseball caps, head phones, and hoods); different backgrounds (simple and complex); different movement speeds (2 and 0.5 seconds for a complete movement); variations in distance and angle between face and camera (distances of 20, 45, and 60cm; angles of 30 and 45 degrees).

4.1 Head Pose Estimation

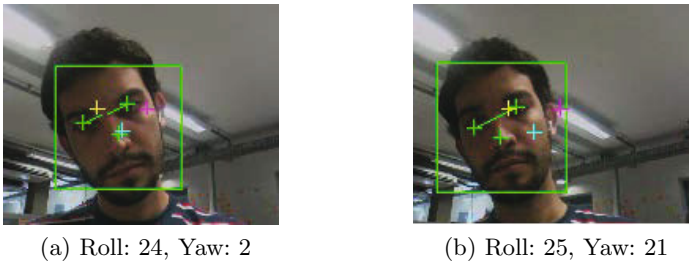
To validate our final results, we adopted a black-box approach. Our facial pose estimation technique is intended to be used as an API for mobile phones, thus, we validated only its resulting estimation movements. Moreover, evaluating the algorithm correctness on videos proved to be a tricky issue. Users tend to subtly move their head in more than one direction, even when they intend to perform movements in just one direction. After these observations, we have considered a pose estimation to be correct if the detected predominant movement is the same as observed in the video.

The tests were performed running the algorithm on PCs, as it was not feasible to test it with videos directly on the mobile platform. Each video execution was performed manually, and the results were marked as correct or incorrect. Overall results can be observed in Table 1.

Table 1. Overall results for technique proposed

Total	Correct	Incorrect
363	311	52
100%	86%	14%

The major drawbacks of our approach were occlusion and fast movements. Eye occlusion occurred often when users performed yaw movements, and caused a major tracking deterioration. Also, when fast movements were made, tracking points got easily out of place, due to the camera blurring. Another issue is related to the interference between roll and yaw movements: by definition, roll movements should be performed as rotations using the nose as axis (as conveyed in Fig. 4a), but we noticed that users do not perform it in this manner, as it is not comfortable for them. Instead, they tend to incline and move their head altogether to the same side (Fig. 4b), resulting in high values for yaw and roll movements.



(a) Roll: 24, Yaw: 2 (b) Roll: 25, Yaw: 21

Fig. 4. Interference between roll and yaw movements

5 Conclusion and Discussion

In this paper, a human-mobile interaction procedure based on a real-time head posed estimation for mobile devices is proposed. The method is composed of the Viola and Jones face detector optimized for mobile devices, eyes and nose detection based on histogram projection, the Lucas-Kanade tracking algorithm also optimized to be used in a mobile environment, a control unit to verify the detection and tracking phases and three set of equations to calculate the facial movements. The results of the experiments showed that the method is satisfactory for both time and performance. Even though most of the fundamental algorithms were developed based on well-know methods, the proposed method proved to be challenging to develop due to the optimizations required by the restricted mobile device.

Several different applications are possible once head tracking is available in the mobile device, such as game control and head gestures interpretations. With the widespread usage of mobile devices, we believe that these types of novel interaction methods are an interesting new way of interaction using natural human movements.

References

1. Jaimes, A., Sebe, N.: Multimodal human-computer interaction: A survey. CVIU 108(1-2), 116–134 (2007)
2. Lew, M., Erwin, M., Bakker, N.S., Huang, T.S.: Human-Computer Intelligent Interaction: A Survey. Springer, Heidelberg (2007)
3. Murphy Chutorian, E., Trivedi, M.: Head pose estimation in computer vision: A survey. PAMI 31(4), 607–626 (2009)
4. Peng, K., Chen, L., Ruan, S., Kukharev, G.: A Robust Algorithm for Eye Detection on Gray Intensity Face without Spectacles. Journal of Computer Science and Technology 5(3), 127–132 (2005)
5. Rahman, M., Ren, J., Kehtarnavaz, N.: Real-time implementation of robust face detection on mobile platforms. In: Proceedings of the 2009 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Washington, pp. 1353–1356 (2009)
6. Ren, J., Kehtarnavaz, N., Estevez, L.: Real-time optimization of viola -jones face detection for mobile platforms. In: 2008 IEEE Dallas Circuits and Systems Workshop: System-on-Chip-Design, Applications, Integration, and Software, pp. 1–4 (December 2008)
7. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. I–511–I–518 (2001)
8. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137–154 (2004)

Alternative Quality Measures for Time Series Shapelets

Jason Lines and Anthony Bagnall

School of Computing Sciences
University of East Anglia
Norwich
UK
{j.lines,anthony.bagnall}@uea.ac.uk
http://www.uea.ac.uk/cmp

Abstract. Classification is a very broad and prevalent topic of research within data mining. Whilst heavily related, time series classification (TSC) offers a more specific challenge. One of the most promising approaches proposed for TSC is time series shapelets. In this paper, we assess the current quality measure for shapelet extraction and introduce two statistical tests for shapelet finding. We show that when compared to information gain, these two quality measures can speed up shapelet extraction whilst still producing classifiers that are as accurate as the original.

Keywords: time series, shapelets, classification.

1 Introduction

Classification is a very broad and prevalent topic of research within the field of data mining. Whilst heavily related, time series classification (TSC) offers a more specific challenge. TSC typically involves problems where the ordering of data plays a critical role, often where it has been recorded in temporal order at fixed intervals of time. Many solutions for TSC have been explored, with much of the contribution focused on alternative distance measures for 1-Nearest Neighbour (1-NN) classifiers using either raw time series or transformed representations of the raw data (a comprehensive summary can be found in [5]). In particular, there is strong evidence to support the use of 1-NN classifiers with a Euclidean or Dynamic Time Warping (DTW) distance metric. However, 1-NN approaches suffer from drawbacks such as poor interpretability of results and relatively slow classification. As a result, many alternatives have been proposed. These include: shapelets [14,18,19], weighted DTW [8], support vector machines built on variable intervals [15], tree based ensembles constructed on summary statistics [4], fusion of alternative distance measures [2] and transform-based ensembles [1]. Of these, we feel that shapelets in particular have good potential for TSC due to their interpretability and fast classification of new cases.

Shapelets were first introduced in [19] as time series subsequences that are representative of class membership. The authors construct a decision tree classifier by recursively searching for the most discriminatory shapelet in a data set.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 475–483, 2012.

To assess the quality of a shapelet, the authors calculate the distance from the shapelet to each element of a data set, sort the distances into order, and then split the distances by finding the point where information gain is maximised. In addition to this implementation, shapelets have also been used in many other applications, such as early classification [17], gesture recognition [6] and as a filter transformation for TSC [11]. For the purpose of this work we do not focus on a specific application of shapelets, but rather we investigate the algorithm used for initially selecting shapelets.

In this paper we investigate the shapelet quality measure used for shapelet extraction by [19]. Whilst it lends itself neatly to a decision tree implementation, we feel that the use of information gain (IG) to assess shapelet candidates involves more computation than is necessary. In response to this, we introduce two new statistical tests into the context of measuring shapelet quality: Kruskal-Wallis (KW) and Mood's Median (MM) tests. We demonstrate the validity of KW and MM for shapelet discrimination in two stages; firstly, we show that there is no significant difference between shapelet tree classifiers built with KW and MM when compared to an IG implementation of [19]. Secondly, we demonstrate that the computation time of the generic shapelet finding algorithm can be reduced by using either KW or MM as the quality measure.

2 Time Series Classification

A time series is a sequence of data that is typically recorded in temporal order at fixed intervals. For the problem of time series classification, suppose we have a set of n time series $T = T_1, T_2, ..., T_n$, where each time series T_i has m real-value ordered readings $T_i =< t_{i,1}, t_{i,2}, ..., t_{i,m} >$ and a class label c_i. For simplicity, we assume that all series in T are of length m, but this is not a requirement for TSC. Given a set of data in the form of T, the problem of TSC is to find a function that maps from the space of possible time series to the space of possible class values. Whilst this problem is very similar to the general classification problem, TSC varies from standard approaches as it is often assumed that similarity between time series is to some extent embedded within the autocorrelation structure of the data.

As with all time series data mining, TSC relies to some degree on the use of a similarity measure to compare data. These typically fall into one of three broad categories: similarity in time (correlation-based); similarity in structure (autocorrelation-based); and similarity in change (shape-based). A detailed discussion of time series similarity can be found in [9] and [13]. Many shape-based applications of time series similarity use an elastic measure such as DTW with an instance based classifier (i.e. 1-NN with DTW). However, such an approach risks ignoring discriminatory shapes within a series as they may be masked by noise. This is one of the main strengths of shapelets for TSC; they allow a mechanism for identifying phase-independent shape-based similarity on a local level, unlike global measures such as DTW that must calculate similarity across entire series.

3 Shapelets

Shapelets were first introduced in [14] to provide a mechanism for measuring the similarity of time series using subsections that are particularly indicative of class membership. There are three main components of shapelet discovery: candidate generation; a distance measure between a shapelet and a time series; and a measure of shapelet quality.

3.1 Generating Candidates

A shapelet candidate is any contiguous subsequence S of length l within a time series T_i of length m, where $l \leq m$. A series of length m contains $(m-l)+1$ unique subsequences of length l. We denote the set of all subsequences of length l for series T_i to be $W_{i,l}$, and the set of all possible subsequences of length l for the data set to be $W_l = W_{1,l}, W_{2,l,}, ..., W_{n,l}$. The set of all candidates in T is $W = W_{min}, W_{min+1}, ..., W_{max}$ where $min \geq 1$ and $max \leq m$. For all possible lengths $l = 1, 2, ..., m$, there are a total of $m\frac{(m+1)}{2}$ shapelets in W. As this number can be very large with long series, [19] specify a minimum and maximum length parameter to constrain the search. The generic shapelet finding algorithm is defined in Algorithm 1.

Algorithm 1. ShapeletSelection (T, min,max)

```
1: bsfQuality = 0;
2: bestShapelet = ∅;
3: C = classLabels(T);
4: W = generateCandidates(T, min, max);
5: for l = min to max do
6:     for all subsequence S in W_l do
7:         D_S = findDistances(S, W_l);
8:         quality = assessCandidate(S, D_S);
9:         if quality > bsfQuality then
10:            bsfQuality = quality;
11:            bestShapelet = S;
12:        end if
13:    end for
14: end for
15: return bestShapelet;
```

Note that our implementation of Algorithm 1 independently normalises each element of W before using the distance function. We justify this as we are searching for local similarity between series, so wish to remove any offset caused by scale. Whilst no mention of this appears in [19], an amortised constant-time normalised distance measure is proposed in [14].

3.2 Shapelet Distance Calculations

The Euclidean distance between two subsequences S and R, where both are of length l, is calculated as:

$$dist(S,R) = \sum_{i=1}^{l}(s_i - r_i)^2. \quad (1)$$

The distance between a time series T_i and a subsequence S of length l is calculated using a sliding window to find the minimum distance between S and all possible subsequences in T_i of length l

$$d_{i,S} = \min_{R \in W_{i,l}} dist(S, R). \quad (2)$$

As $d_{i,S}$ is a minima, an early abandon is used to avoid unnecessary calculations. This calculation is used during shapelet extraction to calculate the distance from a candidate S to each time series in a data set T, $D_S = D_{S,1}, D_{S,2}, ..., D_{S,n}$, where n is the number of series in T. Note that [14] use a more efficient constant-time distance calculation based on maintaining a set of statistics. As the distance metric is incidental to the contribution of this paper, we retain the use of this simpler distance measure to keep the emphasis on shapelet quality measures.

4 Shapelet Quality Measures

The shapelet finding algorithm defined in Algorithm 1 requires an objective function for assessing shapelet quality, which is performed in [19] using information gain. This is where the main contribution of this paper lies; we believe that whilst information gain provides a good solution and lends itself neatly to decision trees, it involves an excessive amount of computation that could be removed by using different quality measures. Furthermore, additional applications of shapelets may not require an explicit split point to be found by the quality measure (such as a shapelet filter [11]). In such applications, it would be more appropriate to use alternative measures of quality that are faster. Therefore, we introduce Kruskal-Wallis and Mood's Median tests into the context of shapelet finding.

To assess the quality of shapelet S for data set T, a prerequisite of each quality measure is that the set of distances $D_S = D_{S,1}, D_{S,2}, ..., D_{S,n}$ must be calculated, where n is the number of instances in T and $D_{S,i}$ is the distance between S and instance i of T.

4.1 Information Gain

Information gain [16] (IG) is a non-symmetrical measure of the difference between two probability distributions. The shapelet finding algorithm in [19] uses IG as the quality measure to assess candidate shapelets. D_S is sorted and the information gain at each possible split point sp is assessed for S, where a valid split point is the average between any two consecutive distances in D_S. For each possible sp, IG is calculated by partitioning all elements of $D_S < sp$ into A_S, and all other elements into B_S. The information gain at sp is calculated as

$$IG(D_S, sp) = H(D_S) - \frac{|A_S|}{|D_S|}H(A_S) + \frac{|B_S|}{|D_S|}H(B_S), \quad (3)$$

where $|A_S|$ is the cardinality of the set A_S, and $H(A_S)$ is the entropy of A_S. Entropy is calculated by

$$H(D_S) = - \sum_{c \in classes\{D_S\}} p_c \log_2 p_c. \tag{4}$$

The IG $info_S$ of S is calculated as

$$info_S = \max_{sp \in D_S} IG(D_S, sp). \tag{5}$$

Note that [19] introduce an upper-bound for calculating IG. However, in this paper we do not implement the early abandon. We justify this for two reasons; firstly, in the most pessimistic cases for multi-class problems, the computation involved for implementing a naïve approach of the upper-bound would far out way the benefits provided by it. Secondly, the style of upper-bound used by [19] could also be implemented for KW and MM. We wish to directly compare the three quality measures, so by not using an early abandon, any implementation of the three quality measures must evaluate the same number of shapelet candidates.

4.2 Kruskal-Wallis

Kruskal-Wallis [10] (KW) is a non-parametric test to observe whether data originates from a single distribution. The calculated statistic represents the squared-weighted difference between ranks within a class and the global mean rank. For use with shapelets, KW is calculated for S as

$$KW_S = \frac{12}{|D_S| \cdot (|D_S| + 1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(|D_S| + 1), \tag{6}$$

where $|D_S|$ is the cardinality of D_S, k is the number of classes in D_S, R_i is the sum of ranks for class i and n_i is the number of instances of class i in D_S. Note that in order to calculate ranks, D_S must be sorted as it was with IG. However, we believe that KW will be more efficient for shapelet finding than IG because the statistic only needs to be calculated once, rather than for each possible split point in D_S.

4.3 Mood's Median

Mood's Median [12] (MM) is a non-parametric test to determine whether the medians of two samples originate from the same distribution. Unlike IG and KW, MM does not require D_S to be sorted, so therefore should be faster. Only the median is required for calculating MM, which can be found in $O(n)$ time using quickselect [7]. The median is used to create a contingency table from D_S, where the counts of each class above and below the median are recorded. The MM statistic is obtained by calculating the Chi-Squared statistic of the table

$$\chi^2 = \sum_{j=1}^{c} \sum_{i=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}, \tag{7}$$

where r and c are the rows and columns of the contingency table and o_{ij} and e_{ij} are the observed and expected values of row r, column c respectively.

5 Experimental Procedure

The experiments in this paper are designed to establish the validity and advantages of using KW and MM for shapelet finding. This is demonstrated in two stages; firstly, we use a diverse range of data to build shapelet decision trees akin to [19] using IG, KW and MM as quality measures, and show that the classifiers produced are not significantly different. Secondly, we perform timing experiments to show the relative time performance of KW and MM compared to IG for finding the most discriminatory shapelet in a data set.

5.1 Shapelet Classifier Implementation

We implement four distinct shapelet tree classifiers; the first uses IG as the shapelet quality measure as in [19], the second uses KW, and the final two use MM. We slightly modify the algorithm for KW and MM classifiers due to the nature of the statistics calculated. In the KW tree, we initially find the best shapelet by replacing IG with KW. However, unlike IG, the KW statistic is only calculated once across the set of distances and no split point is implied for classification. To accommodate this, we establish the best shapelet using KW and then use a single set of IG calculations to find the best split point. We justify this because the costly IG calculations for each candidate are replaced by KW, and IG is only used once for the best shapelet. For MM, we implement two classifiers; the first simply uses the median from the MM calculation of the best shapelet as the split point, whilst the second classifier uses the same approach as the KW classifier to identify the final split point using a single set of IG calculations.

The minimum and maximum shapelet lengths for each data set were computed using the simple cross-validation approach in [11]. The parameters vary across data sets, but are consistent for each classifier to ensure that they each evaluate the same number of candidates and are directly comparable.

6 Results

The results that we report are split into two sections. Firstly, we wish to demonstrate that KW and MM are valid statistics for measuring the quality of shapelets. We demonstrate this through a number of classification experiments and report the error rates accross a diverse range of data sets. Secondly, we wish to demonstrate that these new quality measures speed up shapelet discovery. We demonstrate this with a number of timing experiments using the same data sets.

6.1 Classification Performance

The results in Table 1 show that whilst the IG classifier achieves the top rank on more data sets than any other classifier (10 of 26), it is in fact the MM with IG tree that has the best overall rank. The KW tree also has a better overall rank than IG, whilst MM using the median to split has the lowest overall rank. This supports our

decision to use IG to find the best split point in the MM tree. To further demonstrate the validity of KW and MM as quality measures, we show that there is no significant difference between the classifiers in Figure 1 using a critical difference diagram (as decribed by [3]). The diagram is derived from the overall test of significance of mean ranks, where classifiers are grouped into *cliques* that are represented by the solid bars. The diagram shows that all classifiers are part of a single clique, and therefore are not significantly different from one another. This supports our claim that MM and KW are valid metrics of shapelet quality.

Table 1. Classification error rates for the shapelet tree classifiers

Data Set	IG	KruskalWallis	MoodMedian	MoodMedIG
Adiac	**0.7008(1)**	0.734(3)	0.7928(4)	0.7289(2)
Beef	**0.5(1)**	0.6667(2.5)	0.6667(2.5)	0.7(4)
ChlorineConcentration	**0.412(1)**	0.474(3)	0.4648(2)	0.4789(4)
Coffee	**0.0357(1)**	0.1429(3)	0.1429(3)	0.1429(3)
DiatomSizeReduction	**0.2778(1)**	0.3889(2)	0.5392(3)	0.5523(4)
DP_Little	0.3456(4)	0.32(3)	**0.2567(1)**	0.29(2)
DP_Middle	0.2947(2)	0.3067(3)	0.35(4)	**0.2633(1)**
DP_Thumb	0.4189(4)	**0.28(1)**	0.3233(3)	0.2967(2)
ECGFiveDays	0.2253(4)	0.1278(2)	0.1568(3)	**0.072(1)**
ElectricDevices	0.451(3)	**0.4416(1)**	0.4492(2)	0.5317(4)
FaceFour	**0.1591(1)**	0.5568(2)	0.5795(3)	0.5909(4)
GunPoint	0.1067(4)	**0.06(1)**	0.1(3)	0.08(2)
ItalyPowerDemand	0.1079(3)	0.0904(2)	0.1322(4)	**0.0894(1)**
Lighting7	**0.5068(1)**	0.5205(2)	0.7671(4)	0.726(3)
MedicalImages	0.5118(3)	0.5289(4)	**0.5(1)**	0.5105(2)
MoteStrain	0.1749(4)	**0.1605(2)**	**0.1605(2)**	**0.1605(2)**
MP_Little	0.3361(4)	0.3033(3)	**0.2667(1)**	0.2967(2)
MP_Middle	0.2899(4)	**0.25(1)**	0.2867(3)	0.28(2)
PP_Little	0.4036(4)	**0.28(1)**	0.3433(3)	0.3267(2)
PP_Middle	0.3858(4)	0.3167(3)	0.31(2)	**0.3033(1)**
PP_Thumb	0.3917(4)	0.2867(3)	**0.2667(1)**	0.27(2)
SonyAIBORobotSurface	**0.1547(1)**	0.2729(4)	0.2479(2)	0.2512(3)
Symbols	**0.2201(1)**	0.4432(4)	0.4201(2)	0.4261(3)
SyntheticControl	**0.0567(1)**	0.1(2)	0.1867(4)	0.1433(3)
Trace	0.02(2)	0.06(3)	0.08(4)	**0(1)**
TwoLeadECG	0.1493(3)	0.2362(4)	**0.1343(1)**	0.1466(2)
Mean Rank	2.5385	2.4808	2.5962	2.3846

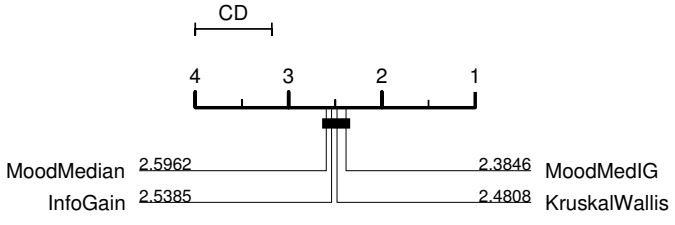


Fig. 1. Critical difference diagram for the four different shapelet tree classifiers

6.2 Timing Results

The results in Table 2 were produced using IG, KW and MM to find the best shapelet from each data set. This approach was adopted to ensure fair comparisons could be made between measures, as comparing the build times of whole

Table 2. Relative computation times of KW and MM against IG

Data Set	Kruskal-Wallis	Mood's Median
Adiac	27.23%	26.44%
Beef	102.81%	98.15%
ChlorineConcentration	61.34%	57.35%
Coffee	102.17%	97.16%
DiatomSizeReduction	102.11%	98.03%
DP_ Little	93.19%	89.22%
DP_ Middle	53.39%	51.03%
DP_ Thumb	94.98%	90.34%
ECGFiveDays	99.85%	100.21%
ElectricDevices	79.78%	75.87%
FaceFour	104.71%	101.29%
GunPoint	103.77%	101.20%
ItalyPowerDemand	50.81%	49.03%
Lighting7	98.74%	96.25%
MedicalImages	51.48%	23.55%
MoteStrain	101.49%	94.57%
MP_ Little	95.75%	90.32%
MP_ Middle	98.51%	93.37%
PP_ Little	93.95%	90.08%
PP_ Middle	94.97%	90.00%
PP_ Thumb	95.08%	89.82%
SonyAIBORobotSurface	93.32%	97.15%
Symbols	102.90%	101.32%
SyntheticControl	44.21%	41.31%
Trace	97.53%	101.59%
TwoLeadECG	91.35%	90.99%
Average	85.98%	82.14%

decision trees would be biased if the classifiers were of different depths. Extracting a single shapelet ensures that the same number of candidates are processed for each quality measure.

Table 2 shows that there are few cases where IG is fastest, and even in these cases the difference is marginal. It is clear that KW and MM perform much better on some data sets whilst providing at least a modest speedup on the majority of cases. MM is the fastest overall and provides almost an 18% speed-up over IG, whilst KW also provides a marked improvement of approximately 14% on average. On first glance this may not seem significant, but shapelet extraction can be time consuming and can potentially take hours in some cases, so an improvement of almost 20% may be critical in some applications.

7 Conclusions and Future Work

In this paper we have introduced two new quality measures for shapelet extraction in TSC. We have demonstrated the effectiveness of the Kruskal-Wallis and Mood's Median statistics as discriminatory measures by using them to build shapelet decision tree classifiers in the style of [19]. We use these classifiers to illustrate two points. Firstly, using these alternative measures does not degrade the discriminatory power of the shapelets that are extracted. This is demonstrated by producing classifiers that are shown to not be significantly different over 26 data sets. Secondly, we limit the shapelet finding algorithm to extract

only the best shapelet from each data set, allowing us to directly compare the computation times of the three statistics. Our results show an average improvement in computation time across the 26 data sets of approximately 14% and 18% for Kruskal-Wallis and Mood's Median respectively. With a view to the future, we can investigate the potential of these alternative quality measures in further applications of shapelets, such as extending a shapelet filter for TSC [11].

References

1. Bagnall, A., Davis, L., Hills, J., Lines, J.: Transformation based ensembles for time series classification. In: Proc. 12th SDM (to appear, 2012)
2. Buza, K.: Fusion methods for time-series classification. Ph.D. thesis, University of Hildesheim, Germany (2011)
3. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. JMLR 7, 1–30 (2006)
4. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. Tech. report, Arizona State University (2011)
5. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: Experimental comparison of representations and distance measures. In: Proc. 34th VLDB (2008)
6. Hartmann, B., Link, N.: Gesture recognition with inertial sensors and optimized DTW prototypes. In: Proc. IEEE SMC (2010)
7. Hoare, C.A.R.: Quicksort. The Computer Journal 5(1), 10–16 (1962)
8. Jeong, Y., Jeong, M., Omitaomu, O.: Weighted dynamic time warping for time series classification. Pattern Recognition 44, 2231–2240 (2010)
9. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Mining and Knowledge Discovery 7(4), 349–371 (2003)
10. Kruskal, W.H.: A nonparametric test for the several sample problem. The Annals of Mathematical Statistics 23(4), 525–540 (1952)
11. Lines, J., Davis, L., Hills, J., Bagnall, A.: A shapelet transform for time series classification. In: Proc. 18th ACM SIGKDD (2012)
12. Mood, A.M.F.: Introduction to the theory of statistics. McGraw-hill (1950)
13. Mörchen, F., Mierswa, I., Ultsch, A.: Understandable models of music collections based on exhaustive feature generation with temporal statistics. In: Proc. 12th ACM SIGKDD, pp. 882–891 (2006)
14. Mueen, A., Keogh, E.J., Zhu, Q., Cash, S., Brandon Westover, M.: Exact discovery of time series motifs. In: SDM, pp. 473–484 (2009)
15. Rodriguez, J., Alonso, C.: Support vector machines of interval-based features for time series classification. Knowledge-Based Systems 18, 171–178 (2005)
16. Shannon, C.E.: A mathematical theory of communication. Bell System Technichal Journal 27, 379–423, 623–656 (1948)
17. Xing, Z., Pei, J., Yu, P., Wang, K.: Extracting interpretable features for early classification on time series. In: Proc. 11th SDM (2011)
18. Ye, L., Keogh, E.: Time series shapelets: A new primitive for data mining. In: Proc. 15th ACM SIGKDD (2009)
19. Ye, L., Keogh, E.: Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. Data Min. Knowl. Discov. 22(1-2), 149–182 (2011)

Using SOM to Clustering of Web Sessions Extracted by Techniques of Web Usage Mining

Fábio A. Procópio de Paiva[1] and José Alfredo F. Costa[2]

[1] IFRN, Zona Norte Campus, Natal, Brazil
fabio.procopio@ifrn.edu.br
[2] UFRN, Department of Electrical Engineering, Natal, Brazil
jafcosta@gmail.com

Abstract. Everyday a huge amount of pages are published on the Web, and, as a consequence, the users' difficulty to locate those that will meet their needs is increasingly bigger. The challenge for web designers and e-commerce companies is to identify groups of users that present similar interests in order to personalize navigation environments to meet those interests. In an attempt to offer that to the countless web users, in the last years, several researches have been done on clustering applied to Web Usage Mining. In this paper, a log file is preprocessed to map the sequence of visits for each user's session. A Session-Path Matrix is used as input to SOM Map and identifying patterns between each session. The results show the similarities between the sessions based on time spent on visited paths and volume transferred.

Keywords: Web Usage Mining, SOM, users' sessions, clustering, navigation path.

1 Introduction

The available content on WWW is stored in many formats (audio, images, text and others), covers different areas of knowledge and includes users with multiple interest profiles. However, the challenge is to provide rapid access to information and, above all, make them relevant to the users' interests. Thus, the WWW appears as an ideal environment to apply Data Mining techniques, known as Web Mining.

Web Mining consists in extracting interesting and potentially useful patterns and implicit information from artifacts or activity related to the WWW. Moreover, Web Mining is one of the most important areas of Computer Science and Information Science [9]. The Web Mining is classified in three categories [13]: Web Content Mining, Web Structure Mining and Web Usage Mining. The Web Content Mining is the process that extracts knowledge from the Web and analyzes the contents of its documents. Web Structure Mining tries to discover the model underlying the link structures of the Web. And finally, the purpose of Web Usage Mining is to apply statistical and data mining techniques to the preprocessed web log data, in order to discover useful patterns [1].

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 484–491, 2012.

Nowadays Web Usage Mining is an area of interest for many researchers [13] because a) the record of accessed pages allows mapping the users' behavior; b) the frequent accesses can be used to improve link structure and; c) it allows suggesting changes in pages design.

In reference to [12], an important point to be observed in Web Usage Mining is the users' clustering according to their characteristics. From an analysis of clusters, a web designer can identify the users' interest and thus offer more personalized services to a group of them. Still, according to [14], the results generated by clustering techniques can be used to analyze the systems' performance and network communication. A method to cluster users is measuring the similarity between them based on their interests. There are several measures that can be used [12]: Usage Based Measure, Frequency Based Measure, Viewing-time Based Measure and Visiting-Order Based Measure.

In literature, there are several researches that propose methods to identify web users' interests, although this is a complex task [8]. According to [6], the authors considered the time as a good measure to evaluate the users' interest and used the naïve Bayes method to model and predict the users' navigational behavior. A model based on ant colonies has been proposed by [3] to identify users' browsing patterns. In this model, the authors used access frequency and the time spent as measures to identify the users' interest. In reference to [5], the authors proposed a method based on Self-Organizing Maps that uses Web Content Mining and Web Usage Mining clustering techniques to help visitors identify relevant information quickly.

In this paper, a log file is preprocessed to map the sequence of visits (i.e. path) for each user's session. Then a Session-Path Matrix is used as input data to Self-Organizing Map (SOM) and to identify patterns between each session. The results show the similarities between the sessions based on time spent on visited paths and volume transferred.

This paper is organized as it follows: section 2 presents the concepts of Web Usage Mining; Section 3 describes briefly the Self-Organizing Maps (SOM); Section 4 details the data preprocessing and sessions' clustering using the SOM. The last section presents conclusions and future research directions.

2 Web Usage Mining

Web Usage Mining is a Data Mining process used to discover usage patterns of the information on the Web and aims to provide an understanding of the interests and behavior of web users[10]. Thus, analyzing the user's access logs on websites, it is possible to understand their actions and thus enable customization of a navigation environment.

Often, Web Usage Mining includes the following steps [2]:

(a) Preprocessing – removes inconsistencies and noise of the data sources in order to leave only those that are really significant. The stage also includes tasks such as users' identification, sessions' identification and definition of the full path of navigation [9].

(b) Pattern discovery – discovers the user's interests (or a group of them) and build a model according to these preferences.
(c) Pattern Analysis – the main objective is to filter the information that are apparently irrelevant to viewing and interpreting the user interest patterns.

When a user navigates on a website his/her interactions are recorded in files called web server log file. This record has the form of a single transaction and is appended in ASCII text file. Nowadays there are three ways to obtain user's access logs: a) client log file, b) proxy log file and c) server log file. The delimiter of this file type can be a comma, a blank or a tab.

There are three types of server log file available to capture the activities of a user on websites [4]: Common Log Format, Log Format and Extended Log Format IIS. This work is based on a server log file using Common Log Format.

The most common and simple way to analyze a log file is using a statistical method. However, there are more sophisticated methods, such as Association Rules Mining, Sequential Pattern Discovery, Clustering and Classification [1].

3 Self-Organizing Maps (SOM)

The Self-Organizing Map (SOM) is one of the most popular artificial neural network algorithms and it is based on unsupervised competitive learning, which means that the training is entirely data-driven. The training of the neurons present competitive and cooperative processes [15]. The SOM defines a mapping from the high dimensional input data space onto a regular, usually, two-dimensional array of nodes. Each neuron i of the SOM is represented by an p-dimensional weight vector $m_i = [m_{i1}, m_{i2}, ..., m_{ip}]^T$, where p is equal to the dimension of the input vectors[16].

3.1 SOM Properties

Assume that $\Re^p$ is an input space with a topology defined by the metric relation between the vectors $\boldsymbol{x} \in \Re^p$. Consider K a discrete output space with a topology that is defined by arranging a set of neurons like nodes on a grid. According to Haykin [21], the SOM algorithm can be defined as a non-linear transformation, Φ, called a *feature map*, which maps the input space $\Re^p$ to the output space K:

$$\Phi: \Re^p \rightarrow K$$

Given an input vector $\boldsymbol{x}$, the SOM algorithm identifies the winner neuron c in the output space K according to a features map Φ. The main properties of feature mapping computed by the SOM include [22]:

1. Approximation of the input space – the features map Φ, represented by the set of code vectors $\boldsymbol{w}_i$ in the output space K, provides a good approximation of the input space $\Re^p$. This strategy is based on the vector quantization theory, the motivation for which is data compression [21];
2. Topological ordering – the SOM algorithm attempts to preserve as well as possible the topology of the original space, i.e., it tries to make the neighboring

neurons in the rectangular grid (output space K) present weight vectors that represent neighboring patterns in the input space $\Re^p$.

3. Density matching – when properly trained, the features map Φ approximates the probability distribution of data in the input space $\Re^p$.

4 Data Preprocessing

The log files cannot be used before a treatment [7] because they have some records which do not add any value to the Data Mining. However, it makes it difficult to analyze the users' behavior. These irrelevant data are called noise and usually are generated by web robots or when images, videos, audios, CSS, javascripts and flash animations are loaded within the pages of a website.

The access logs used in this paper did not show which user made a particular access, thus users were identified from the recorded IP address into a log file. However, this does not prevent that the same IP can be used by different users or even the same user to different IPs. Thus, the analysis was performed on the basis of access sessions. In this context, a session consists of an access to a page (or set of them) recorded to the same IP and the time difference between the instants t_i and t_{i-1} (where t_i is the time that the page p_j was accessed and t_{i-1} the access time to the page p_{j-1}) is less than or equal to 30 minutes [11][18]. The method which is often used to distinguish two sessions is setting the time of timeout. Many web usage analysts and commercial applications set the timeout threshold at 30 minutes [7][19].

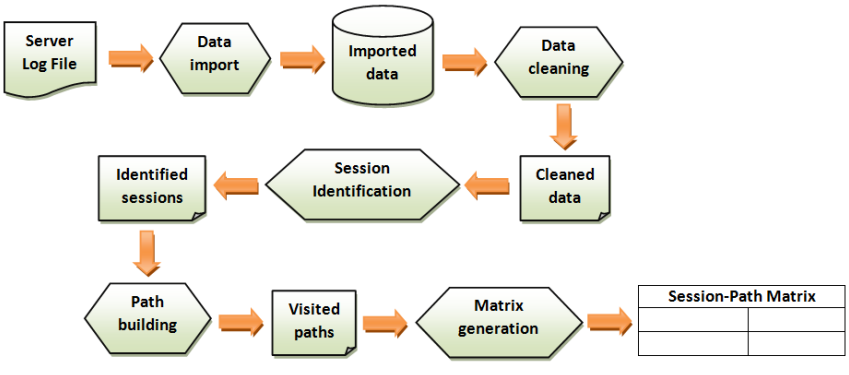


Fig. 1. Data preprocessing to Session-Path Matrix generation

As it was mentioned before, data from a log file cannot be used before being preprocessed. Fig. 1 shows the procedures used for this. Next, each step is described:

1. Data import – imports logs from a text file to database. To facilitate the data manipulation, the logs were imported into SQL Server 2008 R2 Express.
2. Data cleaning – performs noise removal from original data. After importing data, unsuccessful requests logs have been removed. In addition, records of images, videos, audios, javascript and flash animations.

3. Session identification – aims to identify the user's sessions. The data were sorted by timestamp field. When the user (identified by IP) of the r_i record is different from r_{i-1} record or when the difference between the timestamps of r_i and r_{i-1} is greater than 30 minutes a new session is created.
4. Path building – consists of mapping the full path (sequence of visits to pages) of each session, calculating the time spent and the transferred data volume.
5. Session-Path Matrix – creates a matrix containing session identifier, path identifier, time spent on each page and transferred volume.

The result of preprocessing produces the Session-Path Matrix, Fig. 1, in which rows represent sessions and columns are visited path information by each session.

5 Sessions Clustering

After preprocessing we use the data as input to SOM Toolbox, a SOM library developed to Matlab. Based on Path-Session Matrix, a file sample is generated by preprocessing script.

In reference to [12] the authors highlighted Visiting-Order as a measure to evaluate degree of interest of web users and this contributed to what we assume that sequence of pages accessed by users in a user's sessions (i.e. path) is a important variable for measuring the similarity in the degree of web sessions.

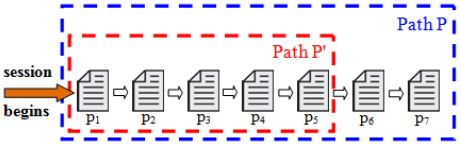


Fig. 2. Path P' is a segment of Path P

We have used access-logs of the University of Saskatchewan (available at http://ita.ee.lbl.gov/html/traces.html), located in Saskatoon, Saskatchewan, Canada. After preprocessing, we have extracted 125 records from Session-Path Matrix and such records are divided into 5 classes, each with 25 instances of visited paths. The classes indicate the paths visited by sessions: Path6, Path9, Path13, Path29 and Path34. Here, P is the full path visited by a session S. Eventually P can be segmented into smaller path called P'. In Fig. 2, e.g., P is composed of 7 pages (p_1, p_2, p_3, p_4, p_5, p_6 and p_7) and S begins when page p_1 is accessed. From path P, path P' has been created which is a shortest length path containing 5 pages (p_1, p_2, p_3, p_4 and p_5). In our experiments, we evaluate paths with length greater than or equal to 5 pages because they presented a higher representation of data. When paths P presented length greater than 5 pages, we generated a path P' of length 5 in order to compare paths only of the same length. The attributes defined for each instance are:

- timeToPg2: elapsed time between accesses page p_1 and page p_2;
- timeToPg3: elapsed time between accesses page p_1 and page p_3;

- timeToPg4: elapsed time between accesses page p_1 and page p_4;
- timeToPg5: elapsed time between accesses page p_1 and page p_5 and;
- volume: volume transferred (bytes) in P'.

In order to visualize the results produced by SOM Toolbox, we set parameters for initialization, for training and for visualization of the map. The evaluation was performed in several experiments using the combination of parameters.

Table 1. Configuration parameters of the SOM

Parameter	Value
Data normalization	Variance
Inicialization	Linear
Topology	Hexa
Neighborhood function	Gaussian
Training	Batch
Epochs	3000
Dimension X / Dimension Y	15 / 15

To measure the map generation quality after training, we use two metrics: Quantization Error (Qe) and Topographical Error (Te). Quantization Error measures the average distance between each data vector and its best matching unit [20] and Topographical Error indicates how much the SOM preserving topology of the input data. Based on the parameters defined, in Table 1, the results obtained for the two measurements were Qe = 0.2533 and Te = 0.0400.

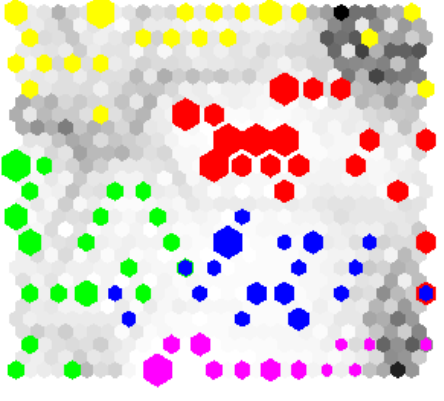


Fig. 3. U-Matrix

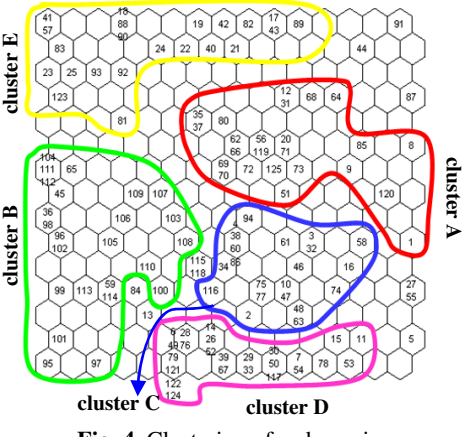


Fig. 4. Clustering of web sessions

After training, the map is displayed by U-Matrix in which different colors are used to represent the distances between neurons. The U-Matrix is commonly used visualization method for the cluster analysis using SOM and based on the distance in input space between a weight vector and its neighbors on map [17]. In U-Matrix a light

shade means that they are close and a darker shade can be interpreted as a clusters separator.

We also use the histogram hits that allow the identification of the parts of the map that best represent the data. In Fig. 3, the U-Matrix is colored in 5 different classes, each representing a sessions' cluster. In Fig. 4, the labels, such as 41 and 57 (cluster E), represent the sessions' identifiers. We draw the clusters' line manually and we noted that:

- Cluster A – represents the users' sessions that accessed Path 6;
- Cluster B – represents the set of users' sessions that browsed on Path 9;
- Cluster C – represents the users' sessions that accessed Path 13;
- Cluster D – represents the set of users' sessions that browsed on Path 29;
- Cluster E – represents the users' sessions that accessed Path 34.

6 Conclusion

The Internet popularization has had an impact on the number of pages published on the network. The amount of available information configures a huge source of data and the difficulty encountered by web users to find resources that meet their interests is a problem that some areas such as Information Retrieval and Web Mining have been trying to solve.

In this paper, five steps were performed to preprocess data extracted from a log file: Data import, Data cleaning, Session identification, Path building and Session-Path Matrix building. In order to visualize the similarities between users' sessions, the Session-Path Matrix has been used as an input data to Kohonen's Map. The figures 3 and 4 show the existence of similar patterns between sessions that were browsed by the same path.

Future works will focus ontologies to assign semantics to the pages visited and therefore the browsed paths. Furthermore, such works will create a mechanism to enable the similar paths recommendation to analyzed sessions.

References

1. Eirinaki, M., Vazirgiannis, M.: Web Mining for Web Personalization. ACM Transactions on Internet Technology 3, 1–27 (2003)
2. Etminani, K., Delui, A.R., Yanehsari, N.R., Rouhani, M.: Web usage mining: Discovery of the users' navigational patterns using SOM. In: First International Conference on Networked Digital Technologies, pp. 224–249. IEEE Press, New York (2009)
3. Ling, H., Liu, Y., Yang, S.: An Ant Colony Model for Dynamic Mining of Users Interest Navigation Patterns. In: IEEE International Conference on Control and Automation, pp. 281–283. IEEE Press, New York (2007)
4. Hussain, T., Asghar, S., Masood, N.: Web usage mining: A survey on preprocessing of web log file. In: International Conference on Information and Emerging Technologies, pp. 1–6. IEEE Press, New York (2010)

5. Petrilis, D., Halatsis, C.: Combining SOMs and Ontologies for Effective Web Site Mining. In: Self Organizing Maps - Applications and Novel Algorithm Design, pp. 109–124. In-Tech, Rijeka (2011)
6. Khosravi, M., Tarokh, M.J.: Dynamic mining of users interest navigation patterns using naive Bayesian method. In: IEEE International Conference on Intelligent Computer Communication and Processing, pp. 119–122. IEEE Press, New York (2010)
7. Markov, Z., Larose, D.T.: Data Mining the Web – uncovering patterns in Web Context, Structure, and Usage. Wiley, New Jersey (2007)
8. Nadi, S., Saraee, M., Davarpanah-Jazi, M.: A fuzzy recommender system for dynamic prediction of user's behavior. In: International Conference on Internet Technology and Secured Transactions, pp. 1–5. IEEE Press, New York (2010)
9. Pamnani, R., Chawan, P.: Web Usage Mining: A Research Area in Web Mining. In: Proceedings of ISCET 2010, pp. 73–77 (2010)
10. Vellingiri, J., Pandian, S.C.: A Survey on Web Usage Mining. Global Journal of Computer Science and Technology, 66–72 (2011)
11. Xiao, J., Zhang, Y.: Clustering of web users using session-based similarity measures. In: International Conference on Computer Networks and Mobile Computing, pp. 223–228. IEEE Press, New York (2001)
12. Xiao, J., Zhang, Y., Jia, X., Li, T.: Measuring similarity of interests for clustering Web-users. In: 12th Australasian on Database Conference, pp. 107–114. IEEE Press, New York (2001)
13. Pani, S.K., Panigrahy, L., Sankar, V.H., Ratha, B.K., Mandal, A.K., Padhi, S.K.K.: Web Usage Mining: A Survey on Pattern Extraction from Web Logs. International Journal of Instrumentation, Control & Automation 1(1) (2011)
14. Wang, S., Xu, C., Wu, R.: Clustering Method Based on Fuzzy Multisets for Web Pages and Customer Segments. In: International Seminar on Business and Information Management, vol. 2, pp. 125–128. IEEE Press, New York (2008)
15. Kohonen, T.: Self-Organizing Maps, 2nd edn. Springer, Berlin (1997)
16. 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)
17. Yamaguchi, T., Ichimura, T.: Visualization using multi-layered U-Matrix in growing Tree-Structured self-organizing feature map. In: IEEE International Conference Systems, Man and Cybernetics (SMC), pp. 3580–3585 (2011)
18. Cheng, X., Liu, H.: Personalized Services Research Based on Web Data Mining Technology. In: Second International Symposium on Computational Intelligence and Design, vol. 2, pp. 177–180 (2009)
19. Dong, Y., Zhang, H., Jiao, L.: Research on Application of User Navigation Pattern Mining Recommendation. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, vol. 2, pp. 6106–6110 (2006)
20. Uriarte, E.A., Martín, F.D.: Topology Preservation in SOM. International Journal of Mathematical and Computer Sciences, 19–22 (2005)
21. Haykin, S.: Neural networks: A comprehensive foundation, 2nd edn. Macmillan College Publishing Company, N. York (1999)
22. Gonçalves, M., Netto, M., Zullo Jr., 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)

A Comparative Study of Use of Shannon, Rényi and Tsallis Entropy for Attribute Selecting in Network Intrusion Detection

Christiane F.L. Lima, Francisco M. de Assis, and Cleonilson Protásio de Souza

[1] Federal Institute of Maranhão, Department of Education
São Luís, MA - Brazil
[2] Federal University of Campina Grande, Department of Electrical Engineering
Campina Grande, PB - Brazil
[3] Federal University of Paraíba, Department of Electrical Engineering
João Pessoa, PB - Brazil

Abstract. Intrusion Detection Systems of computer networks carry out their detection capabilities observing a set of attributes coming from the network traffic. Such a set may be very large. However, some attributes are irrelevant, redundant or even noisy, so that their usage may also decrease the detection intrusion efficiency. Therefore, the primary problem of identifying an optimal attribute subset is the choice of the criterion to evaluate a given attribute subset. In this work, it is presented an evaluation of Rényi and Tsallis entropy compared with Shannon entropy in order to obtain an optimal attribute subset which increases the detection capability to classify the traffic as normal or as suspicious. Additionally, we studied an ensemble approach that combines the attributes selected by Rényi, Tsallis and Shannon information measures. The empirical results demonstrated that by applying an attribution selection approach based on Rényi or Tsallis entropies not only do the number of attributes and processing time are reduced but also the clustering models can be builded with a better performance (or at least remains the same) than that built with a complete set of attributes.

Keywords: Attribute selection, network intrusion detection, Shannon, Rényi and Tsallis entropy.

1 Introduction

According to [1], a network intrusion is defined as a set of actions that can compromise the integrity, confidentiality or availability of resources in a network context. Complete or partial intrusions take place as a result of successful attacks which exploit system vulnerabilities. Since it is impossible to achieve invulnerable network system, it is more appropriate to assume that intrusions can happen. Hence, the central challenge with computer security is determining the difference between normal and potentially harmful activity.

Alongside other techniques for preventing intrusions such as encryption and firewalls, intrusion detection systems (IDSs) are software systems designed with the purpose of identifying and preventing unauthorised use, misuse and abuse of computer networks and systems.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 492–501, 2012.

In intrusion detection, enormous amounts of data are collected from the network, generating large log files and raw network traffic, which make human inspection impossible. This poses a great challenge. Thus, these activities need to be summarized into higher-level events, described by some attributes (features). Therefore, selecting relevant attributes is a crucial activity and requires extensive domain knowledge.

The selection of an optimal reduced subset of attributes is essential in: (i) removing irrelevant and redundant data; (ii) reducing the use of resources; (iii) increasing detection precision and (iv) achieving rapid and effective response against attacks. Although such problems have been tackled by researchers for many years, there has been recently a renewed interest in feature extraction. Thereby, according to [2], the identification of a representative set of attributes is a main problem in IDS in order to both optimize the effectiveness of intrusion detection and decrease the complexity of the IDS.

In this work, we investigated four different approaches to select optimal attributes. Firstly, we considered a mofified *gain ratio* that incorporates Rényi [3] and Tsallis [4] establishing a comparison with Shannon [5] information measure criteria for constructing C4.5 decision trees [6]. Additionally, we studied an ensemble approach that combines the attributes selected by Rényi [3], Tsallis [4] and Shannon [5] information measures. These schemes were applied to a data set for network intrusion detection based on KDD Cup 1999 data [7].

In order to evaluate their clustering performance on the smaller subsets of attributes selected using various approaches, we considered different models using two clustering algorithm: SimpleKMeans [8]and FarthestFirst [9]. The experimental results demonstrate that the clustering performance of the models built with smaller subsets of attributes is comparable and sometimes better than that associated with the complete set of attributes for DoS and Probing attack categories.

The remainder of the paper is organized as follows. Section 2 describes the scheme to select attributes based on C4.5 decision tree algorithm. The experimental environment is explained in Section 3. Results are reported in Section 4 and Conclusions are drawn in Section 5.

2 Selection of Attributes Based on Decision Tree Algorithm

Attribute selection is a strategy for data reduction process since irrelevant and redundant attributes often degrade the performance of algorithms devoted to data characterization, rule extraction and construction of predictive models, both in speed and in prediction accuracy. The goal of the attribute selection process is, given a dataset that describes a target concept using N attributes, to find the minimum number M of relevant attributes which describe the concept as well as the original set of attributes does, in such a way that characteristic space is reduced according to some criterion [10].

Attribute selection algorithms can fall into two broad categories: the filter model or the wrapper model. The filter model tries to choose an attribute subset independently from the learning algorithm to be used, by examining the intrinsic characteristics of the data and by estimating the quality of each attribute considering just the avaliable data. In contrast, the wrapper model evaluates the goodness of the subset of attributes by applying a predetermined learning algorithm on the selected subset of attributes.

So, for each new subset of attributes, the wrapper model needs to learn a classifier and uses its performance to evaluate and determine which subset of attributes are selected. This approach tends to find attributes best suited to the predetermined classification algorithm resulting in superior learning performance, but it also tends to be more computationally expensive than the filter model [11].

For N attributes, there are 2^N possible subsets. An exhaustive search for an optimal subset of attributes can be impracticable, especially when N and the number of data classes increase. Therefore, heuristic methods that explore a reduced search space are commonly used for attribute subset selection. These methods are typically greedy in the sense that, while searching through attribute space, they always make what seems to be the best choice at the time. Their strategy is to make a locally optimal choice in the hope that this will lead to a globally optimal solution. Such greedy methods are effective in practice and may come close to estimating an optimal solution [10].

Decision trees (DTs) are originally known to be effective classifiers in a variety of domains. Most of the decision tree algorithms developed have used a standard top-down greedy approach to building trees. Decision tree induction is the learning of decision tree classifiers and it uses the training data, which is described in terms of the attributes. It constructs a directed graph, where each internal node (non leaf node) denotes the test on the attribute, a branch represents an outcome of the test and a leaf node corresponds to a class label (see Figure 1).

In our attribute selection approach, a decision tree induction is used for selecting relevant attributes. All attributes that do not appear in the tree are assumed to be irrelevant. So, the set of attributes appearing in the tree represents the subset of selected attributes. 1.

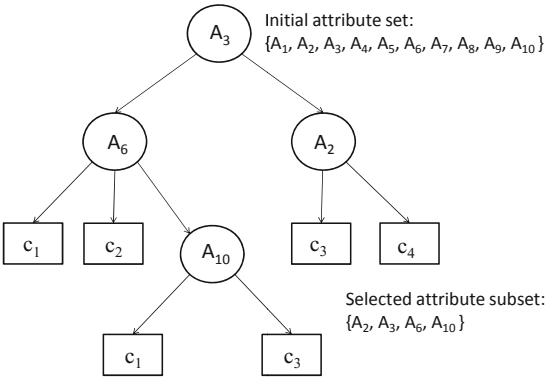


Fig. 1. Decision tree induction for attribute selection

The most popular DTAs are the *ID3* (Induction of Decision Tree) [12] and its successor, the *C4.5* [6] algorithm. Using a top-down process, both algorithms are capable of building decision trees by selecting appropriate attribute for each decision node based on Shannon's entropy measure [5]. For each iteration, the best attribute is that one with highest mutual information [5] among all others. This basic criterion is used in ID3

algorithm to select attributes while the tree is being designed. However, although presenting good results, it has a strong bias in favor of attributes with many values. To solve this problem, Quinlan [6] proposed in the C4.5 algorithm a kind of normalization, called *gain ratio*, in which the apparent gain assigned to attributes with many values is adjusted. For more details on the general algorithm for building C4.5 decision trees based on Shannon, Rényi and Tsallis entropies, see paper [13].

In this context, there are other entropy measures, such as Rényi and Tsallis entropies, that could be applied to select the attribute subset. Thus, we propose to investigate whether these measures can be adequately used in this problem. In the following, each entropy formulation is duly described.

The motivation to apply these entropies rises from the remark that for $\alpha > 1$ more frequent events are emphasized [14] and the limitation of the size of data set from which one has to capture the more adequated tree atributes.

2.1 Shannon Entropy

The concept of entropy is related with the amount of information into a message as a statistical measure. Based on the work of Shannon [5], given a class random variable C with a discrete probability distribution $\{p_i = \Pr[C = c_i]\}_{i=1}^{k}, \sum_{i=1}^{k} p_i = 1$ where c_i is the *i*th class. Then entropy $H(C)$ is an expected amount of information needed for class prediction, defined as

$$H(C) = -\sum_{i=1}^{k} p_i \log p_i. \tag{1}$$

There are N attributes each denoted $A_i, ,\ i = 1, 2, \ldots N$. In turn, each attribute A_i has v_i values which it can assume; v_i is finite. Shannon defined another basic concept in information theory with respect to the idea of dependence between two random variables C and A_i, which is called *mutual information* $I(C; A_i)$, and can be expressed in terms of Shannon entropies as follows

$$I(C; A_i) = H(C) - H(C|A_i), \tag{2}$$

where $H(C|A_i)$ stands for the conditional entropy of C given A_i. The mutual information is interpreted as the amount of uncertainty in C which is removed by knowing A_i.

Other entropies measures have been proposed as, for instance, Rényi entropy [3] and Tsallis entropy [4]. Rényi and Tsallis entropies contain additional parameter α which can be used to make them more or less sensitive to the shape of probability distributions.

2.2 Rényi Entropy

Rényi's entropy constitutes a measure of information of order α, having Shannon's entropy as limit case, and is defined by the following expression:

$$R_\alpha(C) = \frac{1}{1-\alpha} \log \sum_{i=1}^{k} p_i^\alpha, \quad \alpha \geq 0,\ \alpha \neq 1 \tag{3}$$

where $\sum_{i=1}^{k} p_i = 1$ and $\lim_{\alpha \to 1} R_\alpha(C) = H(C)$.

Using Rényi entropy of order $\alpha \in (0, 1)$, the mutual information can be generalized as follows:

$$I_\alpha(C; A_i) = R_\alpha(C) - R_\alpha(C|A_i) \tag{4}$$

2.3 Tsallis Entropy

Another generalized entropy, defined by Constantino Tsallis [4], is given by:

$$S_\alpha(C) = \frac{1}{\alpha - 1}\left(1 - \sum_{i=1}^{k} p_i^\alpha\right) \tag{5}$$

where $\alpha \geq 0$ and $\lim_{\alpha \to 1} S_\alpha(A) = H(A)$.

For $\alpha > 1$, Tsallis mutual information is defined as [15]:

$$I_\alpha(C; A_i) = S_\alpha(C) - S_\alpha(C|A_i) \tag{6}$$

Using Shannon entropy, events with high or low probability do not have different weights in the entropy computation. However, using Tsallis entropy, for $\alpha > 1$, events with high probability contribute more than low probabilities ones for the entropy value. Hence, the higher is the value of α, the higher is the contribution of high-probability events for the final result. Furthermore, increasing of α coefficient ($\alpha \to \infty$), Rényi entropy is increasingly determined by events with higher probabilities, and lower values of α coefficient ($\alpha \to 0$) weigh the events more equally, regardless of their probabilities.

2.4 Proposed Attribute Selection Schemes

Based on a given training set, to build a decision tree using C4.5 algorithm, first the mutual information ($I(C; A_i)$), and gain ratio ($\frac{I(C;A_i)}{H(A_i)}$) are calculated for all attributes. After, the attribute that yields the highest decrease of uncertainty about prediction of classes at that node is selected. The selection of attributes is repeated recursively until decision tree to be completely designed.

In this work, four different approaches to select a subset key attributes to identify four attacks categories are used, that fit the filter model. To do so, it was applied individually, *gain ratio* based on Rényi [3] and Tsallis [4] information measures compared with Shannon [5] information measure criteria for constructing C4.5 decision trees [6] and an ensemble approach in order to choose optimal attribute subset that can be used to find more efficient alternatives to increase the capability of IDS. The attribute selection schemes are shown in Figure 2.

3 Simulation Environment

For simulation of attributes selection schemes and to build all clustering models, WEKA toolkit(*Waikato Environment for Knowledge Analysis*) [16] has been used. In WEKA toolkit, the source code of the class J48 for generating standard C4.5 decision tree was

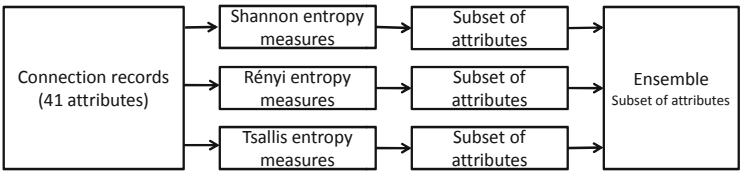


Fig. 2. Attribute selection schemes

modified by the authors, replacing the Shannon entropy by the Rényi entropy and Tsallis entropy, depending on the α value [13]. We used java programming for implementation.

In proposals evaluating of IDSs, it is usually used as benchmark the intrusion datasets available in the *Knowledge Discovery and Data Mining Competition - KDD Cup 99* [7] for both training and tests of propositions. This dataset is currently used by researchers because it still has the capability to allow researchers to compare different intrusion detection techniques on a common dataset. Each network connection (or instance) in the KDD CUP 99 data set contains 41 attributes [7]. Also, each instance is labeled either as normal or as an attack-specified type. These attacks are of 22 different types falling into four main categories:

- **DOS** - denial-of-service attacks.
- **Probing** - when an attacker scans a network to obtain information or seeks vulnerabilities.
- **Remote to Local (R2L)** - when a remote-machine user tries to get access to a local server.
- **User to Root (U2R)** - when an authorized user tries to get access as superuser (root).

Usually, a subset of network traffic is necessary to be collected in advance for designing intrusion detection systems. However, it is difficult to collect all attack information because in real world intruders constantly develop new attack codes to exploit security vulnerabilities of organizations. The collected data always encloses uncertainty when only limited information about intrusive activities is available. Accordingly, in order to simulate the problem of uncertainty existing in the KDD99 data set and to decrease computational cost without compromises the research results, a subset of the individual category of attack was randomly selected from the used intrusion datasets. As in Table 1, each category contains instances corresponding to certain attacks or normal behavior.

4 Experimental Results and Analisys

Considering the better results obtained in experiments, performed by the authors [13], for building decision trees based on Shannon [5], Rényi [3] and Tsallis [4] entropies, in terms of classification accuracy and size of the tree, we chose the best decision trees designed to be analyzed. For example, for DoS category, we selected the decision tree

Table 1. Attacks per Category

DoS	PROBING	R2L	U2R
back (1026)	ipsweep (586)	ftp-write (8)	buffer-overflow(21)
land (11)	nmap (151)	guess-passwd(53)	loadmodule (10)
neptune(10401)	portsweep (155)	imap (11)	perl (3)
pod (69)	satan (16)	multihop (11)	rootkit (7)
smurf (7669)	spy (4)	phf (5)	normal (1676)
teardrop (15)	normal (1704)	wareszclient (60)	
normal (2573)		warezmaster (20)	
		normal (1934)	

Table 2. Selected attributes by Shannon, Rényi and Tsallis information measures and ensemble approach

Attacks	Measures	Selected Attributes
DoS	Shannon	2, 5, 7, 8, 23, 34, 36, 39
	Rényi	2, 5, 7, 8, 23, 32, 35, 36, 39
	Tsallis	2, 5, 7, 8, 23, 26, 34, 39
	Ensemble approach	2, 5, 7, 8, 23, 26, 32, 34, 35, 36, 39
Probing	Shannon	1, 2, 4, 5, 6, 23, 30, 33, 37, 38, 40
	Rényi	1, 2, 5, 6, 25, 30, 32, 33, 37, 38, 40
	Tsallis	1, 2, 4, 6, 23, 30, 31, 33, 37, 38, 40
	Ensemble approach	1, 2 4, 5, 6, 23, 25, 30, 31, 32, 33, 37, 38, 40
R2L	Shannon	1, 3, 5, 6, 9, 10, 11, 17, 19, 22, 32, 33, 35
	Rényi	2, 5, 6, 10, 11, 12, 19, 33, 35, 37, 38, 39
	Tsallis	1, 3, 5, 6, 10, 11, 17, 19, 22, 37, 38
	Ensemble approach	1, 2 , 3, 5, 6, 9, 10, 11, 12, 17, 19, 22, 32, 33, 35, 37, 38, 39
U2R	Shannon	13, 16, 17, 18, 32, 33
	Rényi	13, 18, 32, 33, 36
	Tsallis	13, 16, 18, 32, 33
	Ensemble approach	13, 16, 17, 18, 32, 33, 36

designed by Rényi entropy with $\alpha = 0.5$, and the best decision tree builded using Tsallis entropy with $\alpha = 1.2$.

After to select the decision trees, individually, a subset of attributes was selected for each dataset according to individual category of attacks. Moreover, a new attributes subset were selected based on ensemble approach, using the subsets of attributes extracted by information measures of Shannon [5], Rényi [3] and Tsallis [4].

Since different types of attack have their own patterns, different categories of attacks may have different optimal subsets of attribute. So, four experiments were conducted to evaluate which attribute subset is more suitable for detecting individual category of attacks according to the respective entropy used. The experiments results are seen in Table 2.

In the experiments, the subsets of selected attributes are then used into SimpleKMeans [8] and FarthestFirst [9] algorithms for data clustering, using Weka toolkit. Cluster analysis is under strong development. Contributing areas of research include statistics, machine learning, data mining, pattern recognition, and image processing [17].

Table 3. Experimental Result

Attacks	Method	Using 41 attributes		Using Shannon		Using Rényi		Using Tsallis		Using ensemble approach	
		DAR(%)	AUC	DAR(%)	AUC	DAR(%)	AUC	DAR(%)	AUC	DAR(%)	AUC
DoS	FF	90.28	0.6497	91.16	0.6412	88.02	0.6972	91.05	0.6223	91.12	0.8074
	SKM	66.61	0.8578	68.59	0.9515	68.6	0.9511	68.59	0.9524	67.95	0.7854
Probing	FF	52.18	0.4648	54.2	0.504	57.45	0.5478	48.74	0.6313	47.25	0.4578
	SKM	61.28	0.7275	57.99	0.6981	52.68	0.6562	70.07	0.7653	68.77	0.7644
R2L	FF	98.53	0.9754	95.34	0.9675	96.43	0.9719	96.67	0.8983	96.77	0.966
	SKM	48.86	0.7355	36.16	0.667	47.29	0.7309	42.96	0.6988	36.16	0.667
U2R	FF	98.89	0.8769	95.57	0.9815	81.36	0.8724	94.41	0.9755	82.18	0.8305
	SKM	51.49	0.7575	43.91	0.7187	48.46	0.7434	43.91	0.7187	48.46	0.7434

FF = FarthestFirst, SKM = SimpleKmeans

The WEKA SimpleKMeans algorithm, its implementation of the *k-means* algorithm [8], uses Euclidean distance measure to compute distances between instances and clusters. The WEKA FarthestFirst algorithm provides the *Farthest First Traversal Algorithm* by Hochbaum and Shmoys [9], which works as a fast simple approximate clusterer modeled after simple *k-means*. FatherestFirst is a variant of *k-means* that places each cluster center in turn at the point farthest from the existing cluster centers. This point must lie within the data area.

For training and testing of *SimpleKMeans* and *FarthestFirst* algorithms, first is applied on all 41 attributes and the results of clustering are calculated. After that training and test is done with reduced subset attributes and the results of clustering is calculated. To evaluate the effectiveness of the selected attributes, the detection results using the selected attributes were compared with the results using all the 41 attributes based on the same test data.

In the mode *classes to clusters evaluation*, used in this work, Weka evaluates the clusterings in two steps: first ignores the class attribute and generates the clustering. Then during the test phase it assigns classes to the clusters, based on the majority value of the class attribute within each cluster. Then it computes the classification error, based on this assignment and also shows the corresponding confusion matrix. The criteria for evaluation was the detection accuracy (DAR) and the Area Under ROC (Receiver Operating Characteristic) Curve (AUC) [18]. The result is showed in Table 3.

In the experimental analysis on the attribute selection scheme performance, the results are significantly different if the difference is statistically significant at the 1% level. Furthermore, performance varies depending on both the clustering algorithm and the performance metric is used to evaluate models.

Compared with using all the 41 attributes (see Table 3), it can be said that among the four attribute selection techniques, the subset of key attributes selected by Tsallis entropy performs better than the other three techniques in terms of DAR and AUC when models are built using the Dos and Probing data sets and SimpleKMeans algorithm. Selecting attributes based on ensemble approach achieve better results in terms of DAR and AUC when models are built using the DoS data set and FarthestFirst algorithm. There were no significant differences in terms of DAR and AUC among Shannon, Rényi and Tsallis attribute selection techniques when models are built using the DoS data set and SimpleKMeans algorithm.

Based on Table 3, the preliminary empirical results point out that when one attribute selection scheme performed best in terms of one performance metric, this may not be true when other performance metric is used to evaluate models. For example, Rényi performed best on performance metric DAR, Tsallis performed best in terms of AUC performance metric when models are built using the Probing data set and FarthestFirst algorithm. Another obtained result in this case is that Rényi performed better than complete data set in terms of AUC.

From the Table 3, the detection results reported by the research indicate that the clustering performance in terms of both DAR and AUC for SimpleKMeans and FarthestFirst algorithms on the complete data set (with 41 attributes) significantly outperforms those on the attribute subsets selected by any attribute selection scheme for R2L and U2R attacks categories. This is consistent with Sabhnani and Serpen [19]. In this paper [19], the autors investigated the deficiencies of KDD 99 intrusion detection datasets and concluded that it is not possible to achieve a high level of detection rate on R2L and U2R attacks categories, involving content.

In particular, Shannon and Rényi entropies for feature selection do not bring any improvement in performance (DAR and AUC) for SimpleKMeans algorithm on Probing dataset (see Table 3).

Excluding the complete attribute set, we can summarize the following facts:

- Selecting attributes based on Rényi entropy performs better than the other three techniques in terms of DAR and AUC when models are built using the R2L data set and both SimpleKMeans and FarthestFirst algorithms.
- Rényi entropy technique and ensamble approach perform better than Shannon and Tsallis entropies techniques in terms of DAR and AUC for SimpleKMeans algorithm on U2R dataset.
- Shannon entropy technique is better than the other three techniques in terms of DAR and AUC for SimpleKMeans algorithm on U2R dataset.

Another obtained result, compared with Shannon entropy, is that using Tsallis and Rényi entropies, it was achieved the same size or smaller set of attributes to detect attacks for all attacks categories and using Rényi entropy, it was achieved the same size or smaller set of attributes for Probing, R2L and U2R attacks categories.

5 Conclusion

In this paper, it is presented an evaluation of Rényi and Tsallis entropy compared with Shannon entropy and their applications to intrusion detection system. Additionally, we studied an ensemble approach that combines the attributes selected by Rényi, Tsallis and Shannon information measures. The experimental results shows that in general, selecting attributes based on Rényi and Tsallis entropies can to achieve better results, compared with Shannon entropy and ensemble approach considering an individual data set and the clustering algorithm employed, since one works better than other. Moreover, attribution selection approach based on Rényi or Tsallis entropies reduce the number of attributes and processing time. For future research, it will be used more detailed attributes from real network traffic that supposedly are able to better characterize packet contents as well as header data.

Acknowledgment. The authors would like to thank Brazilian Coordination for Improvement of Higher Education Personal (CAPES), National Council for Scientific and Technological Development (CNPq) and State Research Supporting Foundation of Maranhão (FAPEMA).

References

1. Crosbie, M., Spafford, E.: Defending a computer system using autonomous agents. Department of Computer Sciences, Purdue University, CSD-TR-95-022; Coast TR 95-02 (1995)
2. Estévez, P.A., Tesmer, M., Perez, C.A., Zurada, J.M.: Normalized mutual information feature selection. IEEE Tran. on Neural Networks 20(2), 189–201 (2009)
3. Rényi, A.: On measures of entropy and information. In: Proc. the 4th Berkeley Symposium on Math. Statistics and Prob., pp. 547–561. Univ. of California Press, Berkeley (1960)
4. Tsallis, C.: Possible generalization of boltzmann-gibbs statistics. Journal of Statistical Physics 52(1-2), 479–487 (1988)
5. Shannon, C.E.: A mathematical theory of communication. Bell Systems Technical Journal 27, 623–656 (1948)
6. Quinlan, J.R.: C4.5 Programs for Machine Learning. Morgan Kaufmann Publishers, San Diego (1993)
7. Kdd cup 99 intrusion detection data set (retrieved March 01, 2010), http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
8. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Le Cam, L.M., Neyman, J. (eds.) Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, vol. 1, pp. 281–297 (1967)
9. Hochbaum, D.S., Shmoys, D.B.: A best possible heuristic for the k-center problem. Mathematics of Operations Research 10(2), 180–184 (1985)
10. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco (2006)
11. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Tran. on Knowledge and Data Engineering 17, 491–502 (2005)
12. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
13. Lima, C.F.L., de Assis, F.M., Souza, C.P.: Decision tree based on shannon, renyi and tsallis entropies for intrusion tolerant systems. In: Fifth International Conference on Internet Monitoring and Protection, pp. 117–122 (May 2010)
14. Tsallis, C.: Nonextensive statistics: Theoretical, experimental and computational evidences and connections. Brazilian Journal of Physics 29, 1–35 (1999)
15. Furuichi, S.: Information theoretical properties of tsallis entropies. Journal of Mathematical Physics 47(2) (2006), http://link.aip.org/link/?JMP/47/023302/1
16. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 2nd edn. Morgan Kaufmann Publishers, California (2005)
17. Kaufman, L., Rousseeuw, P.: Finding Groups in Data An Introduction to Cluster Analysis. Wiley Interscience, New York (1990)
18. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27(8), 861–874 (2006), http://dx.doi.org/10.1016/j.patrec.2005.10.010
19. Sabhnani, M., Serpen, G.: Why machine learning algorithms fail in misuse detection on kdd intrusion detection data set. Intell. Data Anal. 8, 403–415 (2004)

Supervised Growing Neural Gas

Klaifer Garcia and Carlos Henrique Quartucci Forster

Instituto Tecnológico de Aeronáutica, São Jose dos Campos, SP, Brasil
Klaifer.garcia@gmail.com, forster@ita.br

Abstract. We present a new approach to supervised vector quantization inspired on growing neural gas network. An advantage of the new method is that it reduces the need for prior knowledge about the problem under study because it is able to determine at runtime the size of the codebook. Another advantage is that the training is less dependent on the initial state of the codebook vectors in contrast to methods like Learning Vector Quantization. Finally, it is shown that for some real datasets the classification performance is superior to other methods of supervised vector quantization.

Keywords: Supervised Learning, Learning Vector Quantization, Growing Neural Gas, Neural Networks.

1 Introduction

Vector quantization is a widely used technique in both supervised and unsupervised cases. The objective is to build a set of representative vectors, called codebook, to represent the input dataset with minimum loss under some metric. Thus, two important factors in building this codebook are the number of representative vectors to be used and the values of its attributes.

A widely used supervised method is Learning Vector Quantization (LVQ) because it is simple to implement and has good generalization even for multiclass problems. Among the related publications of this method we can highlight alternative measures of dissimilarity [1], and problem solving like divergent reference vectors [2].

It is known that the quality of the solution obtained with the LVQ is related to the initial state of the codebook and for this reason it is common to use a pre-processing step to start the codebook vectors with an unsupervised method like SOM [3]. The problem with this segmented approach is that it is only possible to know whether the result of unsupervised step is satisfactory after finishing the supervised step, which makes the application of this technique laborious.

Besides, the supervised and the unsupervised methods differ in purpose. In unsupervised methods, in general, the objective is to obtain codebook vectors distributed as close as possible to the training set, minimizing the dissimilarity of samples to its

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 502–507, 2012.

representative vector. For this reason the execution of methods such SOM networks produce codebook vectors approximately uniform distributed among the instances.

In contrast, the goal on supervised models is the reduction of misclassification rate. Therefore, it is not necessary to represent well all the instances, but the decision frontiers between the classes. To illustrate this, in the Figure 1 is presented the results of the execution of a SOM followed by its adaptation with LVQ.

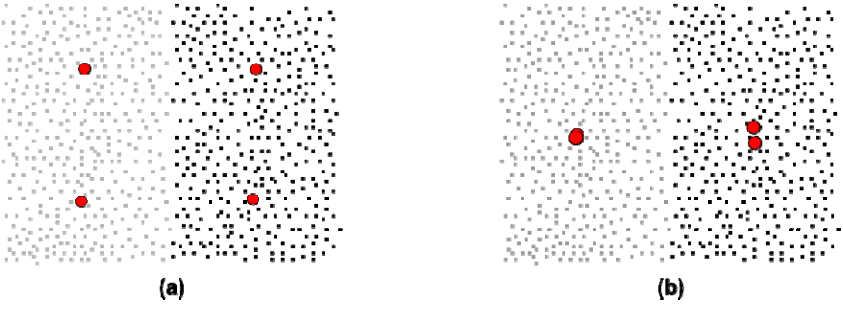


Fig. 1. Result of processing with 4 codebook vectors. The points are instances and circles are codebook vectors. On the left (a): the result of processing with SOM and, on the right (b): the adjustment with LVQ3.

The integration of these two phases was proposed in works such as [4], adding to LVQ a better ability to adapt to the distribution of examples in the attributes space to reduce the influence of the codebook vectors initialization on the result. In addition, in this work, we seek to define the codebook size at runtime.

2 The Proposed Method

The proposed algorithm is inspired by the unsupervised neural network "Growing Neural Gas" [5]. The GNG network is a neural network which uses a competitive learning method and is capable of creating or deleting prototype vectors to adjust the codebook to the distribution of the instances on attribute space.

We maintain this ability to adjust the codebook to the problem, with modifications to allow the creation and adjustment of codebook vectors guided by a supervised procedure, making the boundaries of the corresponding Voronoi regions to approximate the borders of the class distribution. The algorithm of the proposed method is described in Figure 2:

Analogously to what happens in SOM and GNG networks, the maintenance of connections (2:a-d) is made with a Hebbian-like method and is responsible for the topological organization of the network. These connections are independent of the unit class and are important to the adjustment of the network.

- Consider a set R of instances available for training
- Consider a set G of units. Each unit $s \in G$ has a codebook vector w_s associated.
- Consider a set N of connection (or edges) among pair units. N_s denote the set of units connected with s.

1. Initialize G to contain one unit to each class with random codebook vectors, all interconnected.
2. For each new randomly selected instance $x \in R$:
   a) Search the codebook vectors of the set G to find:
   b) s_0 : The nearest unit to "x"
   c) s_1 : The second-nearest unit to "x"
   d) s_2 : The nearest unit to "x" with the same class of "x" (if it exists)
   e) If s_0 and s_1 are connected, set the connection age to 0. Otherwise connect them and set the connection age to 0.
   f) Increment the connection age to all neighbours of s_0 .
   g) Remove connections with age larger than a limit. Remove disconnected units.
   h) If s_0 isn't the same class of "x", increment the s_0 error to the class of "x".
   $\Delta error = \left\| x - w_{s0} \right\|$
   i) Update s_2 (if it exists) and its neighbours $s_n \in N_{s2}$.
   To s_2 : $\Delta w_{s2} = \alpha (x - w_{s2})$
   To s_n with the same class of "x": $\Delta w_{sn} = \beta (x - w_{sn})$
   To s_n with class different of "x": $\Delta w_{sn} = -\beta (x - w_{sn})$
   j) Repeat step 2 until all examples in R are processed
3. If some criterion is met, insert a new unit on G:
   a) Search G to find the unit s_m and the class "c" with larger error.
   b) Duplicate this unit and set the class to "c"
   c) Connect this new unit to s_m .
4. Decrease all error variables by multiplying them with a constant.
5. If a stopping criterion is not yet fulfilled go to step 2.

Fig. 2. Supervised Growing Neural Gas Algorithm

It is important to note that the error in each unit is stored separately for each class (2:e). The adjustment (2:f) is winner-take-most and not made to the most similar unit but to the most similar in the same class of the selected instance. It is expected that, as the network converges, more frequently the winning unit belongs to the same class of the example shown.

Concerning the units that are not near the decision boundary, and therefore have no neighbors of different classes, the algorithm behaves in the same way as the original GNG network, promoting a distribution of codebook vectors approaching the Delaunay triangulation.

In the adjustment step (2:f), units surrounding the winner in the same class of the instance 'x' are pulled towards the presented instance and those from different classes are pushed away from it in a similar fashion to what occurs in LVQ. The algorithm does not restrict the exclusion of all units of a class, although, in the general case, it is unlikely. For this reason the condition "if it exists" was included. The adjustment factor of the neighborhood (β) must be small compared to that of the winner (α) to avoid instability in the network. Despite being small, this neighborhood adjustment is an important procedure for allowing cooperation between the units.

The criterion for inserting a new unit (step 3) may be the stabilization of the error decay or a fixed number of training epochs. Unlike the original GNG algorithm, the

insertion of a new unit is not made through the interpolation of two units. This is because according to Cover [6], close instances (with respect to some appropriate metric) probably belong to the same class. Thus, the interpolation produces a codebook vector in a region on the attribute space different from the region of its class, and therefore does not represent any advantage for network convergence.

Step 4 corresponds to the decrement of the error of all units. This operation is performed once per epoch. In the original algorithm, the decay happens after the presentation of each instance, but this decrement made by epoch is less dependent on the problem, and is therefore more general.

3 Results

3.1 Artificial Dataset

Tests were performed with a synthetic dataset to evaluate the ability of the algorithm to adjust the codebook to the problem. The dataset used was two dimensional with 200 samples divided into three classes, having two of these classes greater intersection.

The algorithm was configured to insert a new unit every 50 epochs and maintain connections with their neighbors until the maximum age of 30. The learning rate used for the winner was 0.05 and for its neighbors 0.006. The training was performed with all the dataset until no more errors occur.

Figure 3a represents the results of these tests. It is possible to see a higher concentration of codebook vectors in the more complex regions of the problem. For comparison, Figure 3b represents the result of processing with LVQ 3 with the same number of codebook vectors, for which the resulting error did not reach zero.

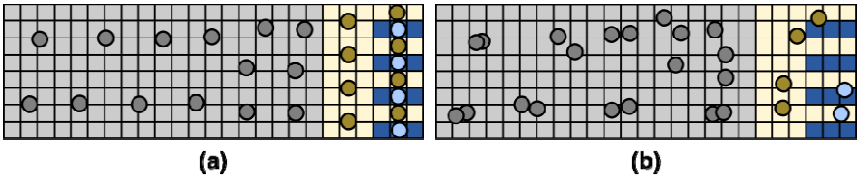


Fig. 3. Results of training with synthetic dataset. The squares are instances and circles are codebook vectors.(a) Results with SGNG. (b) Results with LVQ.

3.2 Real World Dataset

Tests were performed with datasets from the UCI [7] as shown in Table 1, the value in parentheses in the CAR dataset represents the number of attributes resulting from the conversion from nominal to boolean. The attributes were normalized to values between 0 and 1. All experiments were repeated 100 times using 75% of the data for training and 25% for testing.

Table 2 presents the results of the SGNG, LVQ and GCS [8]. Three implementations of LVQ algorithms were considered, following the model described in LVQ PAK [9]. In all trials, these algorithms were configured to produce the same number of codebook vectors obtained with SGNG in order make comparison among the methods easier.

The algorithm Growing Cell Structures (GCS) is an RBF network, which uses a GNG network in its hidden layer to determine the centers of Gaussian activation functions and their respective standard deviations. Also, to facilitate the comparison the number of units in the hidden layer was limited to the same number of codebook vectors obtained with SGNG. This limitation causes a negative effect on the results with GCS for some of the datasets, achieving similar results only with a larger number of units in its hidden layer.

For all of the algorithms the parameters were chosen using the weka [10] CVParameterSelection metaclassifier, and experiments to refine these settings. Values in parentheses represent the standard deviation observed in the experiments.

Table 1. Datasets descriptions

	Glass	*Sonar*	*Car*	*Wine*	*Iris*
Examples	214	208	1728	178	150
Attributes	9	60	6 (21)	13	4
Classes	5	2	4	3	3

Table 2. Codebook size and correct classification rate (in %) of test set with diferents databases

	Glass	*Sonar*	*Car*	*Wine*	*Iris*
Codebook	40	21	33	31	8
LVQ1	61.93(6.31)	80.36(5.62)	76.73(1.85)	95.78(2.65)	87.30(19.79)
LVQ2.1	60.11(6.59)	81.07(5.46)	88.49(2.00)	95.35(3.14)	91.08(10.81)
LVQ3	62.52(6.20)	83.41(5.37)	90.76(1.78)	**96.52(2.47)**	90.82(10.86)
GCS	55.47(14.75)	71.42(9.73)	64.85(21.31)	95.56(4.50)	93.24(6.31)
SGNG	**67.99(5.83)**	**85.96(4.34)**	**90.84(1.69)**	95.92(3.17)	**95.61(3.26)**

4 Conclusion

We presented an algorithm for supervised vector quantization inspired on GNG network capable of determining in runtime the number of codebook vectors and its distribution between classes to increase the accuracy in the classification. Tests were conducted to demonstrate the performance of the method using well known datasets and the results show higher classification accuracy compared to LVQ and GCS in most of the evaluation cases.

References

1. Mwebaze, E., Schneider, P., Schleif, F.-M., Haase, S., Villmann, T., Biehl, M.: Divergence based learning vector quantization. In: Proceedings of the 18th European Symposium on Artificial Neural Networks (ESANN), pp. 247–252 (2010)
2. Sato, A.: A formulation of learning vector quantization using a new misclassification measure. In: Proceedings of Fourteenth International Conference Pattern Recognition, vol. 1, pp. 322–325 (1998)
3. Kohonen, T.: The self-organizing map. Proceedings of the IEEE 78(9), 1464–1480 (1990)
4. Hammer, B., Strickert, M., Villmann, T.: Supervised Neural Gas with General Similarity Measure. Neural Processing Letters 21, 21–44 (2005)
5. Fritzke, B.: A growing neural gas network learns topologies. In: Advances in Neural Information Processing Systems, vol. 7, pp. 625–632 (1995)
6. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13, 21–27 (1967)
7. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2010), `http://archive.ics.uci.edu/ml`
8. Fritzke, B.: Supervised Learning with Growing Cell Structures. In: Advances in Neural Information Processing Systems, vol. 6, pp. 255–262 (1994)
9. Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J., Torkkola, K.: LVQ PAK: The Learning Vector Quantization program package. Report A30, Helsinki University of Technology, Laboratory of Computer and Information Science (1996)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)

A General Approach for Adaptive Kernels in Semi-Supervised Clustering

Sílvia Grasiella Moreira Almeida, Frederico Gualberto F. Coelho, Frederico Gadelha Guimarães, and Antonio Pádua Braga

Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
silvia.almeida@ifmg.edu.br, fredgfc@gmail.com,
{apbraga,fredericoguimaraes}@ufmg.br

Abstract. Semi-supervised clustering aims at accomplishing the clustering task by considering also labels or constraints provided by an external agent. Usually, the agent would provide the output label for a reduced number of patterns or, in the case of lack of posterior information about labels, some pairwise constraints indicating whether or not two patterns should be joined in the same cluster. Constraints may be inferred from some ad-hoc information from sampling, such as their geographical location, which are not directly considered as an input atribute. The objective is to accomplish the clustering task by considering also the pairwise constraints. In this paper we extend the previous work of Yan et al. [10] by obtaining derivative expressions for sigmoidal and polinomial kernels in order to accomplish kernel-clustering semi-supervised tasks. The resulting kernel-clustering task is optimized in relation to kernel parameters which do not need to be provided in advance like in most kernel-clustering tasks. Instead, kernel parameters are obtained as the outcome of the optimization problem.

1 Introduction

Many problems in machine learning involve extracting knowledge and structure from input data, without further information about target outputs for the construction of inductive classification models. Discovering the underlying structure in the data is an important challenge in knowledge discovery and in data analysis. In this context, when we have an input dataset $\mathcal{X} = \{\mathbf{x}_i\}$ but we do not know *a priori* the classification of each sample in $\mathcal{X}$, clustering techniques can be employed to uncover strucutre in the dataset. Structuralization would be helpful if the consistency principle between structure and output classes is valid. In order to estimate the modes of the generator functions, K-means and Fuzzy C-means are popular clustering techniques adopted, although hierarchical and kernel-based methods can also be used [9].

In Active Learning [5], samples for labelling are selected and presented to an external agent (system user or expert) so that learning is accomplished as a result of information extracted from data and of the assignments provided by the external agent. Since very few samples are labeled, structuralization plays

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 508–515, 2012.

a major role in such a context, since lack of information may be compensated by structural information extracted from the dataset. Learning in the context of large datasets and scarce labelling is the challenge of Semi-Supervised Learning (SSL) [7] and of Semi-Supervised Clustering (SSC) [8].

External information can be provided by labelling some selected samples from $\mathcal{X}$ as well as by imposing pairwise constraints, which indicate whether or not two samples shoud be grouped in the same cluster. Assuming that the consistency principle is valid, pairwise constraints for clusters can be extended for labels. Although labels are assigned by the external agent, pairwise constraints can be induced from structural information extracted from $\mathcal{X}$.

In the context of SSL the supervised tasks are related to label assignment and pairwise constraints, while the unsupervised tasks are related to structuralization. More specifically, the *must-link* (ML) constraints define those samples that must belong to the same cluster, whereas the *cannot-link* (CL) constraints define those samples that must belong to different clusters. The remaining data is related to the unsupervised learning problem. Clustering is then accomplished as a combination of the contributions resulted from pairwise similarity and constraints.

Although clustering is usually accomplished in the input space, the task can also be carried out after nonlinear kernel mapping into a feature space [6]. This is particularly interesting because kernels already embody pairwise similarity measures and uncover pattern and cluster relations. In a previous work [10], the kernel clustering task for RBF kernels was described by an objective function composed by similarity and pairwise constraint terms. Since the original work was restricted to RBF kernels, in this paper, we generalize the idea by developing gradient terms for polinomial and sigmoidal kernels, which can now be applied to the SSC task.

In this paper, we present a general approach for semi-supervised clustering based on kernel functions. The cost function is the average distance between data, with the distance measure given by a kernel function applied to the feature space of the problem. We present a method for adapting the parameter of the kernel function, showing the methodology for three different kernel functions: (i) gaussian, (ii) polinomial and (iii) sigmoidal. Finally we compare the results in some benchmark problems using these three different kernel functions.

2 Semi-supervised Kernel-Based Clustering

Clustering in feature space can be described by the minimization of the objective function J presented in Equation 1 that follows.

$$J = \sum_{c=1}^{k} \sum_{\mathbf{x}_i \in \pi_c} \delta \left(\phi \left(\mathbf{x}_i \right) - \mathbf{m}_c^{\phi} \right) \tag{1}$$

where k is the number of clusters, $\delta(\cdot)$ is a distance metric function, $\phi(\mathbf{x}_i)$ is the feature space representation of input pattern $\mathbf{x}_i$ resulted from the mapping $\phi(\cdot)$,

and

$$\mathbf{m}_c^{\phi} = \frac{1}{|\pi_c|} \sum_{\mathbf{x}_i \in \pi_c} \phi(\mathbf{x}_i)$$

is the centroid of cluster π_c in feature space.

It is assumed in this work that ML and CL pairwise constraints are given by an external agent, so the constraint violation objective functions should be jointly minimized together with Equation 1 in the definition of the general optimization problem. The algorithm must find a partition of the data $\mathcal{X}$ into k clusters such that the mean distance to centroids and the pairwise constraint violations are minimized. A linear combination of the similarity objective function and constraint violations functions for ML and CL was adopted by Yan et al. [10] as an approach for solving the resulting SSC problem, as shown in Equation 2.

$$J(\{\pi_c\}, k) = \sum_{c=1}^{k} \sum_{\mathbf{x}_i \in \pi_c} \delta\left(\phi(\mathbf{x}_i) - \mathbf{m}_c^{\phi}\right) - \sum_{\mathbf{x}_i, \mathbf{x}_j \in ML, l_i = l_j} w_{ij} - \sum_{\mathbf{x}_i, \mathbf{x}_j \in CL, l_i = l_j} \bar{w}_{ij} \tag{2}$$

where w_{ij} and $\bar{w}_{ij}$ represent rewards for satisfying the ML and CL constraints respectively.

The general optimization problem defined in Equation 2, however, requires that kernel parameters characterized by the mapping function $\phi(\mathbf{x}_i)$ are set in advance. In [10], some alternatives for setting kernel parameters are given, namely, kernel alignment [3] and Fisher discriminant to estimate the optimal radius of gaussian kernels. Given the influence of the kernel parameter on the performance of the clustering method, the authors also present an adaptive method for semi-supervised kernel-based clustering. The corresponding objective function of the *Adaptive Semi-Supervised Kernel K-Means* described in [10] is presented in Equation 3.

$$\begin{aligned} J = \sum_{c=1}^{k} \sum_{\mathbf{x}_i, \mathbf{x}_j \in \pi_c} \frac{1 - K(\mathbf{x}_i, \mathbf{x}_j)}{|\pi_c|} + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in ML, l_i \neq l_j} 2w_{ij}(1 - K(\mathbf{x}_i, \mathbf{x}_j)) \\ + \sum_{\mathbf{x}_i, \mathbf{x}_j \in CL, l_i = l_j} 2\bar{w}_{ij} \left(K(\mathbf{x}_i, \mathbf{x}_j) - K(\mathbf{x}', \mathbf{x}'')\right) - \sum_{\mathbf{x}_i \in \mathcal{X}} 2(1 - K(\mathbf{x}_i, \mathbf{x}_r)) - C \end{aligned} \tag{3}$$

where $\mathbf{x}_r$ is one point randomly selected from X, **x'** e **x"** are the farthest points in feature space and C is a constant that avoid the degenerate case in which the objective function is trivially minimized.

The optimization of the kernel parameter is performed by using the Gradient method. The initial clusters are initialized from the ML and CL constraints, applying the *Farthest First Traversal* algorithm.

3 A General Approach for Adaptive Kernels

We employ the same objective function proposed in [10], however, we develop the derivatives for some kernel functions and use this information together with

the secant method to obtain the optimal value for the kernel parameter σ. At the optimal value, we have $\frac{dJ(\sigma^*)}{d\sigma} = 0$ or, in other words, the optimal value of the kernel parameter is also the root of the corresponding nonlinear equation. In numerical analysis, there are many methods for computing the root of nonlinear equations, among them the secant method is a simple and efficient option, presenting superlinear convergence [1]. For a given kernel function, the analytical expressions $dJ/d\sigma$ for the kernel parameters are needed in order to be used in association with the secant method and then to obtain the optimal value with precision, fast convergence and without need of additional parameters. The secant method is defined according to Equation 4.

$$\sigma^n = \sigma^{n-1} - g(\sigma^{n-1})\frac{\sigma^{n-1} - \sigma^{n-2}}{g(\sigma^{n-1}) - g(\sigma^{n-2})} \tag{4}$$

where $g(\cdot) = dJ/d\sigma$.

Therefore, in order to solve the nonlinear equation $\frac{dJ(\sigma^*)}{d\sigma} = 0$ with the secant method described in Equation 4, the derivative of the objective function in relation to each one of the kernel parameters needs to be obtained. We derive next the objective function presented in Equation 3 for each one of the most common kernel functions: gaussian, polinomial and sigmoidal [4].

Considering that the polinomial kernel is given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T.\mathbf{x}_j + 1)^d \tag{5}$$

and that the sigmoidal kernel is given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1 + \exp\left(-a||(\mathbf{x}_i - \mathbf{x}_j)||\right)} \tag{6}$$

the parameters d and a can be adapted as with σ for the gaussian kernel.

For the gaussian kernel, the derivative of the objective function with respect to σ is presented below:

$$\begin{aligned}
\frac{\partial J(\sigma)}{\partial \sigma} = &- \sum_{c=1}^{k} \sum_{x_i, x_j \in \pi_c} \frac{1}{|\pi_c|} \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^3} \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right) \\
&- \sum_{(x_i, x_j) \in ML, l_i \neq l_j} 2w_{ij} \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^3} \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right) \\
&+ \sum_{(x_i, x_j) \in CL, l_i = l_j} 2\bar{w}_{ij} \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^3} \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right) \\
&- \frac{||\mathbf{x}' - \mathbf{x}''||^2}{\sigma^3} \exp\left(\frac{-||\mathbf{x}' - \mathbf{x}''||^2}{2\sigma^2}\right) \\
&+ \sum_{x_i \in \mathcal{X}} 2\frac{||\mathbf{x}_i - \mathbf{x}_r||^2}{\sigma^3} \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_r||^2}{2\sigma^2}\right)
\end{aligned} \tag{7}$$

For the polinomial kernel, we have:

$$\begin{aligned}\frac{\partial J(d)}{\partial d} = &\sum_{c=1}^{k} \sum_{(x_i,x_j \in \pi_c)} \frac{1}{|\pi_c|}(\mathbf{x}_i^T \mathbf{x}_j + 1)^d \ln(\mathbf{x}_i^T \mathbf{x}_j + 1) \\ &- \sum_{(x_i,x_j)\in ML, l_i \neq l_j} 2w_{ij}(\mathbf{x}_i^T \mathbf{x}_j + 1)^d \ln(\mathbf{x}_i^T \mathbf{x}_j + 1) \\ &+ \sum_{(x_i,x_j)\in CL, l_i = l_j} 2\bar{w}_{ij}(\mathbf{x}_i^T \mathbf{x}_j + 1)^d \ln(\mathbf{x}_i^T \mathbf{x}_j + 1) \\ &- (\mathbf{x'}^T \mathbf{x''} + 1)^d \ln(\mathbf{x'}^T \mathbf{x''} + 1) + \sum_{x_i \in \mathcal{X}} 2(\mathbf{x}_i^T \mathbf{x}_r + 1)^d \ln(\mathbf{x}_i^T \mathbf{x}_r + 1)\end{aligned} \tag{8}$$

Finally, for the Sigmoidal kernel:

$$\begin{aligned}\frac{\partial J(a)}{\partial a} = &\sum_{c=1}^{k} \sum_{(x_i,x_j \in \pi_c)} \frac{-||\mathbf{x}_i - \mathbf{x}_j|| \exp(-a||\mathbf{x}_i - \mathbf{x}_j||)}{|\pi_c|(1 + \exp(-a||\mathbf{x}_i - \mathbf{x}_j||))^2} \\ &- \sum_{(x_i,x_j)\in ML, l_i \neq l_j} 2w_{ij} \frac{||\mathbf{x}_i - \mathbf{x}_j|| \exp(-a||\mathbf{x}_i - \mathbf{x}_j||)}{(1 + \exp(-a||\mathbf{x}_i - \mathbf{x}_j||))^2} \\ &+ \sum_{(x_i,x_j)\in CL, l_i = l_j} 2\bar{w}_{ij} \frac{||\mathbf{x}_i - \mathbf{x}_j|| \exp(-a||\mathbf{x}_i - \mathbf{x}_j||)}{(1 + \exp(-a||\mathbf{x}_i - \mathbf{x}_j||))^2} \\ &- \frac{||\mathbf{x'} - \mathbf{x''}|| \exp(-a||\mathbf{x'} - \mathbf{x''}||)}{(1 + \exp(-a||\mathbf{x'} - \mathbf{x''}||))^2} + \sum_{x_i \in \mathcal{X}} 2\frac{||\mathbf{x}_i - \mathbf{x}_r|| \exp(-a||\mathbf{x}_i - \mathbf{x}_r||)}{(1 + \exp(-a||\mathbf{x}_i - \mathbf{x}_r||))^2}\end{aligned} \tag{9}$$

4 Experiments

4.1 Datasets

Our experiments are based on in three real and one simulated dataset. The real datasets are available in [2]. The first dataset is Pen-Based Recognition of Handwritten Digits Data Set, or Digits. In this dataset, we have 317 instances and 16 features and we have chosen three classes for clustering: 3, 8 and 9. The second dataset describes the diagnoses of cardiac Single Proton Emission Computed Tomography (SPECT) images. We have 2 classes for this dataset: the patient is classified as normal or abnormal. The data has 267 instances described by 45 features. The third dataset is about segmentation of images. For this dataset, we have 210 instances described by 19 features and divided in 7 classes. The artificial dataset are composed by 5 classes, 500 instances and three features.

The use of constraints affects directly the performance of the results. In this paper, we use probability to define constraints for each dataset. For each problem,

we establish a probability that pairwise must-link and cannot-link belongs or not, respectively, at the same cluster.

For the first problem, Digits, we defined three different probabilities, 45%,70% and 80%. For the second problem, SPECT, we defined 50%, 70%, 80% and 90% of probabilities for the pairwise. In Segmentation problem we defined 50%, 70%, 80% and 90% and, finally, 70%, 80%, 90% and 95% for the artificial dataset.

We use an external cluster validity measure that estimates the quality of the clustering results with respect to the underlying class of the data, the Rand Statistic index [10] to evaluate the clustering results.

5 Results and Discussion

To evaluate the performance of our proposed algorithm for three different kernels, we perform comparisons between them. For each kernel, we defined one initial parameter. In Gaussian Kernel, we started with sigma equal to 100. In Polinomial kernel, we set the initial value of the parameter d as equal to 0.1 and for the

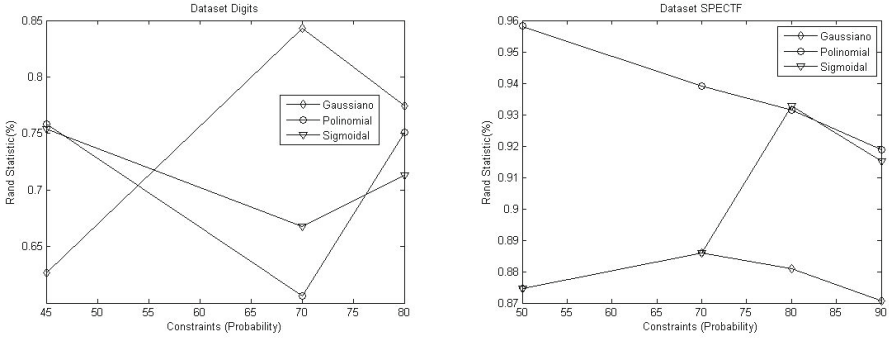


Fig. 1. Clustering for database Digits and SPECTF

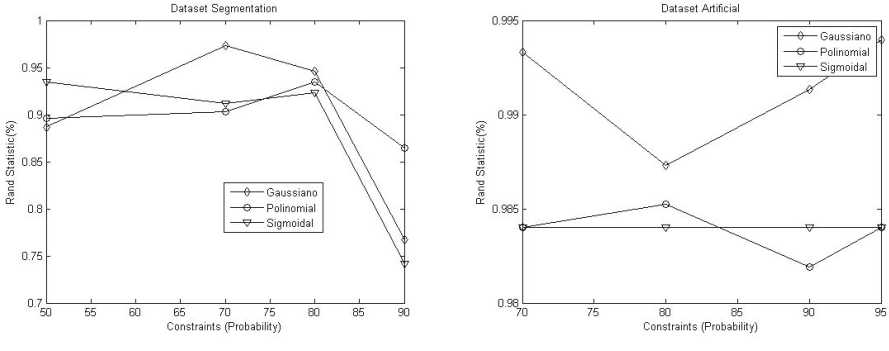


Fig. 2. Clustering for database Segmentation and Artificial

Sigmoidal kernel, we set the initial value of the parameter a as equal to 0.1. We found the initial clusters in a set of constraints.

For dataset Digits, Segmentation and Artificial, the Gaussian kernel provides better results than the Sigmoidal and Polinomial kernels. For the dataset SPECTF and Artificial the Sigmoidal kernel does not show significant sensitivity to the number of constraints. For the database Segmentation, there is a decline in performance for 90% of probability.

In many studies, experiments are performed based on the amount of pairwise constraints. However, we prefer to investigate results based on the pairwise constraints probabilities in our experiments. Thus, there are instances when the amount of pairwise constraints data can be a small set. The results showed in this paper indicate that kernels affects directly the clustering of the data. Since the kernel is responsible for mapping data from the input space to a feature space, a variation of performance is expected. The choice of Kernel parameters by means of an optimization procedure is more effective than the random choice of parameters.

6 Conclusion

We proposed a general approach for adaptive semi-supervised Kernel-KMeans algorithm. Our approach integrates constraints with three kernels functions. We established these constraints in terms of its probability of belonging or not to the same cluster. The proposed algorithm calculates automatically the optimal parameters by using the secant method for every kernel. It uses these parameters to find new clusters based on the distance in feature space. In our future work we intend to explore tecnhiques that add strutuctural information to find and integrate constraints to the problem.

References

1. Barnes, J.G.P.: An Algorithm for Solving Non-Linear Equations Based on the Secant Method. The Computer Journal 8, 66–72 (1965)
2. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Cristianini, N., Shawe-Taylor, J., Elisseeff, A.: On Kernel-Target Alignment. In: Neural Information Processing Systems, NIPS (2001)
4. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature Extraction, Foundations and Applications. STUDFUZZ. Physica-Verlag, Springer (2006)
5. Mazzoni, D., Wagstaff, K.L., Burl, M.C.: Active Learning with Irrelevant Examples. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 695–702. Springer, Heidelberg (2006)
6. Kulis, B., Basu, S., Dhillon, I., Mooney, R.J.: Semi-supervised graph clustering: a kernel approach. In: International Conference on Machine Learning (2005)
7. Li, T., Ogihara, M.: Semisupervised learning from different information sources. Knowledge and Information Systems 7(3), 289–309 (2005)

8. Ruiz, C., Vallejo, C.G., Spiliopoulou, M., Menasalvas, E.: Automated Constraint Selection for Semi-supervised Clustering Algorithm. In: Meseguer, P., Mandow, L., Gasca, R.M. (eds.) CAEPIA 2009. LNCS, vol. 5988, pp. 151–160. Springer, Heidelberg (2010)
9. Xu, R., Wunsch, D.: Clustering. IEEE Press Series on Computational Intelligence. Wiley-IEEE Press (2008)
10. Yan, B., Domeniconi, C.: An Adaptive Kernel Method for Semi-supervised Clustering. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 521–532. Springer, Heidelberg (2006)

Ensemble Methods for Prediction of Parkinson Disease

Sami M. Halawani and Amir Ahmad

Faculty of Computing and Information Technology,
King Abdulaziz University, Rabigh, Saudi Arabia

Abstract. Parkinson disease is a degenerative disorder of the central nervous system. In the present paper, we study the effectiveness of regression tree ensembles to predict the presence and severity of symptoms from speech datasets. This is a regression problem. Regression via classification (RvC) is a method in which a regression problem is converted into a classification problem. A discretization process is used to convert continuous target value to classes. The discretized data can be used with classifiers as a classification problem. In this paper, we also study a recently developed RvC ensemble method for the prediction of Parkinson disease. Experimental results suggest that the RvC ensembles perform better than a single regression tree. Experiments also suggest that regression tree ensembles created using bagging procedure can be a useful tool for predicting Parkinson disease. The RvC ensembles and regression tree ensembles performed similarly on the dataset.

Keywords: Parkinson disease, Ensembles, Decision trees, Regression trees.

1 Introduction

Parkinson disease (PD) is a degenerative disorder of the central nervous system [16,20]. Parkinson disease (PD) is the second most common neurodegenerative disorder after Alzheimer's [7]. The most obvious symptoms of this disease are movement-related, including shaking, rigidity, slowness of movement and difficulty with walking and gait [16]. As there is no available cure, early diagnosis is important to improve the patient's quality of life and to prolong it [21]. Unified Parkinson's Disease Rating Scale (UPDRS) is used to measure the presence and severity of symptoms. Motor UPDRS is used to measure motor symptoms (tremor, rigidity, slowness of movement, and postural instability). As physical examinations of patients are time consuming, Tsanas et al. [24] suggested a remote speech test for Parkinson patients. These speech datasets were used for predicting UPDRS. The proposed technique has potential to be very useful for monitoring Parkinson patients. The accurate prediction of UPDRS by using speech datasets is one of the key factors for this technique.

In machine learning and data mining fields, supervised learning plays an important role [3,18]. In a regression problem, the target values are continuous, whereas in the classification problem we have discrete set of classes. The other difference is that regression values have a natural ordering, whereas for the classification the class values are unordered [3,18]. The value of UPDRS is in real value, hence this prediction comes under regression.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 516–521, 2012.

Regression models are not easily understood by domain experts, and thus provide little help in understanding the problem, whereas classification models are more comprehensible, but not very useful, when the target values are continuous. There are some learning schemes, like naive Bayes, which are very successful as classification techniques, however, they are difficult to use as regression schemes. Decision trees [5,19], neural networks [3,18], naive Bayes [3,18], support vector machines [26,6] etc. are quite popular for classification problems, whereas regression trees [5], neural networks [3,18], support vector machines [26,6] etc. are used for regression problems.

Discretization [10] is a process that divides continuous numeric values into a set of intervals (bins) that can be considered as categorical values. Equal-width intervals, equal-frequency intervals and entropy minimization discretization [11] etc. are popular discretization methods.

Researchers [22,15,23] suggest that the discretizaion process can be used to convert continuous target values into a discrete set of classes and then classification models are used to solve the classification problems. In other words, in a RvC problem, a regression problem is solved by converting into a classification problem. This method employs any classifier on a copy of the data that has the target attribute discretized. The whole process of RvC comprises two important stages:

1. The discretization of the numeric target variable in order to learn a classification model. There are different discretzation methods eg. equal-width, equal-frequency etc [10].
2. The reverse process of transforming the class output of the classification model into a numeric prediction. We may use the mean value of the target variable for each interval as the final prediction.

Ensembles are a combination of multiple base models [9,14,25]; the final classification or regression results depends on the combined outputs of individual models. Ensembles have shown to produce better results than single models, provided the models are *accurate* and *diverse* [14].

Neural networks and decision tree ensembles are quite popular. Bagging [4] and Boosting methods [12] are general and can be used with any classifiers. Bagging and Boosting have been used with regression trees for the regression problem.

In the paper, we study whether ensembles of regression trees can have better results as compare to single regression tree for the prediction of Parkinson disease and whether the RvC ensemble methods are useful for this purpose.

In the Section 2, we will discuss the methods that are employed in this paper to analyze the Parkinson disease datasets. Results are presented in Section 3. Conclusion and future work are covered inn Section 4.

2 Methods

In this section, we will discuss the methods that are employed in the paper.

2.1 Decision Trees

Decision trees are very popular tools for classification [5,19]. The attractiveness of decision trees is due to the fact that decision trees represent rules. Rules can readily be expressed so that humans can understand them. A decision tree is in the form of a tree structure, where each node is either a leaf node (it indicates the value of the target class of examples) or a decision node (it specifies some test to be carried out on a single feature-value), with two or more than two branches and each branch has a sub-tree. A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the rules for classification of the example.

2.2 Bagging

(Bootstrap Aggregation)[4] generates different bootstrap training datasets from the original training dataset and uses each of them to train one of the classifiers in the ensemble. For example, to create a training set of N data points, it selects one point from the training dataset, N times without replacement. Each point has equal probability of selection. In one training dataset, some of the points get selected more than once, whereas some of them are not selected at all. Different training datasets are created by this process. When different classifiers of the ensemble are trained on different training datasets, diverse classifiers are created. Bagging does more to reduce the variance part of the error of the base classifier than the bias part of the error.

2.3 RvC Ensembles [2]

Ahmad [1] presented a discretization method, Extreme Randomized Discretization (ERD), for creating ensembles of decision trees. *In ERD, bin boundaries for the discretization are created randomly*. Ahmad et al. [2] used ERD to proposed an ensemble method for RvC. They used ERD in stage (1) of RvC. As it creates diverse datasets, different classifiers were created. Uncorrelated models are the keys to the success of any ensemble method [17]. Hence, their method is useful for wide range of problems. They presented two versions of their method, in the first one (unweighted version) the final result of each model is combined by simple averaging whereas in the second one (weighted version) the models with the smaller final bins are given more weight. They show theoretically and experimentally that their method perform better than equal-width discretization method.

3 Experiments

The dataset we used was created by Athanasios Tsanas and Max Little of the University of Oxford, in collaboration with 10 medical centers in the US and Intel Corporation who developed the telemonitoring device to record the speech signals [24]. This dataset is composed of a range of biomedical voice measurements from 42 people with early-stage Parkinson's disease recruited to a six-month trial of a telemonitoring device for

Table 1. Root MSE for motor UPDRS. The average results are given, s.d. is given in bracket.

The number of bins	RvC with equal-width bins	RvC with ERD (ensembles)	RvC with ERD with weight (ensembles)	Bagging	Single Regression tree
2	9.5(0.2)	7.4(0.1)	7.5(0.1)	6.8(0.1)	7.7(0.1)
5	9.4(0.2)	6.8(0.1)	6.9(0.1)	6.8(0.1)	7.7(0.1)
10	9.6(0.1)	6.7(0.1)	6.9(0.1)	6.8(0.1)	7.7(0.1)
20	9.5(0.2)	6.8(0.1)	6.9(0.1)	6.8(0.1)	7.7(0.1)

Table 2. Root MSE for total UPDRS. The average results are given, s.d. is given in bracket.

The number of bins	RvC with equal-width bins	RvC with ERD (ensembles)	RvC with ERD with weight (ensembles)	Bagging	Single Regression tree
2	12.5(0.2)	8.9(0.1)	9.2(0.1)	8.9(0.1)	9.8(0.1)
5	12.3(0.2)	8.8(0.1)	9.1(0.1)	8.9(0.1)	9.8(0.1)
10	12.5(0.2)	8.9(0.1)	9.2(0.1)	8.9(0.1)	9.8(0.1)
20	12.4(0.2)	9.0(0.1)	9.3(0.1)	8.9(0.1)	9.8(0.1)

remote symptom progression monitoring. This dataset consists of 16 biomedical voice measures. There are 5875 data points in the dataset. The task is to predict the motor UPDRS and total UPDRS by using these voice measures. Tsanas et al. [24] used linear regression and regression trees to predict the motor UPDRS and total UPDRS. These results were compared with Clinician's motor UPDRS score and Clinician's total UPDRS score. Their results suggest that regression trees are more useful the prediction.

All the experiments were carried out by using WEKA software [13]. We did experiments with REP regression trees (available in WEKA software) with the Bagging module. The size of the ensembles was set to 100 for all the experiments. For RvC, we used unpruned J48 decision trees (the Weka implementation of C4.5 tree [19] as the classifier. The number of bins were varied from 2-20 for RvC methods. The experiments were conducted following 5×2 cross-validation [8]. A t-test with 95% confidence was used to compare two results.

Results (Root Mean Square Error(RMSE)), presented in Table 1 and Table 2, suggest that the proposed ensemble methods perform consistently better than a single model (RvC with equal width discretization method) and a single regression tree. This shows the effectiveness of the our approach. REP regression tree ensembles (with Bagging) performed statistically better than a single REP regression tree. This suggests that ensembles can be useful in predicting motor UPDRS and total UPDRS.

The comparative study, with REP regression trees ensembles, suggests that our RvC method without weight perform statistically similar to this method (with 5-20 bins for motor UPDRS and 2-20 bins for total UPDRS). This shows that the proposed methods are comparable to the method that is developed specifically for regression problems. This shows the effectiveness of our proposed ensemble method. Hence, we may use

classifier models with the proposed ensemble method to predict motor UPDRS and total UPDRS. Even though the theoretical study suggests that the weights should be useful for the proposed ensembles, we saw there were high RMSEs for this method. We investigated the reasons for this behaviour. We found that in our theoretical calculation, we have assumed that the classification error is 0, whereas in these experiments the average classification errors for different datasets were varying between 20% to 40%. This means we were giving weights to the results which were wrong. Hence, weights are not giving the advantage as expected.

In the proposed ensemble method, we used decision trees as the classifier, however, we may use any other classifier. The number of bins is an important variable, as a small number of bins lead to the better classification, however, the value represented by bins will be less representative of the values. If the number of bins is large, the number of points in each bin will be small; this leads to the poor classification accuracy. However, the value represented by bins will be more representative of the points in the bins. One may use cross validation to find out the best number of bins for the best regression results.

4 Conclusion

Early diagnosis is very important for better management of Parkinson disease. A remote speech test for Parkinson patients [24] has shown a great promise in detection of presence and severity of symptoms. We have to select proper techniques for the accurate prediction of UPDRS from these speech datasets. In this paper, we studied the use of tree ensembles for the prediction of UPDRS.

In supervised learning, the target values may be continuous or a discrete set of class. As UPDRS is a real number, hence its prediction is a regression problem. The continuous target values (the regression problem) can be transferred to a discrete set of classes (the classification problem). The discretzation process is a popular method to achieve this task. In this paper, we used a recently developed ensemble methods [2] for RvC problems. Experiments showed that the proposed RvC ensembles and regression tree ensembles performed statistically better than single regression tree. This suggests that ensembles methods can be used for accurate prediction of UPDRS. Experiments results also suggest that the RvC ensemble method (without weight) performed similar to the regression tree ensembles. This shows that the ensemble method is useful for regression problems. As the proposed method is independent of the choice of the classifier, various classifiers can be used with the proposed method to solve the regression method. In the paper, we carried out experiments with the decision trees, however in future we will do the experiments to study its effectiveness with other classifiers.

References

1. Ahmad, A.: Data Transformation for Decision tree Ensembles. Ph.D. thesis, School of Computer Science, University of Manchester (2010)
2. Ahmad, A., Halawani, S.M., Albidewi, I.: Novel Ensemble Methods for Regression Via Classification Problems. Expert Systems with Applications 39(7), 6396–6401 (2012)

3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer-Verlag New York Inc. (2008)
4. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
5. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth International Group, CA (1984)
6. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
7. de Rijk, M.C., Launer, L.J., Berger, K., Breteler, M.M., Dartigues, F., Baldereschi, Fratiglioni, L., Lobo, A., Martinez-Lage, J., Trenkwalder, C., Hofman, A.: Prevalence of Parkinson's Disease in Europe: A Collaborative Study of Population-based Cohorts. Neurology 54, 21–23 (2000)
8. Dietterich, T.G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation 10, 1895–1923 (1998)
9. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
10. Dougherty, J., Kahavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Proceedings of the Twelth International Conference on Machine Learning (1995)
11. Fayyad, U.M., Irani, K.B.: Multi-interval Discretization of Continuous valued Attributes for Classification Learning. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp. 1022–1027 (1993)
12. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1), 10–18 (2009)
14. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(10), 993–1001 (1990)
15. Indurkhya, N., Weiss, S.M.: Solving Regression Problems with Rule-based Ensemble Classifiers. In: ACM International Conference Knowledge Discovery and Data Mining (KDD 2001), pp. 287–292 (2001)
16. Jankovic, J.: Parkinson's Disease: Clinical Features and Diagnosis. J. Neurol. Neurosurg. Psychiatr. 79(4), 368–376 (2008)
17. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
18. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)
19. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
20. Samii, A., Ransom, B.R., Nutt, J.G.: Parkinson's Disease. Lancet 363(9423), 1783–1793 (2004)
21. Singh, N., Pillay, V., Choonara, Y.E.: Advances in the Treatment of Parkinson's Disease. Progress in Neurobiology 81, 29–44 (2007)
22. Torgo, L., Gama, J.: Regression by Classification. In: Advances in Artificial Intelligence, pp. 51–60 (1996)
23. Torgo, L., Gama, J.: Regression using Classification Algorithms. Intelligent Data Analysis 4(1), 275–292 (1997)
24. 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 57(4), 884–893 (2010)
25. Tumer, K., Ghosh, J.: Error Correlation and Error Reduction in Ensemble Classifiers. Connect. Sci. 8(3), 385–404 (1996)
26. Vapnik, V.: Statistical Learning Theory. Wiley-Interscience, New York (1998)

Behavior Pattern Recognition in Electric Power Consumption Series Using Data Mining Tools

Alynne C.S. de Queiroz and José Alfredo F. Costa

Universidade Federal do Rio Grande do Norte
Department of Electrical Engineering
59072-970 Natal, RN, Brasil
alynnesaraiva@hotmail.com, jafcosta@gmail.com

Abstract. The behavioral patterns identification is very important for time series analysis of energy consumption to assist planning activities and decision making, as well to seek improvements in service quality and financial benefits. In this paper we used a methodology based on data mining tools, including cluster analysis and time series representation. The Time Series Knowledge Mining [1] was adapted to the treatment of consumption electricity series. Results are shown in a case study with hourly consumption measurements of eight power substations.

Keywords: Time Series, Temporal Data Mining, Consumption Electricity.

1 Introduction

Large amounts of data are available in business, scientific, military and industrial applications. Much of the data from the energy sector involve temporal aspects, which most common data are series from the observation of electrical variables over time.

Through the analysis of electric energy consumption series of different areas it is possible to recognize patterns of consumption that will indicate similarities and differences in the energy demand dynamics in different demanding regions.

In contrast with the static data analysis, in the time series analysis we can observe the development of relationships between events over time. Mörchen [1] proposed a mining algorithm, called TSKM, exploring temporal relationships between data. The TSKM consists of a set of algorithms that extract temporal relationships in multivariate time series.

Data mining comprises procedures to aid knowledge discovery in databases, allowing the analysis of complex and difficult-solving problems through methods such as cluster analysis, data visualization, classification and estimation. It enables the discovery of information within the data that queries and reports cannot effectively reveal. The objective is diverse in order to gain knowledge through research of strategic information, pattern discovery, classification and association between them [17].

Clustering is the process of discovering groups within the data, based on similarities, with a minimal, if any, knowledge of their structure. An approach based on

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 522–531, 2012.

cluster analysis and TSKM is presented in this paper. The aim is to observe groups in energy consumption series and temporal relationships.

The remainder of the paper is organized as follows. Section 2 briefly overview the pertinent literature, describing time series clustering, time series knowledge representation and mining. Section 3 describe steps related to method, pattern recognition in electric power consumption series, such as preprocessing, finding aspects, tones, chords, phrases and rules extraction. Section 4 present results in a case study with hourly consumption measurements of eight substations in the city of New England, USA. Section 5 concludes the paper with final remarks and future directions.

2 Literature Review

2.1 Time Series Clustering

Time series clustering is to partition time series data into groups based on similarity or distance. For time series clustering the first step is to work out an appropriate distance/similarity metric, and then, at the second step use clustering techniques to find clustering structures [2].

In this context we can define different similarity measures. We use the Pearson Correlation Coefficient as a distance measure between sets because is a good choice for similar time series data with time shifts [3].

Ward's linkage is a method for hierarchical cluster analysis. The idea has much common with analysis of variance (ANOVA). The linkage function specifying the distance between two clusters is computed as the increase in the "error sum of squares" (ESS) after fusing two clusters into a single cluster. Ward's Method seeks to choose the successive clustering steps so as to minimize the increase in ESS at each step.

2.2 Pearson Correlation Coefficient

The correlation analysis is a important tool for the different areas of knowledge, not only as a final result, but as a stage in the use of other analytical techniques [4]. A prominent measure for time series comparison is the Pearson correlation coefficient, which measures the correlation ρ between two random variables X and Y:

$$\rho = \frac{COV\ (X,Y)}{\sigma_X \sigma_Y} = E\left[\left(\frac{x-\mu_x}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right)\right] = COV(Z_X, Z_Y) \tag{1}$$

In order to use the Pearson correlation coefficient as a dissimilarity measure for time series it is desirable to generate low values for positively correlated (and thus similar) series [5]. The Pearson dissimilarity measure is therefore defined as

$$d_P = 1 - \rho \tag{2}$$

In order to use this method it is necessary that the random variables are measured at the same time.

The Pearson coefficient describes the interdependence between the interpretation between series whose interpretation can be evaluated as follows:

- If $0.00 < |\rho| < 0.30$, low linear correlation exists;
- If $0.30 \leq |\rho| < 0.60$, moderate linear correlation exists;
- If $0.60 \leq |\rho| < 0.90$, there is a strong linear correlation;
- If $0.90 \leq |\rho| < 1.00$, there is a very strong linear correlation;

2.3 Time Series Knowledge Representation (TSKR)

The language described in this section, called Time Series Knowledge Representation, uses a symbolic language to represent hierarchical knowledge of multivariate time series, exploring temporal operators and relations between them [5]. These temporal concepts are defined below [6]:

- Duration: the persistence or repetition of a property over several time points;
- Coincidence: describes the intersection of several intervals;
- Order: the sequential occurrence of time points or time intervals.

The input data model for the first mining step is a multivariate time series of possibly high dimensionality. Before defining the temporal representations it is necessary to know the different Aspects that describe a semantic property of the multivariate input time series represented by a subset of the dimensions. This is particularly useful if the dimensionality is large and correlation exists within subsets of the variables [6].

The first representation of the model corresponds to the concept of duration, where the occurrence of a state in a given time interval is called Tone. The second pattern type expresses the temporal concept of coincidence. An interval where several Tones occur simultaneously is described by a Chord. Finally, a Phrase expresses the temporal concept of partial order [6].

Figure 1 shows the hierarchical relationships of the different knowledge representations.

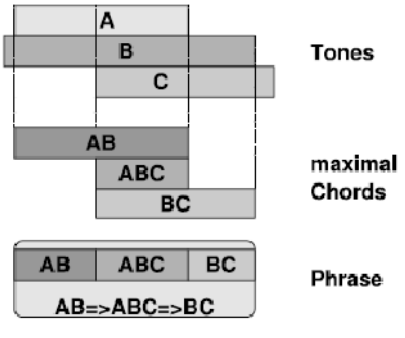


Fig. 1. Tones, Chords and Phrase representation [6]

2.4 Time Series Knowledge Mining (TSKM)

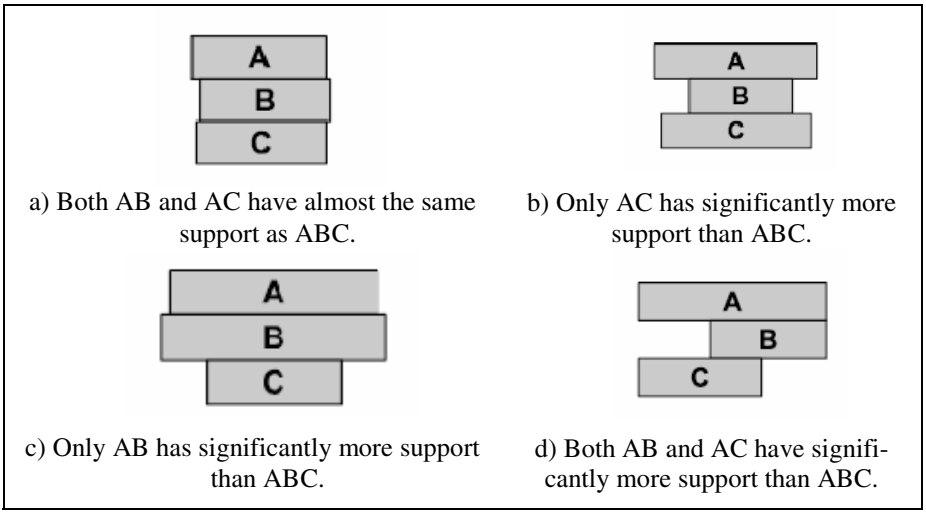
The TSKM describes algorithms for finding temporal knowledge in multivariate time series expressed with the TSKR.

The first task is to find Tones in each Aspect. The Tones values are found by partitioning the series with the bins values, whose intervals between their values can be labeled. The Persist discretization algorithm [1] optimizes the persistence index of the resulting states.

$$\boldsymbol{Persistence}(\boldsymbol{S_j}) = \boldsymbol{sgn}\left(\boldsymbol{A(j,j)} - \boldsymbol{P}(\boldsymbol{S_j})\right)\boldsymbol{SKL}\left(\boldsymbol{A(j,j)}, \boldsymbol{P}(\boldsymbol{S_j})\right) \quad (3)$$

where $A(j,j)$ is a persistence probability of state j, $P(S_j)$ is a transition probability of an event and SKL( ) is the product of the symmetric Kullback-Leibler divergence of the transition and marginal probability distribution for self vs. non-self with an indicator variable.

The algorithm used to mining Chords is based on the CHARM [7], an algorithm for mining closed itemsets. The main adaptation made by Mörchen [6] was to allow only one Tone for Aspect in a Chord, choosing the more representative Chord. Figure 2 shows the fours cases for mining more representative Chords.



a) Both AB and AC have almost the same support as ABC.

b) Only AC has significantly more support than ABC.

c) Only AB has significantly more support than ABC.

d) Both AB and AC have significantly more support than ABC.

Fig. 2. The four cases for mining margin-closed Chords [6]

In these first two steps of knowledge extraction are used filters to exclude very short tones and chords, preventing outliers and noise in the data.

The Phrases will represent the order partial concept, obtained by processing the Chords. To mine the sequences we used the Casas-Garriga algorithm [8] that extracts the sequences through the ordination of local relations items that occurs sequentially, called transactions.

3 Pattern Recognition in Electric Power Consumption Series

The process to mining patterns in electric power consumption series was divided into six stages:

- Preprocessing;
- Finding Aspects;
- Finding Tones;
- Finding Chords;
- Finding Phrases; and
- Rules Extraction

3.1 Preprocessing

The preprocessing step includes the removal of trend and seasonality components and data standardization.

The trend component of a time series shows its general tendency over many months in energy consumption series, through the increase or decrease of data values [9]. Seasonality is a trend that repeats itself systematically over time. In energy consumption series is common to observe the existence of recurring behaviors that characterize the seasonality due to the influence of exogenous factors [10]. Once extracted the trend and seasonality components we have access to the stochastic component of the data, generate by random factors.

For standardization the samples of the time series are transformed linearly so that they have mean zero and standard deviation one [11].

3.2 Finding Aspects

This step includes clustering and dimensionality reduction tasks.

The clustering task was performed using Ward Linkage clustering algorithm using the Pearson Correlation as the metric. In this work we chose to use the Ward hierarchical algorithm [12] for providing a graphical representation (dendrogram) to visualize natural groupings.

In the case where multivariate Aspects are formed it is necessary to use a combination of the series to obtain a univariate Aspect using Principal Component Analysis (PCA).

3.3 Finding Tones, Chords and Phrases

In this step we use the TSKM for extraction of different skill levels.

In the Finding Tones task labels were chosen according to the magnitude of bins values (high or low, for example). These labels are important to characterize the power system behavior and to indicate similarities between different Aspects.

In the Finding Chords task the concept of coincidence is extracted by comparing Tones of different Aspects. Thus, Tones occurred simultaneously were grouped together to form a Chord.

To find the Phrases were analyzed the Chords obtained in the previous step establishing a partial order among them.

3.4 Rules Extraction

The behavioral rules related common events to the Aspects or sequences of events that have probability of occurrence. From the Chords knowledge already have rules that describe local relationships of the data, expressing similarities between the Aspects. From the Phrases extracted global behaviors rules of the system, which express relationships of events sequence.

4 Evaluation

To evaluate the proposed method was done a case study using a multivariate time series composed of hourly measurements of energy consumption of eight supply zones of the city of New England (USA) in 2010, listed in Table 1[1].

Table 1. New England Zones

ID	Cod. Zone	Zone
1	ME	Maine Load Zone
2	NH	New Hampshire Load Zone
3	VT	Vermont Load Zone
4	CT	Connecticut Load Zone
5	RI	Rhode Island Load Zone
6	SEMASS	South East Massachusetts Load Zone
7	WCMASS	West Central Massachusetts Load Zone
8	NEMASSBOST	North East Massachusetts and Boston Load Zone

In the preprocessing step removed the trend and seasonality components. Data normalization was required because we use a clustering algorithm.

Table 2. Aspects

Aspect	Zones
A	RI, SEMASS e CT
B	NH, WCMASS e NEMASSBOST
C	ME e VT

Ward Linkage algorithm using the Pearson Correlation Coefficient was used in the experiments. Table 2 shows the division of the time series that compose each Aspect. As we have multivariate Aspects it is necessary to use the PCA algorithm [13] for obtaining univariate Aspects.

[1] Available in ISO New England site (`http://www.iso-ne.com`).

The Persist algorithm [14] was used to find tones, for each Aspect individually. To avoid the occurrence of very short tones were established the following parameters:

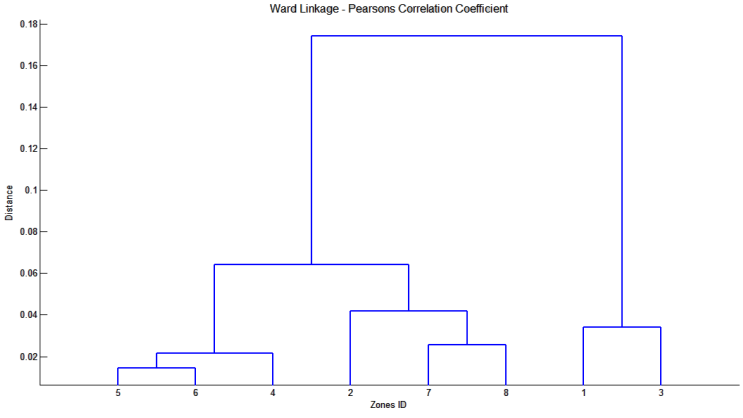


Fig. 3. Dendrogram using Ward Linkage algorithm and Pearson Correlation Coefficient

- **Minimum and maximum number of bins:** The Persist algorithm performs the data discretization in order to find an optimal value within the threshold fixed by the user. For this case were used 3 and 7, respectively.
- **Maximum interruption size and minimum tone size:** These parameters are important to filter interruption along the tones or very small tones which do not constitute an event. The values used were 0.1 and 0.5, repectively.

Figure 3 shows an example of Tones representation (Aspect A). Remember that the labels refer to the magnitude of consumption and the gaps between tones correspond to outliers.

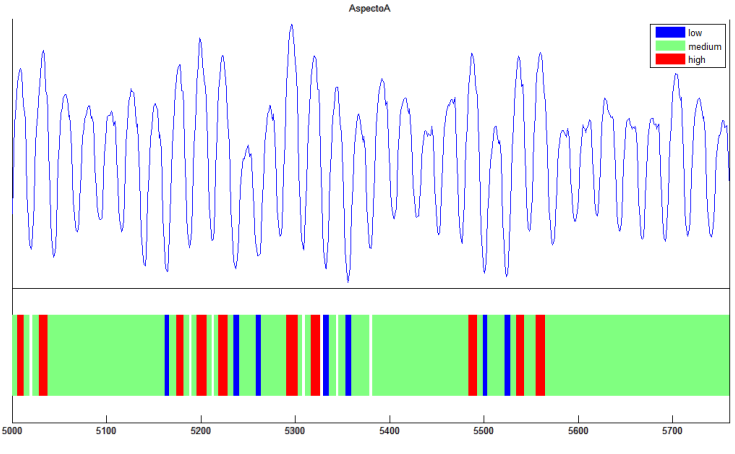


Fig. 4. Aspect A Tones

To mine Chords is necessary to compare the Tones of each Aspect, considering the same interval of occurrence, extracting rules that express the simultaneity concept.

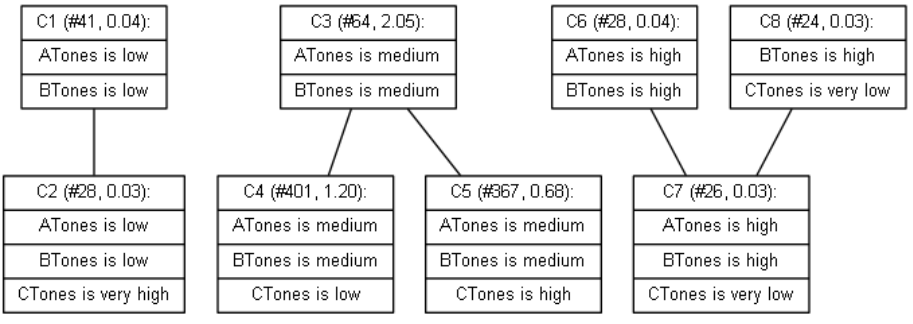


Fig. 5. Chords Representation

Figure 4 shows the eigth Chords and their combinations to form super-Chords. Watching the Chord C4 can extract the following rule:

Chord 'C4' *when* 'ATones is medium' *and* 'BTones is medium' *and* 'CTones is low' *coincide.*

Fig. 6. Textual Representation of a Chord

The next step is to convert Chords in Phrases, using the order partial concept between Chords. Sixteen sequences were found to describe the system, forming two Phrases. One of the Phrases generated can be viewed in Figure 5 where each state represents a Chord.

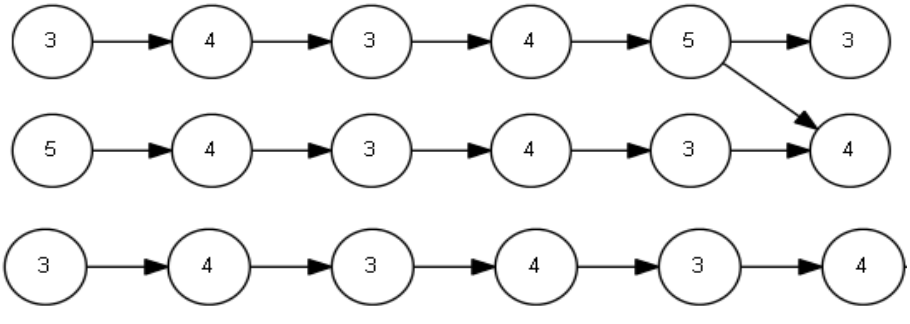


Fig. 7. Phrases Representation

The third Phrase can be represented by the rule shown in Figure 8, depicting the partial order among the Chords.

Phrase '3' *when* 'Chord 3' *then* 'Chord 4' *then* 'Chord 3' *then* 'Chord 4'…

Fig. 8. Textual Representation of a Phrase

5 Conclusion

The task of recognizing behavior patterns in energy consumption series is an activity that aims to systematize, explore, create and test possible visions about the consumption structure of a region. The method proposed in this paper is to draw similarities between zones, represent temporal concepts and extract behavior rules to serve as a reference for energy planning in a region.

The TSKM proved satisfactory for the representation of the power series behavior, relating the series and obtaining a routine operation of the dynamic power consumption.

Future research efforts will include and explore algorithms such as self-organizing maps for time series data clustering and visualization to aid understanding and rule extraction of consumer and demand relations in the energy sector.

References

1. Mörchen, F.: Algorithms For Time Series Knowledge Mining. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2006a)
2. Zhao, Y.: R and Data Mining: Examples and Case Studies, vol. IV. Elsevier, s.l. (2012)
3. Höppner, F., Klawonn, F.: Compensation of Translational Displacement in Time Series Clustering Using Cross Correlation. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J.-F. (eds.) IDA 2009. LNCS, vol. 5772, pp. 71–82. Springer, Heidelberg (2009)
4. Lira, S.A.: Análise de Correlação: Abordagem Teórica e de Construção dos Coeficientes com Aplicações. Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia da Universidade Federal do Paraná. s.n., Curitiba (Fevereiro 2004)
5. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM 26(11), 832–843 (1983)
6. Mörchen, F.: Time Series Knowledge Mining. s.n., Marburg (2006)
7. Zaki, M.J., Hsiao, C.-J.: CHARM: An efficient algorithm for closed itemset mining. In: Proceedings of the 2nd SIAM International Conference on Data Mining, SDM 2002 (2002)
8. Casas-Garriga, G.: Summarizing sequential data with closed partial orders. In: Proceedings of the 5th SIAM International Conference on Data Mining, SDM 2005 (2005)
9. Palit, A.K., Popovic, D.: Computational Intelligence in Time Series Forecasting - Theory and Engineering Applications, vol. 1. Springer, s.l. (2005)
10. Campos, R.J.: Previsão de séries temporais com aplicações a séries de consumo de energia elétrica. Dissertação de Mestrado UFMG (2008)
11. Alencar, A.B.: Mineração e Visualização de Coleções de Séries Temporais. Dissertação de Mestrado ICMC - USP (2007)

12. Everitt, B.S.: Cluster analysis, 3rd edn. Heinemann Educational Books. Academic Press, London (1993)
13. Jolliffe, I.T.: Principal Component Analysis. Springer, s.l. (1986)
14. Mörchen, F., Ultsch, A.: Optimizing time series discretization for knowledge discovery. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2005 (2005)
15. Maletzke, A.G.: Uma metodologia para extração de conhecimento em séries temporais por meio da identificação de motifs e da extração de caracteristicas. Dissertação de Mestrado ICMC-USP (2009)
16. Das, G., et al.: Rule discovery from time series. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, KDD 1998 (1998)
17. Goebel, M., Gruenwald, L.: A survey of data mining and knowledge discovery software tools. ACM SIGKDD 1(1), 20–33 (1999)

The Use of Artificial Neural Network in the Design of Metamaterials

Cristhianne F.L. Vasconcelos[1], Saulo L. Rêgo[1], and Rossana M.S. Cruz[2]

[1] Federal University of Rio Grande do Norte, Natal, RN, Brazil
cflinvas@yahoo.com.br, sauloregо88@gmail.com
[2] Federal Institute of Education, Science and Technology of Paraiba, João Pessoa, PB, Brazil
rossana.cruz@ifpb.edu.br

Abstract. This paper presents an analysis of the resonant characteristics of a composite medium based on a periodic array of interspaced conducting non-magnetic split ring resonators and continuous thin wires. The medium exhibits simultaneously negative values of effective permeability and permittivity within a microwave frequency band, characterizing a metamaterial. An analysis using Artificial Neural Networks is performed to obtain the permeability and permittivity as a function of resonant frequencies for a given geometrical dimensions of the metamaterial, for an optimization of the development medium.

Keywords: Metamaterials, Artificial Neural Network.

1 Introduction

Metamaterials are artificial materials consisting of regular arrays of metal, with dimensions much smaller than the wavelength in question, embedded in a dielectric medium such that simulates an effective constituent with specific parameters and refractive index negative different materials that constitute it. The growing interest in the study of metamaterials is given by the propagation characteristics totally different from the known conventional means, such as negative refractive index, phase inversion, reversal of Doppler shift, to name a few. Several papers have been published in the literature by exploring new effects and to consider new applications for this "fantastic" material. Among the applications can be mentioned developing superlens capable of providing images of objects or structures that are much smaller than the wavelength of light and invisible mechanisms for certain frequencies of electromagnetic oscillations, able to make the object invisible human eye. Other applications include the manufacture of LHMS antennas with new properties, optical nanolithography, high-resolution microscopes, nanocircuits for supercomputers, MIMO WiFi devices, ultra-compact in various bands and modes of communication speed, range and mobility higher.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 532–539, 2012.

This work also describes MLP-ANN models for applied to design the cell unit of metamaterial composed of Split Ring Resonators (SRR) and Thin-Wire (TW), with presented negative parameters, such as permittivity electric and permeability magnetic. The purpose is to use the MLP-ANN model in order to overcome the computational requirements associated with numerical simulations. The analysis of the metamaterial and MLP-ANN were implemented in MATLAB® for the optimal metamaterial synthesis.

This paper is organized in the following way. In Section 2, are presented the analytical models for design of metamaterial and the MLP-ANN used. In Section 3, numerical results are presented and discussed to verify our theoretical predictions. The conclusions are presented in Section 4.

2 Formulation

The analysis is based on analytical models [5] as a supporting tool for the design of metamaterials that attend electromagnetic requirements, within an acceptable accuracy, simplicity and agility. In the next step, we make use of Artificial Neural Networks (ANN), type Multilayer Perceptrons (MLP), to improve the process for developing metamaterials. The ANN are usually applied as an alternative to current methods of analysis, because they are trainable systems capable to learn, and most importantly, to generalize the acquired knowledge from any type of data available, while maintaining the accuracy of the original technique used and combining the low computational cost of neural models [8]-[10].

2.1 Metamaterial Solution

The electrical and magnetic properties of materials can be determined by two constitutive parameters called permittivity (ε) and permeability (μ). Together, the permeability and permittivity determine the response of the material when an electromagnetic wave propagates through it. A periodic array of conducting elements can behave as an effective medium for electromagnetic scattering when the wavelength is much longer than both the element dimension and lattice spacing.

The study of an artificial medium can be performed using analytical models for determining the effective permittivity (ε_{eff}) and effective permeability (μ_{eff}) of metamaterials [1]-[7]. Fig. 1(a) shows a homogeneous structure constituted by Thin-Wires (TW) with negative-ε/positive-μ. The metamaterial shown in Fig. 1(b) is composed of Split Ring Resonators (SRR) with positive-ε/negative-μ. The average cell size p of these homogeneous structures is much smaller than the guided wavelength. The analysis considers that the periodic arrangement of metallic elements is embedded in a dielectric with permittivity ε_r and permeability μ_r, composed of arrays of resonant cells each of which contains the SRR and/or TW.

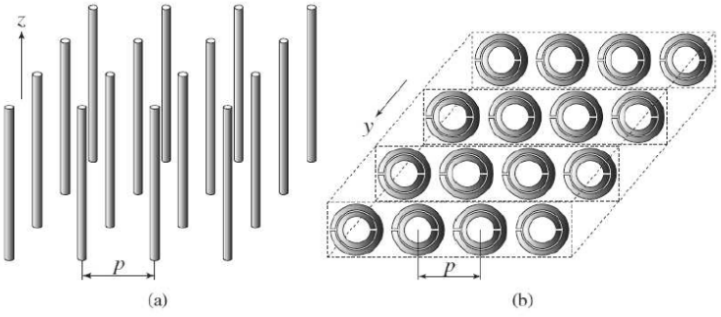


Fig. 1. (a) Structure composed of metal thin-wire (TW). (b) Structure composed of split ring resonator (SRR). (Source: [6])

If the excitation magnetic field H is perpendicular to the plane of the rings of the SRR structure, shown in Fig 1b, resonating currents are induced in the loop and generate equivalent magnetic dipole moments. This medium exhibits a plasmonic-type permeability frequency function given by [5]:

$$\varepsilon_{\text{eff}}(\omega) = 1 - \frac{\omega_{\text{ep}}^2}{(\omega^2 + j\omega\zeta_e)} \tag{1}$$

If the excitation electric field E is parallel to the axis of the metal thin-wires (TW) structure, shown in Fig 1a, so as to induce a current along them and generate equivalent electric dipole moments, this medium exhibits a plasmonic-type permittivity frequency function of the form [5]:

$$\mu_{\text{eff}}(\omega) = 1 - \frac{F\omega^2}{(\omega^2 - \omega_{0\text{m}}^2 + j\omega\zeta_m)} \tag{2}$$

2.2 Artificial Neural Network

In this work, Multilayer Perceptron (MLP) architecture is used in ANN model. The basic MLP building unit is a simple model of artificial neurons. This unit computes the weighted sum of the inputs plus the threshold weight and passes this sum through the activation function (usually sigmoid). In a MLP, the outputs of the units in one layer form the inputs to the next layer, as shown in Fig. 2. The weights of the network are usually computed by training the network using the back propagation algorithm. The MLP has a configuration of N layers, with the minimum number of layers equal to two, since it is necessary to have an input layer and an output layer, at least. Each layer is composed of M neurons, and the number of neurons may vary per layer [8]-[10]. The basic operation of each neuron is as follows. Each neuron has a number of entries equals to the number of data supplied to the neuron. Each input is multiplied by a weight w, and then is made the sum of the products, and this sum is inserted in a transfer function. The transfer sigmoid function is given by [8]:

$$\Delta w = \gamma * \Delta w_{ant} + (1-\gamma) * \alpha * gw \tag{3}$$

where γ is the momentum, α is the learning rate and gw is the gradient of the weight of input neuron. The gradient is directly related to the Mean Square Error (MSE) of the values used in training and simulated values for the network, the greater the gradient the faster the network converges to the expected results during the training phase can view the error is decreasing for each phase training, called epochs.

This work uses the method of Levenberg-Marquardt (LM) method, which makes the algorithm faster as compared to other training methods that use the Quasi-Newton method. LM method is used to find numerical solutions of nonlinear problems, with a capacity greater convergence.

The resulting model is expected to get input/output relations. The values of frequency and radius are used as input parameters for the neural network. The ANN will be trained to get the desired effective magnetic permeability and effective electric permittivity of the metamaterial.

One purpose of this work is to study the influence of some geometric parameters of the structure and the behavior of the final response. From these data an ANN is trained to predict the response of the structure due to the change of the considered parameter, within a given region of interest.

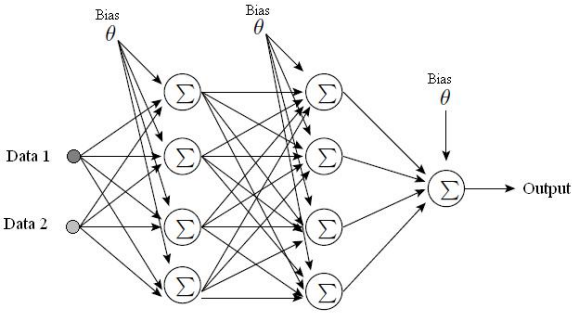


Fig. 2. Neural network MLP structure

3 Results

Based on the analytical formulation described in the last section, we determine the intrinsic properties of metamaterials, given by negative effective permittivity (ε_{eff}) and negative effective permeability (μ_{eff}). The results are presented for a metal periodic array composed of the SRR structure and the TW structure.

Considering the geometrical parameter of μ_{eff} as a function of the radius (r) in mm, and the resonant frequency (f_r) in GHz. Result of the typical frequency dependence of the μ_{eff} of an array of SRR is shown in Fig. 4, which displays the resonant behavior

predicted by Eq. (2). Results were obtained by setting that the SRR structure has period p = 50 mm, conductivity σ = 50 Ω/m, while the inner radius *r* is varied. It is seen that the frequency of the SRR structure shifts to a higher frequency as the *r* is reduced. In addition, it is also noted the effect of the radius variations on the effective permeability. Results obtained for a particular case, by setting *r* = 58 mm, are in agreement with those presented in [5].

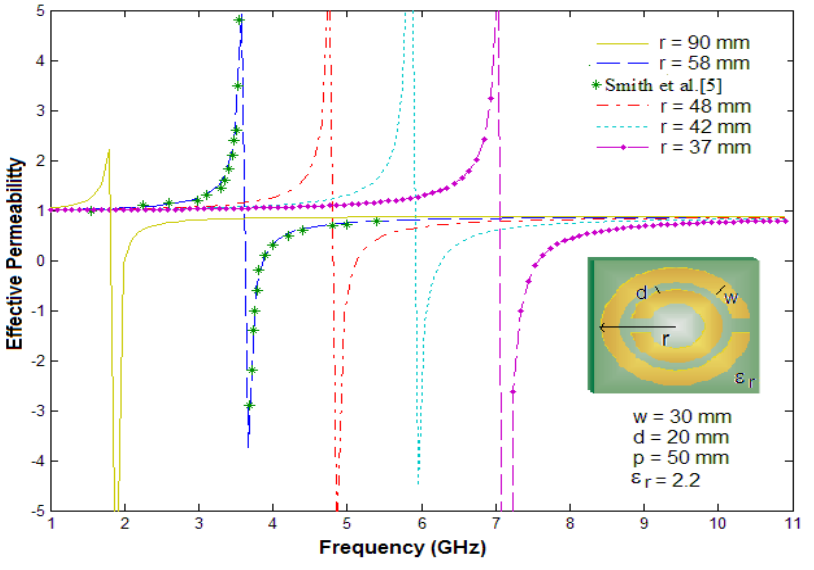


Fig. 3. Effective permeability as a function of frequency for the SRR

The methodology used by the MLP-ANN model is applied for the SRR medium (Fig. 1(b)), using the results obtained in Fig. 4 for r = 58 mm of the SRR considered in [5] for the training. Let us consider the values of *r* and the calculated results of frequency as input values and the calculated results of μ_{eff} as output values. The MLP-ANN are used to determine what value of μ_{eff} would correspond to a given *r* and frequency (whose values are between the maximum and minimum values provided as input). Fig. 5 presents the response after training for both the vectors used during the learning process and the untrained values by the network. For modeling, we used an MLP-ANN with two input, two hidden layers with ten neurons each, and one output. From the curves, it can be noted that the neuromodeling technique was able to extract and learn the pattern of variation for the frequency.

The training process is completed in 5000 epochs reaching a mean square error of $E_{med} = 10^{-5}$. Observe the accuracy of the simulated values compared to the desired ones. The mean square error of output values was falling as the network was trained; reaching a minimum value established when the training was concluded.

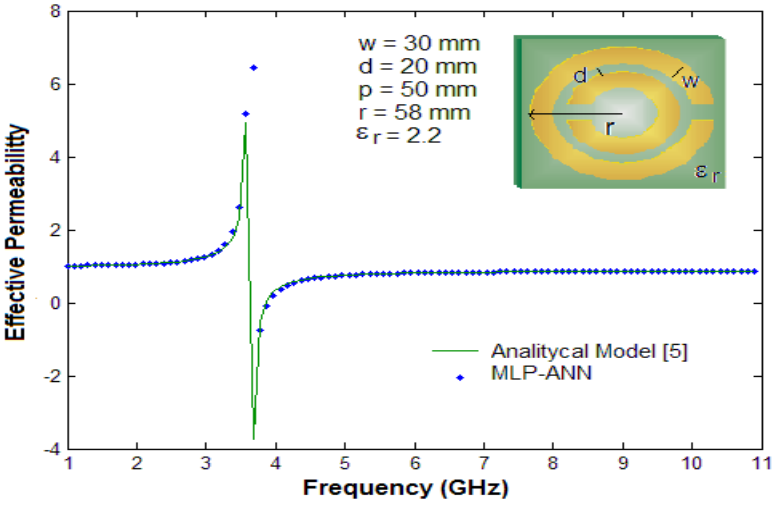


Fig. 4. Effective permeability as a function of frequency for the SRR medium from MLP-ANN model

In Fig. 5, the behavior of the effective permeabilitty (ε_{eff}) as a function of the radius (r) in mm, and the resonant frequency (f_r) in GHz, for a metallic periodic array of TW is presented. The structure has period p = 50 mm and conductivity σ = 100 Ω/m. The results were simulated for several wire radius values.

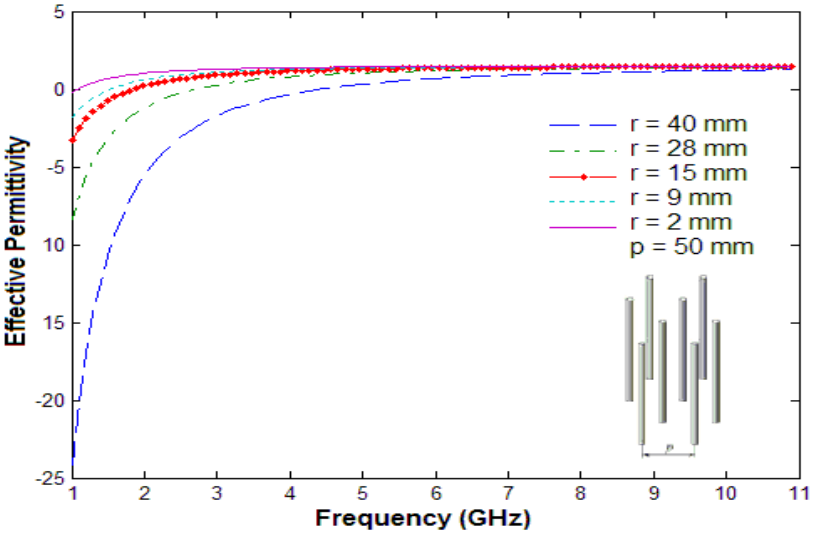


Fig. 5. Effective permittivity as a function of frequency for the TW medium

The methodology used by the MLP-ANN model is applied for a metal TW artificial medium (Fig. 1(a)), using the conventional neuromodeling technique [16]. In Fig. 6, the triangle symbols represent the 101 points per curve, obtained from simulations with five different values of radius *r*; the colored lines are the MLP model interpolation for each of the five curves with different values of *r*, and the black lines represent the MLP model generalization.

Random values for the effective permittivity were chosen within the search space used for the synthesis MLP network. We can see that MLP-ANN technique presented a good agreement with the electromagnetic analysis performed for ε_{eff} values, predicted by Eq. (1). In this way, we can reduce the computational cost needed for the implementation and accomplishment of several simulations, once the optimization technique proved to be a fast and accurate manner of performing the same analysis.

The mean square error was $9.0779.10^{-5}$, for 20000 epochs and a run time of 12.5424 minutes. The stop criterion used was the number of epochs. The run time of more than 12 minutes is due to the large number of points per curve (101 points) used for training. The MLP network used for the metal TW artificial medium was composed by three inputs (-1, r, f_r), twenty hidden neurons and one output (ε_{eff}). It is similar to that presented in Fig. 2, except for the number of hidden layers (01).

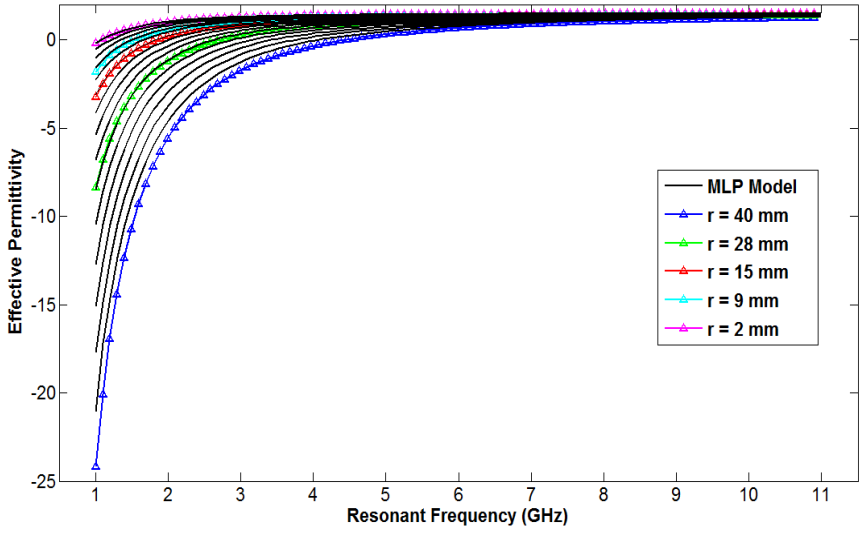


Fig. 6. Effective permittivity as a function of frequency for the TW medium from ANN model

4 Conclusion

This work presented an analysis of the resonant characteristics of a composite medium based on a periodic array of split ring resonators (SRR) and continuous thin wires (TW). Results for the effective permittivity and permeability have been presented, considering the variation of geometric parameters of the homogeneous

structures composed of SRR and TW. It was also verified the influence of geometric parameters on the resonant frequency of metamaterials. The obtained results were then used for training multilayer perceptrons (MLP) artificial neural networks (ANN), thus allowing the optimization of the constitutive parameters of the metamaterials, called effective permittivity and effective permeability.

The fast and accurate obtained results demonstrated the improvements with the utilization of these models, and proved that the MLP network global approximations are able to generalize. The characteristics of ANN models (precision, CPU efficiency and flexibility) can be perfectly used in association with these optimization techniques in order to develop powerful soft computing tools to design metamaterials.

Acknowledgments. This research was financially supported by the FAPERN and Federal University of Rio Grande do Norte.

References

1. Veselago, V.G.: The electrodynamics of substances with simultaneously negative values of ε and μ. Sov. Phys.—Usp. 10, 509–514 (1968)
2. Pendry, J.B., Holden, A.J., Robbins, D.J., Stewart, W.J.: Magnetism from conductors and enhanced nonlinear phenomena. IEEE Trans. Microw. Theory Tech. 47(11), 2075–2081 (1999)
3. Vasconcelos, C.F.L., Albuquerque, M.R.M.L., Silva, S.G., Oliveira, J.R.S., D'Assunção, A.G.: Full-wave analysis of annular ring microstrip antenna on metamaterial. IEEE Transactions on Magnetics 47(5) (2011)
4. Smith, D.R., Padilla, W.J., Vier, D.C., Nemat-Nasser, S.C., Schultz, S.: Composite Medium with Simultaneously Negative Permeability and Permittivity. Appl. Phys. Lett. 84(18), 4184–4187 (2000)
5. Smith, D.R., Vier, D.C., Kroll, N., Schultz, S.: Direct calculation of permeability and permittivity for a left-handed metamaterial. Appl. Phys. Lett. 77(14), 2246–2248 (2000)
6. Itoh, T., Caloz, C.: Electromagnetic metamaterials: transmission line theory and microwave applications. John Wiley&Sons, Inc., New Jersey (2006)
7. Haykin, S.: Neural networks: a comprehensive foundation. Macmillan College Publishing Company, New York (1994)
8. Silva, P.H.D.F., Cruz, R.M.S., D'Assunção, A.G.: Neuromodeling and natural optimization of nonlinear devices and circuits. In: Temel, T. (ed.) System and Circuit Design for Biologically-Inspired Intelligent Learning. IGI Global, Hershey (2010)
9. Zhang, Q.J., Gupta, C.: Neural networks for RF and microwaves design. Artech House, Norwood (2000)
10. Riedmiller, M., Braun, H.: A direct adaptative method for faster backpropagation learning: the RPROP algorithm. In: Proceedings of IEEE International Conference on Neural Networks, San Francisco, USA, pp. 586–591 (1993)

A Simulated Annealing Based Approach to the High School Timetabling Problem

George H.G. Fonseca, Samuel S. Brito, and Haroldo G. Santos

Graduate Program in Computer Science
Federal University of Ouro Preto
Diogo de Vasconcelos st. 150, 35930-000, Ouro Preto, Brazil
george@decea.ufop.br, {samuelsouza,haroldo}@iceb.ufop.br

Abstract. The High School Timetabling Problem remains subject of many research in Artificial Intelligence and Operational Research fields because of its hardness to solve and practical importance. A solution for this problem basically consists in the schedule of lessons to timeslots and the assignment of resources for these lessons. This work considers the solution of the problem of the ongoing Third International Timetabling Competition (ITC), which includes a diverse set of instances from many educational institutions around the world. We proposed an approach based on Simulated Annealing. One important structural feature of our approach is the use of the KHE engine to generate initial solutions combined with a multi-neighborhood search approach. The achieved results were encouraging: nine out of seventeen feasible solutions were found and five out of twenty one all-time best solutions were improved or matched in the processing limit stipulated in the ITC.

Keywords: Simulated Annealing, High School Timetabling Problem, Third International Timetabling Competition.

1 Introduction

The High School Timetabling Problem, denoted as Class×Teacher Timetabling Problem in the early works of [6], consists in the production of a schedule in a such way that no teacher or class attend more than one lesson at same time, respecting among other constraints the availability of teachers and prealocations.

The automated school timetabling has been the subject of much research in the fields of Artificial Intelligence and Operational Research. In [2] were presented some reasons to this interest:

- Hardness to solve. Find a timetabling that satisfies the interest of all elements involved is a hard task. Moreover, often the simple construction of a valid timetabling is a complicated task;
- Practical importance. A good timetabling can improve the staff satisfaction and allow the school to be more efficient in managing their resources. Moreover a good schedule can improve the students performance;

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 540–549, 2012.

– Theoretical importance. The problem addressed is classified as $\mathcal{NP}$-Hard [4] and progress in solving such problems is a major goal of current research in Computer Science and Mathematics.

Methods based in Integer Programming were proposed to the problem [15], but they can only solve exactly a subset of instances of the problem in a viable processing time. Nowadays, metaheuristic approaches were commonly applied to the problem [12] [1] [11].

In this sense, the main goal of the presented work is to propose a Simulated Annealing based approach to the model of High School Timetabling Problem proposed by the ongoing Third International Timetabling Competition 2011 (ITC 2011). For this event, a set of diverse instances from many institutions around the world were collected and specified the XHSTT format [14], allowing for the first time the extensive experimental evaluation of solvers in a really rich set of instances [9].

The remaining of this work is organized as follows: in Section 2 it will be presented the model of High School Timetabling Problem proposed by ITC 2011. In Section 3 it will be presented the solution approach. Section 4 presents computational experiments and finally, in Section 5, concluding remarks are presented.

2 High School Timetabling Problem

Used in the ITC 2011, the addressed model of High School Timetabling Problem came up with the goal of providing a generic model capable to address the various features of the High School Timetabling Problem around the world [7] [17] [13] [16] [3] [15] [14]. The model is split in three main entities:

2.1 Times and Resources

The time entity consists of a timeslot or a set of timeslots (time group). The resources, in turn, are divided in three main categories: students, teachers and rooms [14]:

students: a group of students attends to an event (lesson). Important constraints to the students are control their idle times and the number of lesson by day;

teachers: a teacher can be pre-assigned to attend an event. In some cases it is not pre-assigned and should be assigned according to their qualifications and workload limits;

rooms: most of events take place on a room. One room has a certain capacity and a set of features.

2.2 Events

An event is the basic unit of allocation, representing a simple lesson or a set of lessons (event group). A timeslot assignment to a event is called meet and a

resource assignment to a event is called task. The term course is used to designate a group of students who attend the same events. Other kinds of events, like meetings are allowed by the model [14]. An event has the following attributes:

duration that represents the number of timeslots which have to be allocated to the event;
course related to the event;
pre-assigned resources to attend the event (optional);
workload that will be added to the total workload of resources assigned to the event (optional);
pre-assigned timeslots some events yet present the timeslot in which will be allocated (optional).

2.3 Constraints

Post [14] groups the constraints in three categories: basic constraints of scheduling, constraints to the events and constraints to the resources. The objective function $f(.)$ is calculated in terms of violations to each constraint penalized according to their weight (so this is a minimization problem). They were also divided in hard constraints, whose attendance is mandatory; and soft whose attendance is desirable but not mandatory. Each instance can define whether a constraint is hard or soft. For more details, see [14].

Basic Constraints of Scheduling

1. AssignTimeConstraint: assign timeslots to each event;
2. AssignResourceConstraint: assign the resources to each event;
3. PreferTimesConstraint: indicates that some event have preference for a particular timeslot(s);
4. PreferResourcesConstraint: Indicates that some event have preference for a particular resource(s).

Constraints to Events

1. LinkEventsConstraint: schedule a set of events to the same starting time;
2. SpreadEventsConstraint: specify the allowed number of occurrences for event groups in time groups between a minimum a maximum number of times; this constraint can be used, for example, to define a daily limit of lessons;
3. AvoidSplitAssignmentsConstraint: for each event, assign a particular resource to all of his meets;
4. DistributeSplitEventsConstraint: for each event, assign between a minimum and a maximum meets of a given duration;
5. SplitEventsConstraint: limits the number of non-consecutive meets that an event should be scheduled and his duration.

Constraints to Resources

1. AvoidClashesConstraint: assign the resources without clashes (i.e. without assign the same resource to more than one event at a timeslot);
2. AvoidUnavailableTimesConstraint: avoid assigning resources on the times that they are not available;
3. LimitWorkloadConstraint: schedule the workload of the resources between a minimum and a maximum bound;
4. LimitIdleTimes: the number of idle times in each time group should lie between a minimum and a maximum bound for each resource; typically, a time group consists of all timeslots of a given week day;
5. LimitBusyTimesConstraint: the number of busy times in each time group should lie between a minimum and a maximum bound for each resource;
6. ClusterBusyTimesConstraint: the number of time groups with a timeslot assigned to a resource should lie between a minimum and a maximum limit; this can be used, for example, to concentrate teacher's activities in a as few days as possible.

3 Solution Approach

Our approach uses the KHE school timetabling engine[9] to generate initial solutions and the metaheuristic Simulated Annealing perform local search around this solution. These two elements will be explained in the following subsections.

3.1 Build Method

The KHE is a platform for handling instances of the addressed problem which also provides a solver, used to build initial solutions in the presented approach. The KHE's solver was chosen to generate the initial solutions since it is able to find good initial solutions efficiently (see Table 2).

The incorporated solver is based on the concept of Hierarchical Timetabling [8], where smaller allocations are joint to generate bigger blocks of allocation until a full representation of the solution is developed.

The Hierarchical Timetabling is supported by Layer Tree data structure [8], consisting of nodes that represent the required meet and task allocation. An allocation may appear in at most one node. A Layer is a subset of nodes within the propriety that none of them can be overlapped in time. Commonly, nodes are grouped in a Layer when share resources.

The hard constraints of the problem are modeled to this data structure and then a Matching problem is solved to find the times/resources allocation. The Matching is done by connecting each node to a timeslot or resource respecting the property of Layer. For full details, see [8,9].

3.2 Neighborhood Structure

Five neighborhood structures were used:

- ES (EventSwap): two events e_1 and e_2 have their timeslots t_1 and t_2 swapped;
- EM (EventMove): an event e_1 is moved from t_1 to another timeslot t_2;
- EBM (EventBlockMove): like es, swaps the timeslot of two events e_1 and e_2, but if the events have different duration, e_1 is moved to the following the last timeslot occupied by e_2.
- RS (ResourceSwap): two events e_1 and e_2 have their assigned resources r_1 and r_2 swapped. Resources r_1 and r_2 should play the same role (e.g. both have to be teachers).
- RM (ResourceMove): an event e_1 have his assigned resource r_1 replaced to a new resource r_2.

3.3 Simulated Annealing Implementation

Proposed by [10], the metaheuristic Simulated Annealing is a probabilistic method on an analogy to thermodynamics simulating the cooling of a set of heated atoms. This technique starts its search from any initial solution. The main procedure consists of a loop that randomly generates, at each iteration, one neighbor $s^{'}$ of the current solution s.

Let Δ be the variation of the objective function value incurred from moving to the candidate neighbor ($\Delta = f(s^{'}) - f(s)$). The method accepts the move and the neighbor becomes the new current solution if $\Delta < 0$. If $\Delta \geq 0$ the neighbor can also be accepted, but in this case, with a probability $e^{-\Delta/T}$, where T is a method parameter, called temperature and regulates the probability to accept solutions with higher cost.

The temperature can assume, initially, a high value T_0. After a fixed number of iterations $SAmax$ (that represents the number of iterations needed to the system reaches the thermic equilibrium at a given temperature), the temperature is gradually lowered by a cooling rate α, such that $T_k \leftarrow \alpha \times T_{k-1}$, and $0 < \alpha < 1$. With this procedure, a greater chance to avoid local minimum occurs at the initial iteration and as T approaches zero, the algorithm behave like a descent method as it reduces the likelihood of accepting worsen movements[5].

The developed implementation of Simulated Annealing will be described in Algorithm 1, where $f(.)$ denotes the objective function and $N(.)$, the neighborhood structure. The considered parameters were $\alpha = 0.5$ and $T_0 = 5$. The parameter $SAmax$ was defined according to the number of events (nE) for each instance set by a multiplier. If the initial solution is feasible (i.e. there is no hard constraint violation), $SAmax = nE \times 10$, otherwise, $SAmax = nE \times 100$.

At each iteration the selected movement can be from any of the proposed neighborhoods. If the instance requires the resource assignment (i.e. have a constraint of kind *AssignResourceConstraint*), the kind of neighborhood is chosen based on the following probabilities: $es = 0.2$, $em = 0.4$, $ebs = 0.1$, $rs = 0.2$ and $rm = 0.1$, otherwise, the neighborhood es and er and the odds are: $es = 0.5$, $em = 0.3$ and $ebs = 0.2$. These values were empirically adjusted.

Algorithm 1. Developed implementation of Simulated Annealing

```
Input: f(.), N(.), α, SAmax, T_0, s, timeout
Output: Best solution s found.
s* ← s;
IterT ← 0;
T ← T_0;
while T > 0  e  elapsedTime < timeout do
    while IterT < SAmax do
        IterT ← IterT + 1;
        Generate a random neighbor s' ∈ N(s);
        Δ = f(s') − f(s));
        if Δ < 0 then
            s ← s';
            if f(s') < f(s*) then s* ← s';
        else
            Take x ∈ [0, 1];
            if x < e^{−Δ/T} then s ← s';
    T ← α × T;
    IterT ← 0;
s ← s*;
return s;
```

4 Computational Experiments

All experiments ran on a Intel ® i5 2.4 Ghz computer with 4GB of RAM computer under Ubuntu 11.10 operating system. The programming language used on software development was C++ compiled by GCC 4.6.1. The processing time was adjusted according to the benchmark available from Third International Timetabling Competition 2011. The timeout was set at 1000 seconds (normalized), like required in the second phase of ITC 2011. All of our results was validated by HSEval validator `http://sydney.edu.au/engineering/it/~jeff/hseval.cgi`.

4.1 Dataset Characterization

The set of instances available from ITC 2011 `http://www.utwente.nl/ctit/hstt/archives/XHSTT-2012` was originated from many countries and ranges from small instances to huge challenging ones. The Table 1 presents the main features of the instances.

4.2 Obtained Results

Table 2 presents the results obtained by the solution approach. The results are expressed by pair x/y, where x contains the infeasibility measure and y the

Table 1. Features of instances from ITC2011

Instance	Timeslots	Teachers	Rooms	Class	Events	Duration
AustraliaBGHS98	40	56	45	30	387	1564
AustraliaSAHS96	60	43	36	20	296	1876
AustraliaTES99	30	37	26	13	308	806
BrazilInstance1	25	8		3	21	75
BrazilInstance4	25	23		12	127	300
BrazilInstance5	25	31		13	119	325
BrazilInstance6	25	30		14	140	350
BrazilInstance7	25	33		20	205	500
EnglandStPaul	27	68	67	67	1227	1227
FinlandArtificialSchool	20	22	12	13	169	200
FinlandCollege	40	46	34	31	387	854
FinlandHighSchool	35	18	13	10	172	297
SecondarySchool	35	25	25	14	280	306
GreeceHighSchool1	35	29		66	372	372
GreecePatras3rdHS2010	35	29		84	178	340
GreecePreveza3rdHS2008	35	29		68	164	340
ItalyInstance1	36	13		3	42	133
NetherlandsGEPRO	44	132	80	44	2675	2675
NetherlandsKottenpark2003	38	75	41	18	1156	1203
NetherlandsKottenpark2005	37	78	42	26	1235	1272
SouthAfricaLewitt2009	148	19	2	16	185	838

quality measure. The column ITC 2011 $f(s^*)$ presents the best known solution to the problem instance.

The column SA contains the results obtained by solution approach. The results were collected considering five executions of the method. Thus $f(s^*)$ presents the best solution found, $f(\overline{s})$ presents the average, σ presents the standard deviation of results and t_s informs the average time (in seconds) and also the standard deviation. The bold results were better to or equal to the best known solution.

The column SA_{free} presents the results obtained by Simulated Annealing applied to the best known solution to the instance without time limitation. This aims at the first phase of ITC 2011 where neither does not matter time spent nor technologies used to obtains all-time best solutions. To this column were considered only one execution. The bold results highlight improvements that our method was able to make over the best known solution.

4.3 Discussion of Results

Our heuristic approache used of KHE solver as build procedure and the a diverse set of neighborhood kinds. The KHE solver was able to quickly find good solutions and the proposed neighborhood structures were able to consistently explore the solutions space and perform significant improvements in te initial solution.

Table 2. Obtained Results

Instance	ITC 2011 $f(s^*)$	KHE Solver		Simulated Annealing				$\mathbf{SA}_{free}$
		$f(s)$	t_s	$f(s^*)$	$f(\overline{s})$	σ	$\overline{t_s}$	
AustraliaBGHS98	7.0 / 433.0	11.0 / 415.0	16.7	11.0 / 375.0	11.0 / 381.6	±0.0 / ±5.7	147.5±3.5	**4.0 / 367.0**
AustraliaSAHS96	23.0 / 44.0	11.0 / 15.0	61.3	**11.0 / 11.0**	11.0 / 11.8	±0.0/ ±1.3	167.3±1.6	**22.0 / 44.0**
AustraliaTES99	26.0 / 134.0	6.0 / 418.0	4.7	**6.0 / 148.0**	6.0 / 148.0	±0.0 / ±0.0	72.1±0.4	**25.0 / 124.0**
BrazilInstance1	0.0 / 24.0	0.0 / 81.0	0.0	0.0 / 36.0	0.0 / 39.6	±0.0 / ±6.5	0.7±0.0	**0.0 / 15.0**
BrazilInstance4	0.0 / 112.0	17.0 / 225.0	0.7	12.0 / 150.0	12.0 / 159.0	±0.0 / ±12.0	4.8±0.4	**0.0 / 103.0**
BrazilInstance5	0.0 / 225.0	12.0 / 258.0	0.0	6.0 / 149.0	7.0 / 167.4	±1.0 / ±20.7	4.7±1.2	**0.0 / 210.0**
BrazilInstance6	0.0 / 209.0	11.0 / 339.0	1.0	5.0 / 258.0	5.6 / 262.8	±0.5 / ±10.3	5.3±0.0	**0.0 / 173.0**
BrazilInstance7	0.0 / 330.0	20.0 / 429.0	1.3	13.0 / 285.0	14.6 / 281.4	±0.9 / ±11.5	8.1±0.4	**0.0 / 318.0**
EnglandStPaul	0.0 / 18444.0	0.0 / 49756.0	77.3	**0.0 / 15208.0**	0.0 / 17916.8	±0.0 / ±1991.0	592.1±3.7	**0.0 / 11732.0**
FinlandArtificialSchool	0.0 / 0.0	16.0 / 18.0	1.3	10.0 / 8.0	13.0 / 8.0	±1.7 / ±2.5	53.7±2.3	0.0 / 0.0
FinlandCollege	0.0 / 0.0	20.0 / 747.0	2.7	4.0 / 109.0	5.6 / 114.2	±1.1 / ±15.1	157.3±2.4	0.0 / 0.0
FinlandHighSchool	0.0 / 1.0	7.0 / 446.0	1.3	0.0 / 74.0	1.4 / 89.4	±1.1 / ±17.8	59.2±1.8	0.0 / 0.1
FinlandSecondarySchool	0.0 / 106.0	35.0 / 301.0	17.3	3.0 / 114.0	3.8 / 121.2	±0.8 / ±28.4	164.5±1.9	0.0 / 106.0
GreeceHighSchool1	0.0 / 0.0	0.0 / 0.0	68.7	**0.0 / 0.0**	0.0 / 0.0	±0.0 / ±0.0	73.5±1.9	0.0 / 0.0
GreecePatras3rdHS2010	0.0 / 0.0	8.0 / 399.0	8.7	0.0 / 80.0	0.6 / 91.2	±0.9 / ±19.1	97.9±2.4	0.0 / 0.0
GreecePreveza3rdHS2008	0.0 / 0.0	6.0 / 684.0	8.0	0.0 / 189.0	1.4 / 165.4	±1.1 / ±20.5	87.4±0.8	0.0 / 0.0
ItalyInstance1	0.0 / 28.0	0.0 / 323.0	1.3	0.0 / 34.0	0.0 / 40.4	±0.0 / ±4.5	2.7±0.0	0.0 / 28.0
NetherlandsGEPRO	1.0 / 566	2.0 / 69118	926.0	2.0 / 6095.0	2.0 / 18511.0	±0.0 / ±3641.4	1000.0±0.0	**1.0 / 549.0**
NetherlandsKottenpark2003	0.0 / 1410.0	1.0 / 72413.0	264.7	0.0 / 5481.0	0.2 / 7224.0	±0.4 / ±1185.4	1000.0±0.0	**0.0 / 1189.0**
NetherlandsKottenpark2005	0.0 / 1078.0	18.0 / 22284.0	346.7	10.0 / 2783.0	11.6 / 3045.8	±0.9 / ±270.3	1000.0±0.0	**0.0 / 963.0**
SouthAfricaLewitt2009	0.0 / 58.0	361.0 / 0.0	16.0	**0.0 / 20.0**	0.4 / 28.4	±0.5 / ±8.9	51.6±0.9	**0.0 / 42.0**

As Table 2 shows, our approach was able to satisfactorily process even huge instances problems, improving sometimes the best known solutions, in the small provided timeout. From small to medium instances the approach was very quick. The standard deviation of the processing times also was low, showing consistence of method in terms of time spent.

For some instances, even the production of feasible solutions to configures a very hard task. These instances commonly define most of constraints as hard constraints. Therefore, ITC 2011 do not expect that a solver could find all feasible solutions and the use of pair $infeasibility/quality$ was encouraged. Our approach was able to reach nine out of seventeen [1] feasible solutions in the given timeout.

Our approach was able to improve or match the best known solution on five out of twenty one instances. It is noteworthy that there were no time or technology constraints to the original best known solutions. The proposed approach, even under the given timeout, was able to improve some best known results and get close to the others.

Taking the best known solution as initial solution without time limits, our local search technique has able to improve thirteen out of sixteen [2] solutions. Some of these results was the best achieved on the first phase of ITC 2011.

5 Concluding Remarks

Our Simulated Annealing based metaheuristic was able to improve or match the all-time best solution from ITC-2011 to five out of twenty one instances in a tiny processing time. Taking the best known solution as input of our method make him capable to improve thirteen out of sixteen instances. These results are encouraging due to the problem hardness and practical importance.

Nevertheless, we believe that there is still room for improvement in our approach. Some possible future works are (1) develop new additional neighborhood categories able to make more significant structural changes to a solution in restricted amount of movements, like kempe moves and ejection chains [11]; (2) develop and perform a computational study of other metaheuristics such as Variable Neighborhood Search to the problem, which could exploit available neighborhoods in different ways; and (3) develop a Mixed Integer Programing heuristic to the problem.

References

1. Barbosa, S.H., Souza, S.R.: Resolução do problema de programação de cursos universitários baseada em currículos via uma meta-heurística híbrida grasp-ils-relaxado. In: Proceedings of XLIII SBPO XLIII Simpósio Brasileiro de Pesquisa Operacional. SOBRAPO, Ubatuba, pp. 1:2827–1:2882 (2011)

[1] The remaining four instances were not taken into count because there is no guarantee that a feasible solution exists.

[2] The remaining five instances already had a zero cost known solution.

2. Bufé, M., Fischer, T., Gubbels, H., Häcker, C., Hasprich, O., Scheibel, C., Weicker, K., Weicker, N., Wenig, M., Wolfangel, C.: Automated Solution of a Highly Constrained School Timetabling Problem - Preliminary Results. In: Boers, E.J.W., Gottlieb, J., Lanzi, P.L., Smith, R.E., Cagnoni, S., Hart, E., Raidl, G.R., Tijink, H. (eds.) EvoIASP 2001, EvoWorkshops 2001, EvoFlight 2001, EvoSTIM 2001, EvoCOP 2001, and EvoLearn 2001. LNCS, vol. 2037, pp. 431–440. Springer, Heidelberg (2001)
3. de Haan, P., Landman, R., Post, G., Ruizenaar, H.: A Case Study for Timetabling in a Dutch Secondary School. In: Burke, E.K., Rudová, H. (eds.) PATAT 2007. LNCS, vol. 3867, pp. 267–279. Springer, Heidelberg (2007)
4. Garey, M.R., Jonhson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
5. Glover, F., Kochenberger, G.: Handbook of Metaheuristics. International Series in Operations Research & Management Science. Kluwer Academic Publishers (2003)
6. Gotlieb, C.C.: The construction of class-teacher time-tables. In: Proc. IFIP Congress, pp. 73–77. North Holland Pub. Co., Munich (1963)
7. Kingston, J.H.: A Tiling Algorithm for High School Timetabling. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 208–225. Springer, Heidelberg (2005)
8. Kingston, J.H.: Hierarchical Timetable Construction. In: Burke, E.K., Rudová, H. (eds.) PATAT 2007. LNCS, vol. 3867, pp. 294–307. Springer, Heidelberg (2007)
9. Kingston, J.H.: A software library for school timetabling. Disponível em (2012), http://sydney.edu.au/engineering/it/~jeff/khe/ (acessado em Abril de 2012)
10. Kirkpatrick, S., Gellat, D.C., Vecchi, M.P.: Otimization by simulated annealing. Science 202, 671–680 (1983)
11. Lú, Z., Hao, J.-K.: Adaptive tabu search for course timetabling. European Journal of Operational Research 200(1), 235–244 (2010)
12. Muller, T.: Itc2007 solver description: a hybrid approach. Annals OR 172(1), 429–446 (2009)
13. Nurmi, K., Kyngas, J.: A framework for school timetabling problem. In: Proceedings of the 3rd Multidisciplinary International Scheduling Conference: Theory and Applications, Paris, pp. 386–393 (2007)
14. Post, G., Ahmadi, S., Daskalaki, S., Kingston, J.H., Kyngas, J., Nurmi, C., Ranson, D.: An Xml Format for Benchmarks in High School Timetabling. Annals of Operations Research 3867, 267–279 (2010)
15. Santos, H.G., Uchoa, E., Ochi, L.S., Maculan, N.: Strong bounds with cut and column generation for class-teacher timetabling. Annals OR 194(1), 399–412 (2012)
16. Valourix, C., Housos, E.: Constraint programming approach for school timetabling. Computers & Operations Research 30, 1555–1572 (2003)
17. Wright, M.: School timetabling using heuristic search. Journal of Operational Research Society 47, 347–357 (1996)

Differential Evolution and Perceptron Decision Trees for Classification Tasks

R.A. Lopes[1], A.R.R. Freitas[1],
R.C. Pedrosa Silva[1], and Frederico Gadelha Guimarães[2]

[1] Graduate Program in Electrical Engineering, Federal University of Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil
{rodolfo.ufop,alandefreitas,rcpsilva}@gmail.com
[2] Department of Electrical Engineering, Federal University of Minas Gerais
Belo Horizonte, MG 31270-901, Brazil
fredericoguimaraes@ufmg.br

Abstract. Due to its predictive capacity and applicability in different fields, classification has been one of the most important tasks in data mining. In this task, the Perceptron Decision Trees (PDT) have been used with good results. Thus, this paper presents a Differential Evolution algorithm that evolves PDTs. Furthermore, we also present the concept of legitimacy which is used to reduce the costs of solution evaluation, a time consuming part of the algorithm. The experiments comparing our method with other seven well known classifiers, show that the proposed approach is competitive and has potential to build very accurate models. The best solutions found by it were the best ones in the majority of the tested databases.

Keywords: Evolutionary Computation, Data Classification, Differential Evolution, Perceptron Decision Trees.

1 Introduction

Due to its predictive capacity and applicability in different fields, classification has been one of the most important tasks in data mining. It consists, basically, in finding a model that describes and distinguishes data classes. The classification model is induced based on a training data set, then it is used to examine features of a new instance and assign to it a predefined class.

In this context, Perceptron Decision Trees (PDT) have been used in many real-world classification tasks with good results [1,2,3]. They are decision trees whose nodes test a linear combination of the attributes partitioning the input space by hyperplanes in general positions [4].

In this paper we show an approach to generate PDTs, based on a Differential Evolution (DE) algorithm. DE is an important Evolutionary Algorithm which works with a population of solutions and iteratively searches high quality solutions (Section 3). Thus, DE will work with a population of PDTs iteratively minimizing their classification errors.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 550–557, 2012.

By describing the methodology of this work, this paper introduces the following contributions:

- An algorithm based on Differential Evolution for evolving Perceptron Decision Trees (Section 4)
- An approach for representation and manipulation of PDT in the context of DE solutions (Section 4.1-4.2)
- A strategy for replacing solutions (Section 4.3), in which the worst classifiers give place to new ones
- A method for controlling the legitimacy of the evaluation (Section 4.4), which makes evaluation more legitimate when it seems to be necessary

Comparative tests with well known classifiers are presented in Section 5. Those results show that even though a more extensive study is needed, the proposed algorithm is very promising for building accurate classification models (Section 6).

2 The Classification Task

Classification consists in predicting the value of a user-specified goal attribute (the class) based on the values of other attributes, called predicting attributes. In a movie classification example, the goal attribute might be the *Movie Genre*, taking on the values (classes) "action" or "drama", while the predicting attributes might be the title, cast and soundtrack.

In the classification task we want to build a model which maps predicting attribute values to classes accurately. For this purpose, the data being mined is usually divided into two mutually exclusive data sets, the training set and the test set. In a first step, called training, the classification algorithm builds the model accessing the values of both the predicting attributes and classes of each example in the training set.

Once the training process is finished and the algorithm has built the model, its predictive performance is evaluated on the test set, which was not seen during the training phase. Thus, if the performance of the built model is good enough, it will be used to classify new instances for which the classification is not known.

3 Evolving Perceptron Decision Trees with Differential Evolution

Differential Evolution (DE) [5] is an important Evolutionary Algorithm (EA) developed to improve the quality of a solution over many iterations. In DE new candidate solutions are created by combining a parent individual with several other individuals of the same population. Then, this candidate solution replaces the parent if it has a better fitness value. EA have been successfully used to evolve classifiers, such as Oblique Decision Trees (ODT) [6].

Through the basic iterative process of DE, as the one described below, it is expected that a satisfactory solution will be found:

- Generate many solutions $\mathbf{x}$ with random positions in the search space.
- Initialize the crossover probability CR and differential weight F
- Until a halting criterion is not met, the following steps are repeated:
  - For each solution $\mathbf{x}$:
    - Choose other three solutions $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ from the population that are different from $\mathbf{x}$.
    - In order to generate a new solution $\mathbf{y}$, for each position i of the solution:
      - Choose a random index j between 1 and the number of variables.
      - Generate a real valued number r between 0 and 1 with uniform distribution
      - If $(r < CR \vee j == i)$, $\mathbf{y}_i = \mathbf{a}_i + F(\mathbf{b}_i - \mathbf{c}_i)$, else $\mathbf{y}_i = \mathbf{x}_i$
    - If $\mathbf{y}$ is better than $\mathbf{x}$, $\mathbf{x}$ is replaced by $\mathbf{y}$
- Return the best solution found

The values F, CR and the population size must be chosen by the user. Given a suitable formulation of a data classification problem, DE can then evolve classification rules for the test database.

3.1 Modeling a DE Solution

As mentioned in Section 2, the classification task involves a model generation. A common approach for classifiers is building binary classification trees, in which, each internal node indicates a test over one attribute, each branch indicates a possible result of the test and each leaf node corresponds to a class. In this way, the tests associated with each node are equivalent to axis-parallel hyperplanes in the input space [4]. For an example on the evolution of these ODTs, see [6].

In this work, we use a different approach for defining each tree node. In this approach the internal nodes test a linear combination of the attributes. With the equation $\mathbf{wx} + \theta = z$, where $\mathbf{w}$ as weight scalars, $\mathbf{x}$ are the data attributes and θ is a constant, we can divide the search space by hyperplanes in general positions, which are not necessarily axis-parallel. With this approach, we define a PDT.

Thus, the search space can be divided considering all data attributes at each step, differently from the conventional binary tree. In case we want to consider only the attribute i with the PDT, the vector $\mathbf{w}$ must be constituted of zeros, apart from the position $\mathbf{w}_i$, where its value will be 1.

Many authors have employed similar concepts of aggregating many linear classifiers in a tree under different names [2,7,8], including PDT [4].

4 Methodology

In this section we describe how DE was used for evolving PDT and the whole procedure used in the comparisons with other well known classifiers to validate our approach.

4.1 Representation of a Classifier

In our method, each PDT is a complete binary tree with linear classifiers in the internal nodes. Each of those nodes is defined by a vector of weights $\mathbf{w}$ of size n and a constant θ, being n the number of attributes of the classification problem. Hence, the number of classifiers in a PDT is $2^{(h-1)}-1$, being h a control variable defined as the depth of the PDT.

Besides the classifiers, each PDT contains $2^{(h-1)}$ leaf nodes, which contain a possible classification for a sample. Those nodes are defined in the discrete space. Thus, a complete PDT can be defined by the matrices of size $n \times 2^{h-1}-2$ for the weights, $1 \times 2^{h-1}-1$ for the constants and $1 \times 2^{h-1}$ for the leaf nodes. This is the size of each solution employed by the DE.

While the individual size is $O(2^{h-1}-1)$, the evaluation cost is still $O(h)$ because only one possible path is searched through the PDT. Each of the $2^{h-1}-1$ columns of the classifier matrices defines a classifier. After using the linear classifier at position i of n, the classifier at position $2i$ is used if the output of the classifier is < 0 or the classifier $2i+1$ is used otherwise.

As the total error will be the standard of comparison between the PDTs in the population, the objective function value of each of the solutions is defined as the number of misclassifications, including false positives and false negatives. This objective function, naturally, must be minimized by the DE.

4.2 Operators

Simple operators, as shown in Section 3 were employed for the generation of new solutions. Values of F and CR are defined as random values between 0.4 and 1 and between 0.9 and 1, respectively [5]. Those values are altered at each iteration of a generation.

As for the leaf nodes, a discrete approach must be used for the crossover of solutions [9]. The approach used is inspired in the idea that DE uses a vector of differences to alter the solution, however now this vector represents swap movements between two possible positions.

In doing so, the movement described as $\mathbf{b}_i - \mathbf{c}_i$ is only performed if $\mathbf{a}_i = \mathbf{b}_i$. If $\mathbf{a}_i = \mathbf{c}_i$, the movement $\mathbf{c}_i - \mathbf{b}_i$ is applied. In a third possible case, where $\mathbf{a}_i \neq \mathbf{b}_i$ and $\mathbf{a}_i \neq \mathbf{c}_i$, no movement is performed.

In this case, the value F is used to define if the operation should happen. The operation occurs if a randomly generated number is less than F.

4.3 Replacement of Individuals

As the replacement of individuals is made one by one, it may happen that some solutions are stagnant in bad points of the search space. In order to avoid this problem, a new individual survival operator was developed to keep always new candidates in the population.

At each generation, new random individuals are generated to replace the $x\%$ worst individuals. Moreover, the individuals with an accuracy rate less than $1/n_{classes}$ are also replaced.

4.4 Legitimacy

In the initial generations, where all the individuals are still very random, not many comparisons are needed to perceive that some are better than the others. In most cases, with the employment of classifiers in less than 10 samples it is possible to define clearly which PDTs are the best ones. The same does not apply after many generations, when most solutions classify the samples with a low error rate.

Having this in mind, and the high cost of evaluating solutions, we propose an approach to reduce the time spent to evaluate solutions in the first generations.

The key idea is that initial solutions do not need to be evaluated with the same legitimacy as the solutions in the last generations, where a more refined analysis is necessary to distinguish good solutions. The value l defines the legitimacy utilized in the evaluation of individuals. At a generation with legitimacy l, only l samples are tested in the evaluation of each solution.

The initial value of this parameter was defined as twice the number of attributes of each samples. After each generation, the value l is updated accordingly to the equation $l = \lceil \min(N, l * \alpha) \rceil$ where N is the number of available samples for test and α is the rate of increase of the legitimacy for each generation.

The parameter α was defined as $1+\delta/\sigma$, where δ is the speed of legimitization, defined as 10^{-6}, and σ is the standard deviation of the objective function value of the solutions. Thus, when the solutions are still very diverse, the parameter α is smaller.

It is important to notice that, whenever the objective function value is calculated, the legitimacy value used to obtain the objective function value is also stored. Thus, if the individual is not replaced by a new one after t generations and the legitimacy value has already increased, the individual is reevaluated with the new legitimacy value l. The value of t was defined as 10.

5 Results

In this section we present the computational results. Our approach was compared with other well known classifiers, such as: J48 [10], BFTree [11], RandomTree [12], IB1 [13], RBF [14], MLPs [15] and Naive Bayes [16]. The results of these methods were obtained from the use of the Weka Framework[1] [12].

For the comparisons, eight databases[2] were taken from the UCI repository [17]. They were randomly divided into two sets, 85% for training and 15% for tests. For fair comparison between the methods, the same sets were used by all of them.

The control parameters of the proposed approach were defined in a series of preliminary tests and they are: Population size = 50 individuals; Maximum

[1] Available in http://www.cs.waikato.ac.nz/ml/weka/

[2] Breast Tissue, Vertebral Column, Ecoli, Glass Identification, Hill-Valey, Iris, Libras Movement and Sonar vs. Rocks.

Table 1. Comparison between PDT and each other techniques

Database	PDT x J48	PDT x BFTree	PDT x RandomTree	PDT X IB1	PDT X RBF	PDT X MLP	PDT X NaiveBayes
Breast T.	worse	worse	worse	worse	worse	worse	worse
V. Column	equal	equal	better	better	worse	worse	better
Ecoli	better	equal	better	better	worse	worse	worse
Glass I.	worse	worse	worse	worse	equal	worse	better
Hill-V.	better	better	better	better	better	better	better
Iris	better	better	better	better	worse	better	better
Libras M.	worse	worse	worse	worse	worse	worse	worse
Sonar vs. R.	better	better	better	worse	better	worse	better

Table 2. PDT Results - Accuracy Rate - Test set

Database	PDT - Min.	PDT - Mean(Std.)	PDT - Max.	J48	BFTree	RandTree	IB1	RBF	MLP	NaiveBayes
Breast T.	0.3750	0.5292 (0.0910)	0.6875	0.625	**0.8125**	0.75	0.6875	**0.8125**	0.6875	**0.8125**
V. Column	0.7660	0.8411 (0.0294)	**0.9149**	0.851	0.851	0.8297	0.7659	0.8723	0.8723	0.8297
Ecoli	0.6863	0.7673 (0.0430)	**0.8627**	0.745	0.7647	0.6666	0.745	0.8235	**0.8627**	**0.8627**
Glass I.	0.4242	0.5808 (0.0690)	0.6667	0.6969	0.7575	**0.7878**	0.7272	0.606	0.6969	0.4848
Hill-V.	0.9341	0.9945 (0.0140)	**1.0**	0.5109	0.5769	0.6318	0.6483	0.5054	0.5769	0.5219
Iris	0.8261	0.9435 (0.0305)	**0.9565**	0.8695	0.8695	0.8695	**0.9565**	**0.9595**	**0.9565**	0.913
Libras M.	0.1852	0.3185 (0.0616)	0.4074	0.7222	0.6111	0.6666	**0.7592**	0.6666	0.7407	0.574
Sonar vs. R.	0.6563	0.7729 (0.0614)	**0.9063**	0.5312	0.625	0.7187	0.875	0.75	0.875	0.6562

Table 3. Error Rate - Training

Database	Minimum Error	Mean Error (Std.)	Maximum Error
Breast Tissue	0.2222	0.2952 (0.0473)	0.4111
Vertebral Column	0.1065	0.1343 (0.0130)	0.1559
Ecoli	0.0982	0.1304 (0.0307)	0.2105
Glass Identification	0.2099	0.3098 (0.0472)	0.4199
Hill-Valley	0.0	0.0054 (0.0133)	0.0602
Iris	0.0	2.6247e-004 (0.0014)	0.0079
Libras Movement	0.5327	0.5874 (0.0289)	0.6438
Sonar vs. Rocks	0.0511	0.0977 (0.0219)	0.1591

number of generations = 2000; PDT height (h) = 5 and Replacement rate = 0.1. An increase in the value of the PDT height would lead to a better capacity of representation as well as an increase in computational cost per generation.

Due to the stochastic nature of our method the results related to it present a synthesis of 30 independent runs. On Table 1 a Kruskal-Wallis test ($\alpha = 0.05$) was used to compare the averaged results obtained by our method with other algorithms. We can see among the tree-based classifiers that the proposed approach was significantly better than the J48 and the RandomTree for most databases. It has also had competitive results in relation to the BFTree and IB1. Among the other types of classifiers, the proposed approach was significantly better than Naive Bayes. The RBF and the MPL performed significantly better than the proposed approach.

Table 2 depicts the results obtained by each one of the methods. For our approach, that will be called PDT for simplicity, we present the worst (Min), mean, and best (Max) results, as well as the standard deviation over 30 runs. The best results of our approach were better in 5 of 8 databases. The other methods had at most the best results in only two databases each. This reveals the potential of the representation model. However, in order to improve the average results, some study and effort must be employed in reducing the variance of the solutions derived by it.

Table 3 shows the error obtained by our method in the training phase. We can observe that in the databases in which our results were poor, the method had difficulties in minimizing the errors also during the training phase, terminating the process with high errors.

6 Conclusion

This work presents an approach to build classifier models based on Perceptron Decisions Trees. For the model training we use an adaptation of the Differential Evolution algorithm which works with continuous and discrete variables. Once the evaluation of solutions has a high cost in the presented procedure, this work also presents the concept of legitimacy, which allows a decrease in the solution evaluation time in the first generations.

The results indicate the potential to build very accurate models as the best solutions found were the best ones in the majority of the tested databases. Despite the success of the best solutions, the statistical test revealed that the average performance of the method is only competitive when compared with other important tree-based classifiers. Besides, the errors obtained in the training phase showed that the method had difficulties in minimizing the error in some databases.

In order to improve the average performance, the use of a validation set can be extracted from the training set and work as test set inside the training. The validation set can broaden the generalization capacity of the classifiers, improving its accuracy and it can be used to give support to both the stopping criteria and the replacement mechanism.

Acknowledgements. This work has been supported by the Brazilian agencies CAPES, CNPq (305506 / 2010-2), FAPEMIG (CEX APQ-04611-10) and by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme.

References

1. Bennett, K.P., Auslender, L., Wu, D., Ave, S.: On support vector decision trees for database marketing. Technical report, Department of Mathematical Sciences Math Report No. 98-100, Rensselaer Polytechnic Institute (1998)
2. Bennett, K., Mangasarian, O.: Multicategory discrimination via linear programming. Optimization Methods and Software 3(1-3), 27–39 (1994)
3. Murthy, S.K., Kasif, S., Salzberg, S.: A system for induction of oblique decision trees. Journal of Artificial Intelligence Research 2, 1–32 (1994)
4. Bennett, K., Cristianini, N., Shawe-Taylor, J., Wu, D.: Enlarging the margins in perceptron decision trees. Machine Learning 41(3), 295–313 (2000)
5. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
6. Cantu-Paz, E., Kamath, C.: Inducing oblique decision trees with evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7(1), 54–68 (2003)
7. Breiman, L.: Classification and regression trees. Chapman & Hall/CRC (1984)
8. Brodley, C., Utgoff, P.: Multivariate decision trees. Machine Learning 19(1), 45–77 (1995)
9. Prado, R.S., Pedrosa Silva, R.C., Guimarães, F.G., Neto, O.M.: Using differential evolution for combinatorial optimization: A general approach. In: SMC, pp. 11–18 (2010)
10. Quinlan, J.R.: C4.5: Programs for Machine Learning, 1st edn. Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann (January 1993)
11. Shi, H.: Best-first decision tree learning. Master's thesis, University of Waikato, Hamilton, NZ (2007)
12. Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I.H., Trigg, L.: Weka. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 1305–1314. Springer, US (2005)
13. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6(1), 37–66 (1991)
14. Buhmann, M.D., Buhmann, M.D.: Radial Basis Functions. Cambridge University Press, New York (2003)
15. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)
16. Zhang, H.: Exploring conditions for the optimality of naïve bayes. International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) 19(2), 183–198 (2005)
17. Frank, A., Asuncion, A.: UCI machine learning repository (2010)

Logistic Regression Applied to Airport Customer Satisfaction Using Hierarchical Quality Model

Teófilo C. Mattozo[1], Gutembergue Soares da Silva[2], André P. Fernandes Neto[3], and José Alfredo F. Costa[2]

[1] Universidade do Estado do Rio Grande do Norte, UERN, Natal, Brazil
mattozo@oi.net.br
[2] Universidade Federal do Rio Grande do Norte, UFRN, Natal, Brazil
{alfredo,guttembergue}@ct.ufrn.br
[3] Universidade Federal Rural do Semi-Árido - UFERSA, Mossoró, Brazil
andrepedro@ufersa.edu.br

Abstract. This study was conducted to evaluate the customer satisfaction as passengers at Augusto Severo International Airport in Parnamirim - RN in order to identify the dimensions which most contribute to the result of assessing the overall passenger satisfaction. It was made a literature review to serve as basis for the research from the model by Brady and Cronin so as to measure the level of satisfaction with the quality of services provided based on their temporal and spatial coverage and related to air transport system, which is the same used as the reference model. Results have shown that the determinant dimensions are significant for the explanation of the overall satisfaction, based on the application of logistic regression. The formulated statistical model was revealed to be well adjusted and able to explain the behavior of passenger satisfaction. The main problems identified were: attitude of staff, concept of facilities and waiting time, in addition to limitations and suggestions for improvements which were approached.

Keywords: Customer Satisfaction, Model by Brady and Cronin, Data Mining, Logistic Regression.

1 Introduction

Airports have been increasingly gaining importance since the plane has been proved an important alternative of transportation, even in the face of crisis that the current domestic airline industry has faced. Due to the importance and growth presented by the sector, it has become necessary to develop studies that promote the optimization of operational procedures at an airport and, consequently, an increase of the quality of its service. It is thus essential to determine the degree of satisfaction with the quality of services provided. Monitoring the customer satisfaction in an industry such as air transport, has increasingly become important in the Brazilian context because of the constant changes within the national civil aviation over the last five years, culminating recently with the granting of the major airports to private enterprises.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 558–567, 2012.

In order to measure and monitor customer satisfaction different methods had being developed and tested. Data Mining (DM) has been used for this purpose, allowing the analysis of complex and difficult-solving problems through innovative approaches. Applications include different areas of management, especially in marketing. DM is recent and powerful new technology with great potential to help companies focus on the most important information in the data they have been collected about the behavior of their customers and potential customers. It enables the discovery of information within the data that queries and reports cannot effectively reveal. Data classification, clustering and visualization are amongst used techniques. The objective is diverse in order to gain knowledge through research of strategic information, standards determination and pattern discovery, classification and association between them [1]. Regression analysis also had been widely used to parameter estimation, data interpolation, etc. The regression algorithms enable discovery of models (equations) that try to best fit on the available data, at a given complexity, that map each entity to a numeric value, instead of searching rules, frames, or trees able to map each entity to a class [2].

2 Goals, Research Problem and Relevance

This research aims to assess the level of passenger satisfaction for the quality of services and identify the attributes determining satisfaction of passengers using the services offered by Augusto Severo International Airport in Parnamirim-RN, by applying the adaptive model by Brady and Cronin [3].

Most terminals in Brazil have registered demand exceeding their capacity, particularly in peak hours, compromising the quality of service. This fact is due to the steady growth of tourist flow in particular, Natal, a city that has tourism as one of its main source of income. Concerning about the quality of services, as well as the lack of capacity of passenger terminals are very relevant and has been studied by several researchers, exploring the development of methodologies that can measure these problems, working with analysis of service level using standard indexes.

Tourism in Brazil has shown positive results in recent years and has been consolidated in the country as an important vector of socioeconomic development. In this scenario, the performance of World Cup Football FIFA in 2014, is a great opportunity for tourism as well as Natal for the image of Brazil abroad. Thus, more than ever it is necessary to a study on the host at any airports so as to make them more pleasant places for people who use them.

Chau and Kao [4] applied the SERVQUAL model to characterize the critical performance factors of airlines and to assess quality models in airports in a survey. It was described differences and expectations between two airports located in very different locations, England and Taiwan. The study concluded that there is a significant difference between the perceived value and the expected quality of service, which is only affected by factors such as education, occupation and income.

Specifically for use at airports, Fodness and Murray [5] developed a conceptual model of service quality, leading to an empirical assessment of the expectations of passenger of services industry. The proposed model has three primary dimensions:

servicescape (covering sub dimensions: layout, environmental conditions and signaling), service personnel (including sub dimensions attitude. Behavior and expertise) and services (including the sub dimensions: productivity, maintenance and integrity). Results suggest that expectations of service quality at airports for passengers are structured in three dimensions with their respective primary sub dimensions: function (efficiency, effectiveness), interaction and entertainment (productivity, maintenance and decoration).

Brady and Cronin [3] presented a new measure for the Perceived Quality of Service, using a hierarchical model. Initially, the authors performed a qualitative and quantitative study, aimed at eight different services in order to identify the sub dimensions and items that consumers associated with the higher-order components: Relationship Quality, Physical Environmental Quality and Provided Service Quality. Survey respondents were asked to list all the variables that influenced their perceptions on each of these three components, which were subject to further refinement and categorization.

The concept of the model is presented in Figure 1, which contains a component of measurement and a structural component. Each of the primary dimensions (Relationship Quality, Physical Environmental Quality and Provided Service Quality) have respectively three subdimensions on which customers make their evaluations to obtain their perceptions on the performance of organizations in each primary dimensions which will lead to the perception of overall quality of service. To evaluate each of the nine sub dimensions are placed three questions of different nature: reliability, comprehensibility and empathy.

3 Methodology

3.1 Sampling and Questionnaire Design

The initial part of the questionnaire used contains information on the descriptive variables such as, sex, marital status, age, education level, monthly household income average, job level in the company they work, number of trips they make per year, reasons why they travel, country and city where they live. At the end, the variables concerning to the content of the research related to the model adapted by Brady and Cronin were exposed.

The population of the study consisted of passengers who used the services offered by the Augusto Severo International Airport in Parnamirim-RN from January to February 2011 (high season) and the interviews were conducted in the halls of departures and arrivals of domestic and international flights. The process used to obtain the samples for the research on user's satisfaction was structured through a sampling plan with a trusting interval of 95% and maximum error of five percent (5%), reflecting the greatest variance in variables, i.e., the worst case in terms of variance [6]. The sample size calculation was performed based on the estimated proportion of the population. To achieve the reliable level and tolerable error margin chosen, the sample (385 questionnaires) was determined using the procedure described by Larson [7].

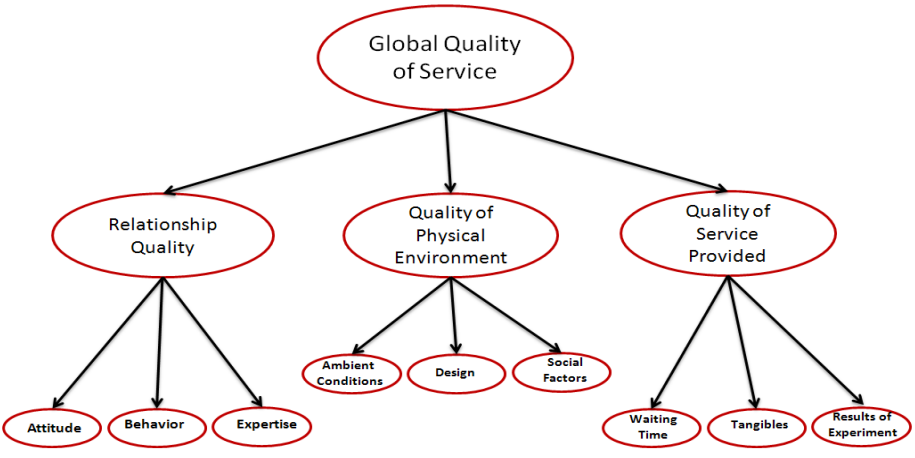


Fig. 1. Formulated Model
Source: Adapted by authors from the model by Brady and Cronin

The choice of data mining techniques will depend on the specific task to be performed and data available for analysis. Harrison [8] suggests that the selection of data mining techniques should be divided into two steps: 1) translating the business problem to be solved in a series of tasks of data mining, 2) realizing the nature of data available in terms of content and types of data fields and structure of relationships between records. This choice can also be based on criteria for classification techniques. A ratio of these types of criteria is given by Harrison [8].

Logistic regression is widely used to model the outcomes of a categorical dependent variable. For categorical variables it is inappropriate to use linear regression because the response values are not measured on a ratio scale and the error terms are not normally distributed. In addition, the linear regression model can generate as predicted values any real number ranging from negative to positive infinity, whereas a categorical variable can only take on a limited number of discrete values within a specified range [9].

Linear discriminant analysis and logistic regression are widely used multivariate statistical methods for analysis of data with categorical outcome variables. Both of them are appropriate for the development of linear classification models, i.e. models associated with linear boundaries between the groups. Nevertheless, the two methods differ in their basic idea. While logistic regression makes no assumptions on the distribution of the explanatory data, linear discriminant analysis has been developed for normally distributed explanatory variables. Whenever the explanatory variables are not normally distributed, the usage of linear discriminant analysis is theoretically wrong, as the assumptions are violated. The goodness-of-fit is therefore only more or less coincidental. On the other hand, the logistic regression fits well many types of distribution. When comparing the robustness towards categorization, linear discriminant analysis remains the favorite method if the number of categories is big enough to let the estimated mean and variance be close to the population values of the continuous explanatory variables. Usually, five categories are enough, but in the case of two or three categories, the advantages of the logistic regression prevail [10].

For these reasons, the study on the general satisfaction of the passengers was made based on the logistic regression model, being possible to determine specific characteristics of different groups of passengers are segmented according to the satisfaction they have. The logistic regression model has characteristics Menard that make it suitable as compared to linear regression [11]. As an example, and taking into account that the vast majority of studies of satisfaction using a Likert scale where they know the meaning of the distances between the scores, for example, between "agree" and "totally agree", it is more correct to use a model that considers only the ratings. Nevertheless, the model in question allows the targeting of the surveyed points for the cutting that allows an additional analysis which would not be possible if the regression model were used. [12].

Based on the notation proposed by Hosmer and Lemeshow [13], the logistic regression model used in this study, also known as binary logistic model is given by equation (1):

$$\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{(\beta_0+\beta_1 x)}} \quad (1)$$

where "π(x)" represents the probability associated with x, "e" is the vector of coefficients to be estimated, characterized as a fixed base of natural logarithms, and "βo+β1" are the vectors of explanatory variables associated with event. In logistic regression, because of its dichotomous nature, the regression coefficients are estimated by applying the maximum likelihood method that generates a combination of factors which maximize the probability of the sample. The transformation is defined in terms of "π(x)" as depicted in equation (2).

$$g(x) = \left[\frac{\pi(x)}{1-\pi(x)}\right] = \beta_{0+}\, \beta_1 x \ldots \infty \quad (2)$$

where [1-π(x)] represents the probability of the event does not occur, [(π(x)) / (1-π(x))] represents the ratio of probabilities, "βi" represents the estimated coefficients and x represents independent variables [14].

In order to assess the quality of the logistic regression model applicable also to measure the degree of association and the total variation of the dependent variable and independent variables. In logistic regression, measures were used to certify the adequacy of the model.

Among these measures, called pseudo R^2, one can highlight the statistics of Cox & Snell R^2 and Nagelkerk R^2 [15], where:

- Cox & Snell is calculated as:

$$R^2_{CS} = 1 - e^{\frac{2\{LL_C - LL_O\}}{n}} \quad (3)$$

Where this statistics never attains the value 1, even when the adjustment is perfect;

- Nagelkerke whose statistics:

$$R^2_N = \frac{R^2_{CS}}{1 - e^{\frac{2LL_O}{n}}} \quad (4)$$

and this statistics was prepared in order to address the main drawback of statistics above, and thus the values may vary from [0,1].

In logistic regression model, the equations are not linear in the parameters and thus require the use of an iterative procedure known as Newton-Raphson method. Briefly, this method can be described as follows. The first step requires the use of an initial solution (candidate) to the values that maximize the likelihood function. The function is approximated in a neighborhood of the initial solution by a second degree polynomial. The second solution obtained in the iterative process is the point of maximum value of the polynomial, and so on. Therefore, the Newton-Raphson method generates a sequence of solutions that converge to the point of maximum likelihood function [16].

3.2 Data Analysis

Statistical analysis was performed using the Statistical Package for Social Science [17], from which we used a logistic regression model that is part of multivariate techniques to describe the relationship between a variable dichotomous response which represents two possible responses (success or failure), assigning a value 1 to the event of interest and value zero to the additional event, setting the probability of an individual belonging to one of two categories (dependent variable) from the other variables (determinant), different from the linear regression model that has responses with continuous variables [18].

The logistic regression base is established to assess the overall adjustment in the comparison between two models. One that considers only the intercept, which is called "constant" by SPSS (in this context, it is reported how well the dependent variable can be predicted without the use of independent variables) and another that takes all the independent variables into account. When comparing the models, it is possible to assess whether those variables contribute to explain the general satisfaction.

The logistic regression model was estimated using the method Forward Stepwise (Likelihood Ratio). This sequential search method is used to estimate the regression of equation to a set of variables, selectively adding or deleting variables until a measure of general criterion is reached, i.e. the process is repeated until the solution represents the best solution with adjustment [19]. As new variables in the model are entered, the goodness of the adjustment improves, since the 2-log likelihood statistics displays successive reductions. Since there is decisive conclusion that contributes to explaining the overall satisfaction, is carried out to verify the contribution of each factor by the value of Nagelkerk R^2.

As determinants of overall satisfaction, three dimensions were considered to reflect the aspects of Relationship Quality, Quality Physical Environment and Quality of Service Provided (Figure 1). In this paper, the dependent variable is the overall satisfaction. The opinion of the passengers was classified into two groups: very satisfied (answering the question on overall satisfaction equal to or greater than 6), moderately satisfied (overall satisfaction between 3 and 5). As there was no response with less than 3 for overall satisfaction, the binary logistic regression was applied.

To measure the scale reliability, Cronbach's alpha coefficient was used (α), which measures the homogeneity of the components of the scale. In general, it is considered a suitable survey instrument to obtain $\alpha \geq 0.70$ [8].

4 Results

Considering the dimensions of the model made and their respective items shown in Figure 1 and the information of Table 1, it can be concluded: The dimension of the Quality of Service (QS) 5 had a mean value on a scale 1-7, corresponding to a good acceptance. Except for the attitude of the staff, all other had appropriate qualification of passengers. On the other hand, the Quality of Physical Environment (QPE) get an average of five points, in which only the design of the facilities was lower than the average of corresponding dimension. The size Quality of Service Provided (QSP) had a mean value 4, indicating only a moderate satisfaction of passengers. The cause of this problem was evident through the averages of the item waiting time.

Table 1. Comparison of dimensions of the model made

Dimension	Number of Items	Note from the General Satisfaction	Cronbach's alpha	Items with low assessment of satisfaction	
				Items	Notes
Relationship Quality	**3**	**5**	0,727	Attitude	4
Quality of Physical Environment	**3**	**5**	0,732	Design	4
Quality of Service Provided	**3**	**4**	0,784	Waiting Time	3

Source: prepared by authors

The characteristics of the model are presented in Table 2. The analysis of log-likelihood of the initial model and the final one has shown a decrease of the final value to indicate that the inclusion of these results in a smaller number of unexplained observations. This result shows that adding the dimensions, it also improves the reliability of the model in classifying the passengers correctly [20]. Otherwise, the model that contains the dimensions can better predict which group belongs to the passenger (fared moderately satisfied or very satisfied).

Classification accuracy reflects the ability of the model to correctly predict the global status category which belongs to a passenger on the basis of size. In the initial model, the classification accuracy is equal to the percentage of individuals in the group with highest incidence (in this study, passenger moderately satisfied). By adding dimensions to the model, we can then assess the extent to which the classification accuracy increases, allowing this analysis to better understand the performance of the regression model [21]. In this model, the addition of dimensions increased the classification accuracy, establishing the percentage of cases correctly classified over 88%.

Table 2. Characteristics of the model

Adequacy of Measures	**Value**	**Df**	**Sig**
-2 Log likelihood ($\chi 2$) initial	227,105		
-2 Log likelihood ($\chi 2$) final	100,812		
Nagelkerke R^2	0,69		
Group of initial classification	74,5%		
Moderately Satisfied	100,0%		
Very satisfied	0,0%		
Group of final classification	88,5%		
Moderately Satisfied	97,3%		
Very satisfied	62,7%		
Test coefficient model ($\chi 2$)	126,3%	2	0,000
Test goodness-of-fit Hosmer and Lemeshow ($\chi 2$)	10,8%	6	0,541

Source: prepared by authors

The measure by Nagelkerke, which acts as a pseudo R^2, also contributes to ascertain how much dimensions should be added for improving the model [22] are commonly used as an approximate measure of variance explained by the dependent variable dimensions [23]. The value by Nagelkerke suggests that the model can explain a good proportion of the variance in overall satisfaction. The three dimensions explain nearly 70% of the variance in overall satisfaction.

Table 3. Determinants of satisfaction

		B	**S.E.**	**Wald**	**df**	**Sig.**	**Exp(B)**
Step 3	QRE	2,519	,680	13,719	1	,000	12,414
	QAF	1,175	,543	4,679	1	,031	3,238
	QSF	1,246	,453	7,570	1	,006	3,476
	Constant	-25,761	5,622	20,997	1	,000	,000

Source: adapted from SPSS report

Table 3 shows the variables in the model as well as their coefficients, and the test results of dimensional significance. Column "B (coefficients)" presents the model coefficients for each dimension, which were obtained in the third processing step of the sequential scan.

Column "SE (Standard Error)" presents a measure of variability of these coefficients. The logistic regression coefficients of these dimensions have statistical significance, as indicated by the Wald test (analogous to multiple regression test). The column "Exp (B)" is the exponential of the estimated coefficients for each variable of the model and indicates the chance of a passenger's condition migrate

moderately satisfied to very satisfied, if a validated dimensions of the model increases to one unit.

For example, considering the size of Relationship Quality that obtained an Exp (B) of 12.414 indicates that an increase of one unit in satisfaction with this dimension increases to 12.414 times the probability of a person moving from moderately satisfied to very satisfied in the general evaluation.

The results suggest that all dimensions of the theoretical model adjusted, notably in descending order of importance: Relationship Quality, Provided Service Quality, and Physical Environmental Quality are statistically significant in explaining the overall satisfaction of passengers.

5 Conclusions

This study revealed how data mining techniques can help managers in understanding how their customers evaluate the quality of service experiences. Results suggest the convenient applicability of hierarchical model by Brady and Cronin and indicated that to increase the satisfaction level of the passenger it will demand improvements in the performance of all dimensions, especially quality of the relationship (the largest coefficient in the regression equation and that represents the greatest influence on the composition of satisfaction). The final index obtained from the General Satisfaction worth five points showed that passengers are not completely satisfied with the services offered at the airport terminals surveyed. The main identified problems were: staff attitude, concept of facilities and waiting time.

Results showed that the dimensions are significant determinants in the explanation of the overall satisfaction, based on the application of the regression method as the data modelling. The statistical model formulated proved to be well adjusted and able to explain the behavior of passenger satisfaction. Furthermore, the high degree of reliability of the scale has shown its replication possible.

Research efforts can be made in order to take into account the passengers in transit in order to evaluate the different services that this process offers and its particularities. By checking if a very long period of time between flights would take passengers to observe, further, the quality of facilities and services that were used, because this airport is the termination point. An analysis of the differences between passenger satisfaction and perception of domestic and international flights could also bring interesting results regarding the level of satisfaction of passengers. Future studies may increase the sample size and thus obtain sufficient questionnaires to analyze the determinants of satisfaction of the passengers of other nationalities.

References

1. Goebel, M., Gruenwald, L.: A survey of data mining and knowledge discovery software tools. ACM SIGKDD 1(1), 20–33 (1999)
2. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery: An Overview. In: Advances in Knowledge Discovery and Data Mining, vol. 611, pp. 11–34. AAAI Press, Menlo Park (1996)

3. Brady, M.K., Cronin, J.J.: Some New Thoughts on Conceptualizing Perceived Service Quality: A Hierarchical Approach. Journal of Marketing 65, 34–49 (2001)
4. Chau, V.S., Kao, Y.: Bridge over troubled water or long and winding road? Gap-5 in airline service quality performance measures. Managing Service Quality 19(1), 106–134 (2009)
5. Fodness, D., Murray, B.: Passengers' expectations of airport service quality. Journal of Services Marketing 21, 492–506 (2007)
6. Malhotra, N.K.: Pesquisa de Marketing: uma orientação aplicada, 3rd edn. Bookman, Porto Alegre (2001)
7. Larson, R., Faber, B.: Estatística Aplicada, 4th edn. Pearson, São Paulo (2010)
8. Harrison, T.H.: Intranet Data Warehouse. Berkeley Brasil, São Paulo (1998)
9. Hair, J.F., Black, W.C., Babin, B.J., Anderson, R.E.: Análise Multivariada de Dados, 6th edn. Bookman, São Paulo (2009)
10. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice Hall, New Jersey (2002)
11. Menard, S.: Applied Logistic Regression Analysis. Sage University Papers Series on Quantitative Applications in the Social Sciences, 07–106. Sage, Thousand Oaks (2002)
12. Mattozo, T.C., Silva, G.S., Neto, A.P.F., Costa, J.A.F.: Application of adaptive model Fodness and Murry the evaluation of passenger satisfaction in Augusto Severo international airport using multivariate regression. In: 9th CONTECSI International Conference on Information System and Tecnology, Brazil (to appear, 2012)
13. Hosmer, D., Lemeshow, S.W.: Applied Logistic Regression. Wiley, New York (2000)
14. Walter, S.A., Tontini, G., Gomes, L., Silva, W.V., Frega, J.R.: Student Loyalty: a logistic regression model. Revista de Administração FACES Jounal 10(4), 129–151 (2010)
15. Larose, D.T.: Data Mining Methods and Models. John Wiley & Sons, Inc. (2006)
16. Pampel, F.C.: Logistic Regression - A primer. Sage University Papers Series on Quantitative Applications in the Social Sciences, 07–132. Sage, Thousand Oaks (2000)
17. SPSS, Inc. SPSS Versions 17.0 for Windows. SPSS Inc., Chicago (2009)
18. Fávero, L.P., Belfiore, P., Silva, F.L., Chan, B.L.: Análise de Dados: modelagem multivariada para tomada de decisão, 1st edn. Campus, Rio de Janeiro (2009)
19. Norusis, M.: SPSS 13.0 Guide to Data Analysis. Prentice Hall, Upper Saddle-River (2004)
20. Tabachnick, B.G., Fidell, L.S.: Using multivariate statistics. Allyn & Bacon, Needham Heights (2001)
21. Moital, M., Dias, R.: Determinantes da Satisfação do Turista de Golfe em Lisboa: uma Comparação entre Britânicos e Nórdicos. In: III Congresso Internacional de Turismo de Leiria e Oeste, Portugal (2009), http://cassiopeia.esel.ipleiria.pt/esel_eventos/files/3903_16_MiguelMoital_4bf56a9848710.pdf (acess. in November 16, 2011)
22. Kleinbaum, D.G.: Logistic Regression: A self-Learning Text. Nova Iorque, 282 p. Springer (1994)
23. Field, A.: Discovering statistics. Using SPSS for Windows: Advanced techniques for beginners. Sage, Londres (2000)

A Transitional View of Immune Inspired Techniques for Anomaly Detection

Guilherme Costa Silva[1], Reinaldo M. Palhares[2], and Walmir M. Caminhas[2]

[1] Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Av. Antônio Carlos, 6627 31270-901, Belo Horizonte, Minas Gerais, Brazil
guicosta@ufmg.br

[2] Department of Electronics Engineering, Federal University of Minas Gerais, Av. Antônio Carlos, 6627 31270-901, Belo Horizonte, Minas Gerais, Brazil
{palhares,caminhas}@cpdee.ufmg.br

Abstract. The use of Immune Inspired approaches for anomaly detection have been adopted in the literature because of its analogy with body resistance in the human immune system provided against agents which causes diseases. There are many models in biology that attempt to explain the immune system behavior, as well some engineering systems inspired on these models. Our goal is to document the development of these models in a transitional view, some aspects which may be considered on these algorithms and on their applicability in engineering problems, with some examples.

Keywords: Immune Inspired Approaches, Anomaly Detection, Immunological Models.

1 Introduction

Immune Inspired Systems are approaches that can be applied to many engineering problems, with a diverse collection of works in literature corresponding to models or theories that attempt to explain their functions. Each model can be applied to a particular problem, as described in [5].

In [4] are shown some imune-inspired techniques and their applications, always based on immunological models and each model has features that represent an application. Thus, each model has applications defined by the features and functionalities present in the human immune system.

After researches on [7], imune inspired algorithms development have grown in recent decades, with a framework that provide bases for new systems [6], or researches about developing more appropriate algorithms for some problems [2].

These studies, according to [20] may involve biology, mathematics and engineering, contributing mutually in order to encourage both immunologists to research about a proper functioning of the immune system, and engineers to develop new systems capable of solving problems with more reliable results.

In Anomaly Detection, the applicability of existing immune theories on the systems may be seen on literature, since the Self/Nonself Discrimination [7,15,3]

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 568–577, 2012.

until Danger Model based algorithms in [12,24,19,23]. It is noteworthy that these inspirations are not mutually exclusive,

This paper will present a review and comparative studies on imune-inspired anomaly detection, establishing a transitional link between Self/Nonself Discrimination Theory and the Danger Model, showing out the features, advantages and disadvantages of each approach, with an example used for a demonstration purpose.

2 Transition among Immune Inspired Aproaches

The models studied in this work follow the description of [10], describing the evolution of immunological theories, along with the corresponding imune inspired approaches. These models are based on signal presences required for immune response, as defined in Figure 1.

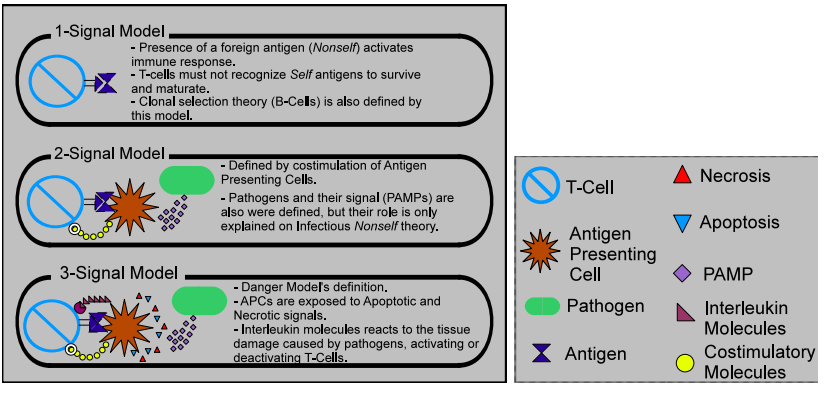


Fig. 1. Description of immunological models discussed in this paper

In the following subsections, the immunological models and their respective algorithms will be discussed. Their characteristics and applicable situations, as well as the advantages and disadvantages of these algorithms will be exposed in the research.

2.1 One Signal - Self-Nonself Model

The model is based on the elimination of immature cells of the immune system which recognize Self patterns. The first immune inspired algorithms were developed from this principle.

The Negative Selection Algorithm (NSA), defined in [7], is an anomaly detection system that consists of analyzing the feature space, and through it, generating detectors in the Nonself region. The algorithm has resemblance to supervised machine learning methods, since the Self data is used as a reference, so that the detectors are located outside of Self region.

The method in fact, can be summarized as a semi-supervised classification problem, in which only one of the patterns (Self) is known and the outliers may belong to another class (Nonself), as shown in Figure 2.

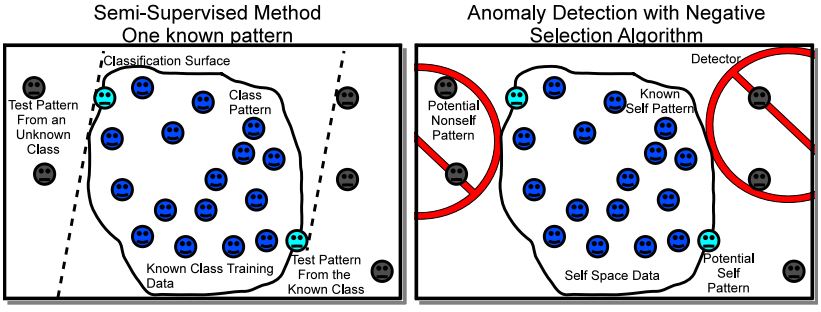


Fig. 2. Illustration of similarities between semi-supervised classification and anomaly detection based on *Self/Nonself* Theory

Many approaches appeared trying to improve operational aspects such as optimization of the coverage area to detectors [15], allocation considering boundaries of Self space [14], or overlap on two or more detectors [18]. In [8], is considered a training method that optimizes the computational cost of the algorithm. Other improvements are considered in [17].

The algorithm is very intuitive and quite simple but has many issues: to allocate the detectors and measure the similarity between these and the data may imply something quite costly and redundant, especially in high-dimensional problems. Furthermore, the algorithm has serious problems concerning the system context. Other issues can be seen in the analysis of [16].

Despite the issues, the NSA is applicable to problems where there are few abstractions on the application and it is possible to set normal behavior. However, the algorithm is very limited considering the application environment.

2.2 Two Signals - Costimulatory Model and Infectious Nonself

The costimulatory model defines two signals required for activation of immune response: the nonself antigen presence and a signal emitted by Antigen-Presenting Cells. This model has inspired few researches in computer networks, such as [13,1]. These models can be considered intermediate or transitional between the Negative Selection and the Danger Model.

The Toll-Like Receptor (TLR) Algorithm [21] is also based in this model, inspired by the of infectious nonself theory. This algorithm was first developed to solve problems of intrusion detection on a FTP server.

The algorithm is based upon T-cell and APC training as the signals outside the normal range are captured by APCs, causing maturation for these cells, otherwise they become semimature. For the T-cell activation, this cell have to recognize antigens collected by a mature APC. This model is summarized in Figure 3.

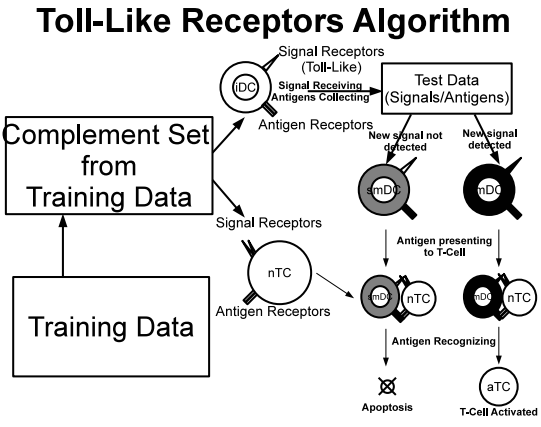


Fig. 3. Summary of Toll-Like Receptor Algorithm

This model represents a "half-way" between the classical immunology, and the next model combining known data sampling and some priori information about the application context.

2.3 Three Signals - Danger Model

The Danger Model defines the immune system which reacts to signals emitted by cells in the body, which may indicate a situation of damage (necrosis) or a situation of natural death (apoptosis) and APCs may emit molecular signals to T-cells that indicates the body context. Figure 4 essentially summarizes the immune response according to the model.

The algorithms inspired on this model consider indicative information about the context of application environment [2], even requiring a more advanced level of abstraction for the analogy with the immune model involved.

Dendritic Cell Algorithm (DCA) [12] is the main representative of the the Danger Model based approaches. The correlation mechanism of the algorithm determines whether a process involved in the system (antigen) has relation with the behavior of the system (Input Signal) according to the illustration in Figure 5. The input signals are divided into three categories:

1. PAMP - signal that determines the evidence of an anomaly.
2. Necrotic or Danger Signal (DS) - High values determine a possible indication of abnormality. Compared to the PAMP signal, is a signal whose presence of abnormality is less certain. Both signals turns the cell mature.
3. Apoptotic or Safe signal (SS) - indicates that the system is operating normally, can suppress immune response and turns the cell semimature.

In DCA, the Inflammatory Signal is also defined, which amplifies the effect of other signals. For reasons of problem modeling this signal is not often used in

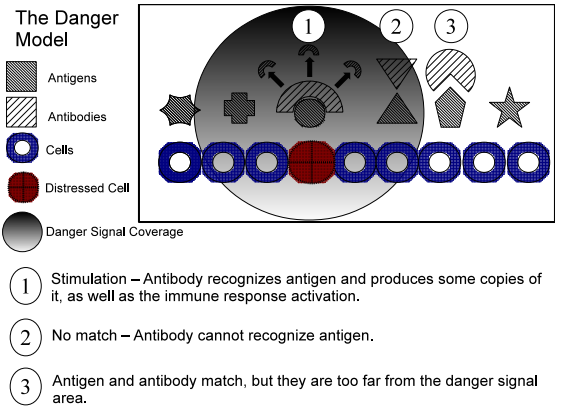


Fig. 4. The antigen recognition in a Danger Model view

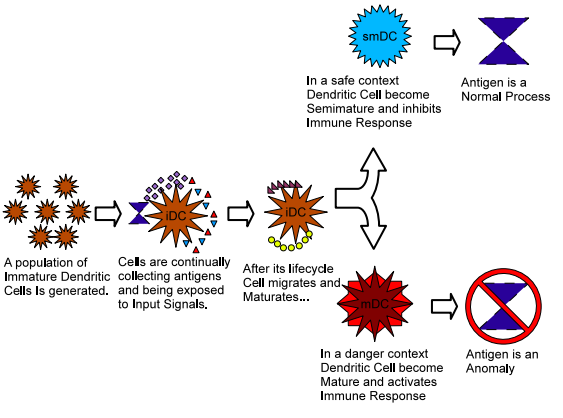


Fig. 5. Summary of abstraction of the Dendritic Cell Algorithm

practice. In addition, a DCA basic implementation requires DS and SS as defined in [9].

The algorithm eliminates the need to define a normal pattern for system, however, requires a specific knowledge about the problem or application, so that the input signals and even the antigen are properly shaped. Furthermore, a data pre-processing is very important.

In [19] is defined another way to represent danger model signals: through fuzzy rules processing, boundaries between the analyzed signals are set. As different from DCA, requires no representation of antigen.

Analyzing the algorithms based on the danger model, the great advantage is the proper consideration of further information on the application environment and more contextual view of signals or antigens. However, there are some

limitations in the approach as the considered in [22]. Furthermore, many systems do not have a model specified by experts to assess any anomalies. Often, these models must be discovered.

2.4 Comparative View of Models and Algorithms

Based on immunological models presented in this paper, immune inspired systems have features whose the application environment must be considered on the implementation.

In order to briefly describe each approach, a comparison between models and algorithms inspired on them was made in Table 1, considering the work proposed in the literature and as described in this article.

To check the survey in this research the two main immune inspired algorithms will be applied to a recurring anomaly detection database.

3 Example: UCI Breast Cancer Database

This test was originally proposed in [11] to validate DCA experiments and to demonstrate some of the algorithm issues.

Likewise, the objective of these tests is to compare test results between imune-inspired models and demystify the superiority idea among the approaches, considering that there are cases in which the negative selection algorithm may overcome the dendritic cells algorithm, or vice versa, for example.

The Two Signals Model was not represented in these tests, because there are few studies focused on these algorithms.

For DCA, the tests were redone considering the same conditions of [11] and ordering the data set, however, the algorithm used was the deterministic version proposed in [9] and the output signals in accordance with equations (1) and (2) with a population of 100 cells which can store up to 50 antigens and lifetime threshold 30.

$$CSM = DS + SS \tag{1}$$

$$K = mat - semi = PAMP + DS - 2SS \tag{2}$$

For the NSA, four approaches were used: the classical algorithm, the V-detector [15], the Multioperational algorithm [18] and a monitoting algorithm based on Fuzzy Antigenic Recognition approach [3]. The tests were performed considering 25% of the normal data used for training, with self radius $r_s = 0.1$ and number of detectors $D = 250$ for the first three algorithms. For the latter algorithm, two thresholds values 0.1 and 0.2 were adopted, whose complement determines the threshold of the fuzzy inference system.

Furthermore, for these algorithms, the use of a feature selection mechanism was considered in order to reduce computational costs and to improve the results. The tool used is the Information Gain, which is based on statistical calculations considering the information processed relevance. Thus, 3 of 9 features

Table 1. Comparison between models and its immune inspired systems

Model	Self/Nonself	Costimulatory	Danger
Theory	Discriminatory model related to antigen patterns	Similar to previous model, but requires an stimulation signal	System is reactive to damage occurred on tissue cells
Antigen	Patterns which have to be recognised as Normal or Anomaly	Patterns recognition under a costimulatory signal presence	Patterns which may be identified as a potential anomaly
Signals	None, this model only deals with antigen recognition	Costimulatory or, in extended model, PAMPs	PAMPs, Apoptotic and Necrotic signals.
T-Cells	Detectors allocated in Nonself space	Agents which interact to APCs	Abstraction for the immune response
APCs	None	Dendritic Cell - emit required signal to the immune response	Dendritic Cell - Surpress or activate immune response
Basis	Supervised methods and recognition	Hybrid approach between models	Expert knowledge about the environment
Training	Required	Usually required	Optional
Priori Knowledge	Training data defines Self space	Training data used for signal tuning	Modeled after evidentiary data
Normal Data	Very important	Important	Inferred by signals
Pre processing	Data may be normalised, ensuring system performance	Important on each algorithm steps	An important factor for smooth operation on input signals
Developing	Quick and simple, depending on the algorithm version	May become complex due to the application environment	Algorithms are well developed, they rely on input signals modeling
Algorithms Runtime	In some cases, can be very costly, depending on the feature space	Depends on processing components used in these algorithms	The runtime must fit the input signals processing
Anomaly Vision	Unknown patterns	Strange and unusual behavior	Inferences about abnormal behavior
Data Processing	Look for unknown patterns	Look for outlier data.	Correlate data to system behavior
Applied to	Databases in which normal data are fully known	Databases in which is possible to deal with partial knowledge	Data in which it is possible to infer when anomalies occur
Criticism	Problematic definition of patterns, weak contextualization for a system	Requires some setting of the application context	It is not always possible to represent a system consistently
Advantages	A simple and intuitive approach, with several possibilities of tunning or implementation	May represent a suitable alternative in some engineering problems	Representation is more realistic, the analogy suits with many applications
Disadvantages	In some cases, algorithm evaluation can lead to high computational cost	Modeling these algorithms may be an empirically complex task	The representation for some problems may require context adaptations

were considered. Interestingly, the tool has selected the top 3 attributes used for calculation of danger signal in [11]. More about the information gain can be found in [25].

Table 2. Results for proposed test with the some immune inspired algorithms

Algorithm/ Scenario	Training Data	Test Data	True Positive	True Negative	False Positive	False Negative
Classical Negative Selection Algorithm	60	640	79%	23%	77%	21%
V-detector Algorithm	60	640	96%	86%	14%	4%
Multi Operational Algorithm	60	640	94%	90%	6%	8%
Fuzzy Antigen Recognition ($t_s = 0.10$)	60	640	98%	59%	41%	2%
Fuzzy Antigen Recognition ($t_s = 0.20$)	60	640	70%	92%	8%	30%
Dendritic Cell Algorithm (1-step)	-	700	98%	87%	13%	2%
Dendritic Cell Algorithm (2-step)	-	700	75%	46%	54%	25%

The Table 2 results presents the classification performed by the methods related to the models in which they were based. At least one of the formulations of Negative Selection Algorithm had a comparable performance to the dendritic cell algorithm.

The classical NSA had the worst performance, with little distinction between the classes. The Multioperational approach and the Fuzzy Monitoring had reasonable performances with a considerable number of false negative, whereas in the latter, the classification threshold determined the model sensitivity. V-Detector were able to classify data with few false negatives and had a significant performance over the DCA.

The Danger Model based approach has a major drawback in this application: the sensitivity to the instant at which data are positioned, due to the correlation mechanism. As demonstrated in this work, the data order determine the classification. Furthermore, the NSA is still suitable to deal with machine learning-based anomaly detection.

4 Concluding Remarks

This paper estabilished a transition regarding immunological models used to build immune inspired systems applied to anomaly detection by comparing immunological models and systems inspired on them. Based on the presented results, it was also confirmed an idea about these algorithms: There is no better

immune inspired model than another. Further researches focused on these approaches may provide more results in many other applications by exploiting and finding more advantages and disadvantages of immunological models for immune inspired anomaly detection.

Although the algorithms based on negative selection has inherent flaws on context or applications, it is possible to make them applicable in certain cases, especially in learning machine problems, moreover, the addition of other immune mechanisms algorithms can be considered.

Danger Model based algorithms have the disadvantage of a knowledge required to model the signals, which usually is not implicit in the databases. However, this approach can be extended to statistical or qualitative aspects.

Another aspect is the applicability of the costimulatory models, which are still poorly widespread, but they may represent a hybrid approach in relation to the anomaly detection, in order to employ the supervised mechanism and to generate rules-based signals to activate the immune response.

Finally, a closer study of the immune engineering tools can be developed and implemented to improve the results and even generate other similar systems, based on the models cited in this study.

Acknowledgements. The authors received support from funding agencies CNPq and FAPEMIG during the development of this research.

References

1. Balthrop, J.L.: Riot: A responsive system for mitigating computer network epidemics and attacks (July 2005)
2. Cayzer, S., Aickelin, U.: The danger theory and its application to artificial immune systems, University of Kent at Canterbury. pp. 141–148 (2002)
3. Costa Silva, G., Palhares, R.M., Caminhas, W.M.: Immune inspired fault detection and diagnosis: A fuzzy-based approach of the negative selection algorithm and participatory clustering. Expert Systems with Applications (Accepted for publication, 2012)
4. Dasgupta, D., Nīno, F.: Immunological Computation: Theory and Applications. CRC Press (2008)
5. de Castro, L., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
6. de Castro, L.N.: Immune cognition, micro-evolution, and a personal account on immune engineering. Semiotics, Evolution, Energy, and Development Journal 3(3), 134–155 (2003)
7. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, p. 202 (1994)
8. Gong, M., Zhang, J., Ma, J., Jiao, L.: An efficient negative selection algorithm with further training for anomaly detection. Knowledge-Based Systems 30, 185–191 (2012)
9. Greensmith, J., Aickelin, U.: The Deterministic Dendritic Cell Algorithm. In: Bentley, P.J., Lee, D., Jung, S. (eds.) ICARIS 2008. LNCS, vol. 5132, pp. 291–302. Springer, Heidelberg (2008)

10. Greensmith, J., Aickelin, U.: Artificial Dendritic Cells: Multi-Faceted Perspectives. In: Bargiela, A., Pedrycz, W. (eds.) Human-Centric Information Processing Through Granular Modelling. SCI, vol. 182, pp. 375–395. Springer, Heidelberg (2009)
11. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 153–167. Springer, Heidelberg (2005)
12. Greensmith, J., Aickelin, U., Cayzer, S.: Detecting Danger: The Dendritic Cell Algorithm. In: Schuster, A. (ed.) Robust Intelligent Systems, vol. 12, pp. 89–112. Springer, Heidelberg (2008)
13. Hofmeyr, S.A.: An immunological model of distributed detection and its application to computer security. Ph.D. thesis, University of New Mexico (May 1999)
14. Ji, Z.: A Boundary-Aware Negative Selection Algorithm. In: Proceedings of the 9th International Conference on Artificial Intelligence and Soft Computing. ACTA Press (2005)
15. Ji, Z., Dasgupta, D.: Augmented negative selection algorithm with variable-coverage detectors. In: Congress on Evolutionary Computation, CEC 2004, vol. 1, pp. 1081–1088 (2004)
16. Ji, Z., Dasgupta, D.: Applicability issues of the real-valued negative selection algorithms. In: Proceedings of the 8th annual Conference on Genetic and Evolutionary Computation, GECCO 2006, pp. 111–118. ACM, New York (2006)
17. Ji, Z., Dasgupta, D.: Revisiting negative selection algorithms. Evolutionary Computation 15(2), 223–251 (2007)
18. Laurentys, C., Ronacher, G., Palhares, R., Caminhas, W.: Design of an artificial immune system for fault detection: A negative selection approach. Expert Systems with Applications 37(7), 5507–5513 (2010b)
19. Laurentys, C.A., Palhares, R.M., Caminhas, W.M.: Design of an artificial immune system based on danger model for fault detection. Expert Syst. Appl. 37, 5145–5152 (2010a)
20. Timmis, J., Hone, A., Stibor, T., Clark, E.: Theoretical advances in artificial immune systems. Theoretical Computer Science 403(1), 11–32 (2008b)
21. Twycross, J., Aickelin, U., Whitbrook, A.: Detecting anomalous process behaviour using second generation artificial immune systems. International Journal of Unconventional Computing 6, 301–326 (2010)
22. Vella, M., Roper, M., Terzis, S.: Danger Theory and Intrusion Detection: Possibilities and Limitations of the Analogy. In: Hart, E., McEwan, C., Timmis, J., Hone, A. (eds.) ICARIS 2010. LNCS, vol. 6209, pp. 276–289. Springer, Heidelberg (2010)
23. Xu, Q., Wang, S., Zhang, C.: Structural design of the danger model immune algorithm. Information Sciences (2012)
24. Zhang, J., Liang, Y.: A novel intrusion detection model based on danger theory. In: Proceedings of the 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application (2008)
25. Zhang, R., Tran, T.: An information gain-based approach for recommending useful product reviews. Knowledge and Information Systems 26, 419–434 (2011)

An Aspect-Oriented Domain-Specific Language for Modeling Multi-Agent Systems in Social Simulations

Diego de S. Braga[1], Felipe Omena M. Alves[2], Fernando Buarque de L. Neto[1], and Luis Carlos de S. Menezes[1]

[1] University of Pernambuco (UPE), Polytechnic School of Pernambuco (POLI)
Street Benfica, 455, Madalena – 50750-410 – Recife, Pernambuco – Brazil
{dsb,fbln,lcsm}@ecomp.poli.br
[2] Estácio do Recife College
Av. Eng. Abdias de Carvalho, 1678, Madalena – 50720-635 – Recife, Pernambuco – Brasil
felipeomena@gmail.com

Abstract. Aspect-oriented programming (AOP) is a programming paradigm which aims to increase modularity by allowing the separation of cross-cutting concerns. This paper presents the definition and characteristics of the domain-specific language, aspect-oriented, AspectNetLogo and its compiler, the AspectNetLogoCompiler and show the use of this system in a multi-agent system in social simulation. This system allows the definition of the elements of the agents in the NetLogo environment in an isolated way and simplify the implementation of social simulations.

Keywords : AOP, MAS, NetLogo, Social Simulation.

1 Introduction

Social Simulation is an area of research that uses computational methods to solve problems in social sciences (politics economics anthropology etc.) using agents.

Agent Oriented Programming [1] defines the computing from social interactions of entities known as agents. An agent is an entity that perceives changes in the environment and act in response to these changes in order to achieve some predetermined goal. Agents in a Multi-Agent System (MAS) [2] have interesting characteristics such as: autonomy, proactivity, learning, communication and coordination of distributed tasks, among others that enable them to deal with complex problems.

Aiming to simplify the execution of social simulations using agents, several tools such as PAX [3], Repast [4] and NetLogo [5] provide to the programmer a basic infrastructure with the main elements of a multi-agent system. These structures have been successfully used to perform social simulations [6][7][8]. However, the tools available today to perform social simulations of agent-based models require considerable effort for modeling as they still require computer science knowledge from the user to define new types of simulations, because the characteristics of a simulation are mixed in the code. These abilities are not commonly found in social researchers who demand this type of tool.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 578–585, 2012.

The Aspect-Oriented Programming (AOP) [9] is a programming model that is designed to increase the modularity of implementation of cross-cutting concerns in computer systems. A cross-cutting concerns is a property of an application to be implemented, requires modification of a large number of modules, it cannot be isolated using conventional techniques of modularization. The principle of separation of concerns is based on construction systems that have different modules for solving different responsibilities. Thus, it will not be found cross-cutting concerns in the code, possible to change old services offered without other parts of the systems being impacted.

This paper proposes the establishment of a domain specific language (aspect-oriented) to the description of multi-agent systems in Social Simulations.

As example of using aspects in multi-agent simulations of this paper implements an aspect-oriented extension of NetLogo. The NetLogo is a programming language and integrated environment for multi-agent modeling. It is particularly well suited for modeling complex systems developing over time. Modelers can give instructions to hundreds or thousands of "agents" all operating independently. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from their interaction. However, the definition and characteristics of agents in complex problems are frequently produced in different parts of the application, which complicates the understanding and maintenance of the models developed.

The use of aspects in an agent oriented system, in particular NetLogo, will allow the elements that compose an agent to be defined and studied in isolation from the rest of the simulation. This feature can facilitate the definition of new simulations and reuse elements of existing simulations in new contexts.

2 AspectNetLogo

The Aspect-Oriented Programming defines structures known as aspects, permitting the declaration of characteristics in isolation from the rest of the system. At some point of compilation, these settings will affect the code of the multi-agent model, in a process called Aspect Weaving (view item 3.4). That way, a new file is generated containing all specifications performed, not changing the original model, as illustrated in Figure 1. The artifact obtained by the implementation of the basic functionality of an multi-agent system, for example, is joined to the characteristics defined in the AspectNetLogo language (e.g. autonomy, adaptation, interaction, etc.). The result of this combination is the generation of a new model containing these characteristics incorporated into the basic functionalities of the application.

The basic features of multi-agent model for example, declarations and procedures are implemented using the language component NetLogo. For the definition of aspects of the model are used the aspect-oriented language, presented in this work, AspectNetLogo [10]. It supports the implementation of aspects, defining their behaviors and situations that will occur in a clear and concise way.

AspectNetLogo is conceptually very similar to AspectJ [11] and AspectC++ [12]. The reason for this similarity is trying to allow people who learned the aspect-oriented paradigm by using AspectJ or AspectC++ implementation to easily switch over to AspectNetLogo, if they are also already familiar with NetLogo. Some important concepts and definition are:

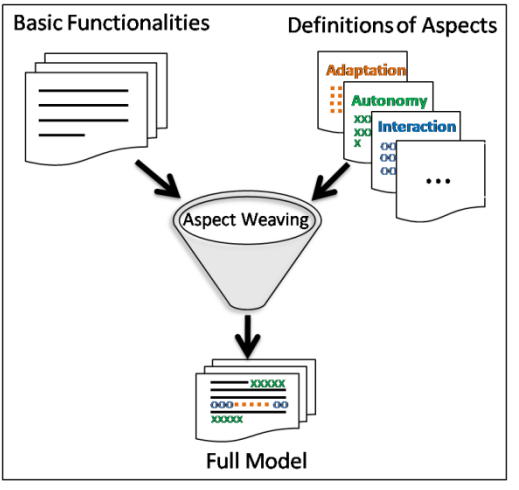


Fig. 1. Illustrates the process of Aspect Weaving

Join Points
The Join Points specify functions in the model, developed in NetLogo, which will suffer some kind of interference created by the aspect. They are defined in the Declaration of Pointcuts.

Pointcuts
Pointcuts are designed to establish rules to specify anywhere in the code of the model, to which to add a new service or obtain data from a particular context, for example.

```
pointcut identifier : join-point logical-operator? designator
```

Advices
The Advices specify the code which will be added at points defined by the Pointcut. It is also essential determine the time of inclusion of the new instruction. This time may be 'before' or 'after' the occurrence of the Pointcut. Beyond these possibilities, there is the 'around' mode that allows changing the code existed previously.

```
time : pointcut-list statement-block
```

Intertypes
Unlike the other components, the Intertypes not need the reference Join Points to identify the locations that the new codes will be added. The intertypes acting adding new statements, whether functions, variables, etc.

```
intertype : procedures ||
intertype : identifier in-block declarations
```

Aspect
The combination of the Pointcut and the Advice is termed an aspect. Aspects can isolate definitions, make the source code more cohesive and provide a better understanding of the model. These isolations can be better understood from Figure 4.

3 AspectNetLogoCompiler

The AspectNetLogoCompiler checks if the encoding used in the imported file, written using AspectNetlogo, is in accordance with the standards defined in this language (ie syntax and semantics) and, from that verification, works in the multi-agent model written in the language of component (NetLogo). This section presents the main elements of the AspectNetLogo compiler.

A file containing the definitions of aspects in AspectNetLogo is read and submitted to the phases of Lexical, Syntax and Semantic Analysis of the compiler. After the Analysis Phase is performed a checking to verify if there is an impact in the code to which the aspects will be applied. Through Weaver (view section 3.4), is realized the joining process in the aspect language code (AspectNetLogo) with the component language code (NetLogo) and generates a file containing the new multi-agent model specifications. The architecture of the compiler is presented in Figure 2.

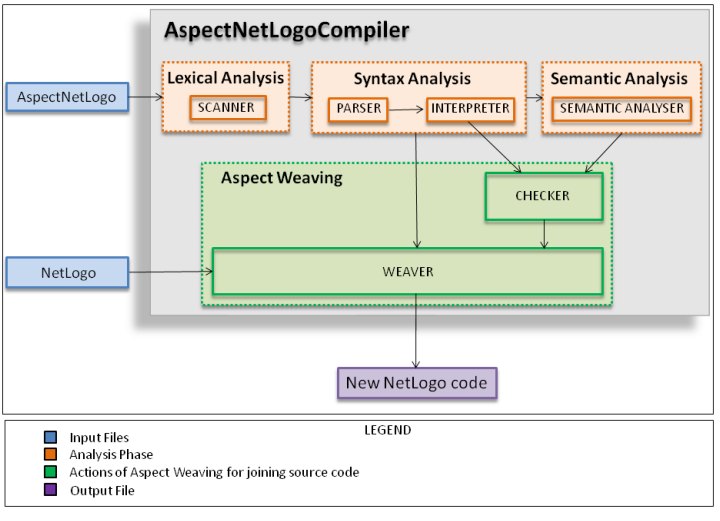


Fig. 2. Architecture of the AspectNetLogoCompiler

3.1 Lexical Analysis

The Lexical Analysis or Scanning is responsible for reading the character stream, from left to right, and group them into tokens that are sequences of characters which have a meaning in collective. Its objective is identify symbols (tokens) used in the source code and check if they are included in the AspectNetLogo language specification. It is also your responsibility to identify the set of ignored tokens defined in the grammar, ignoring its occurrences. This allows characters such as line breaks and comments being ignored.

3.2 Syntax Analysis

The Syntax Analysis (or Parsing) is the process of analyzing a text, made of a sequence of tokens, to determine its grammatical structure with respect to a given formal grammar. It checks for correct syntax and builds a data structure implicit in the input tokens. In AspectNetLogoCompiler the parsing is performed by two components with distinct functions: Parser and Interpreter.

3.3 Semantic Analysis

After the phases of Lexical Analysis (view section 3.1) and Syntax Analysis (view section 3.2), the Semantic Analysis is performed by the component of AspectNetLogoCompiler called Semantic Analyser. This phase checks for semantic errors in the source program and capture the information necessary for the subsequent phase of code generation.

3.4 Aspect Weaving

Aspect Weaving is the name given to the process performed by combining aspects (Aspect Weaver) of the AspectNetLogoCompiler. It is responsible for combining the code developed in the components language (NetLogo) with the code developed in the aspect language (AspectNetLogo). In AspectNetLogoCompiler the process of Aspect Weaving is accomplished by two components: *Checker* and *Weaver*.

This union of code is performed at compile time. First of all is identified which aspects will be applied (through the *Checker*), and subsequently the two implementations are joined to form the final code of the application (through the *Weaver*).

Checker
Checker is the name given to the component responsible for performing the validation of the actuation in the aspect elements found in the NetLogo code. Its function is check if there is impact on the code for each component of the aspect. After this check, the Checker should report at the time of generating the new file, which elements did not pass in this validation, in other words, which elements had no impact on the code generation. E.g., if an intertype act redefining a function and this function does not exist in the code, this element should be reported after the process Weaving because it failed to act.

Weaver
The component Weaver receives the necessary information of the aspects, extracted from the Checker, to perform the correct junction of code. In addition to identifying the Pointcuts in the imported file, the Checker informs the components of aspect which does not affect the code NetLogo, so they are ignored in the weaving process. Weaver concludes his activity, generating a new file containing the services added by the aspect.

4 Case Study in Social Simulation

To validate the use of AspectNetLogo has been chosen a model of social simulation available in the NetLogo´s Models Library called Rebellion [13]. It models the rebellion of a subjugated population against a central authority. It is an adaptation of Joshua Epstein's model of civil violence [14].

The population wanders around randomly. If their level of grievance against the central authority is high enough, and their perception of the risks involved is low enough, they openly rebel. A separate population of police officers ("cops"), acting on behalf of the central authority, seeks to suppress the rebellion. The cops wander around randomly and arrest people who are actively rebelling.

Through this modeling it was identified that a simple change in how individuals make decisions would impact in the need for changes in various parts of the code. For example, to be modified the way in which jail term is defined is required the maintenance in eight functions of the model. Figure 3 (a) illustrates this problem and the definition of an aspect that models the behavior of the agent in the model of the case study.

The use of AspectNetLogo offers the possibility of modeling agents with complete separation of responsibility. The characteristics and behaviors of agents are implemented as isolated aspects, making maintenance simpler and more practical. The proposed solution to the example cited above, is the construction of an aspect that implements the definitions of jail term, and through the Pointcuts set pieces of code in which the aspect will interfere. Thus, future changes would be made only on the aspect constructed. Figure 3 (b) illustrates the definition of an aspect that models the behavior of the agent in the model of the case study. Figure 3 shows the declaration of this aspect in the AspectNetLogo language.

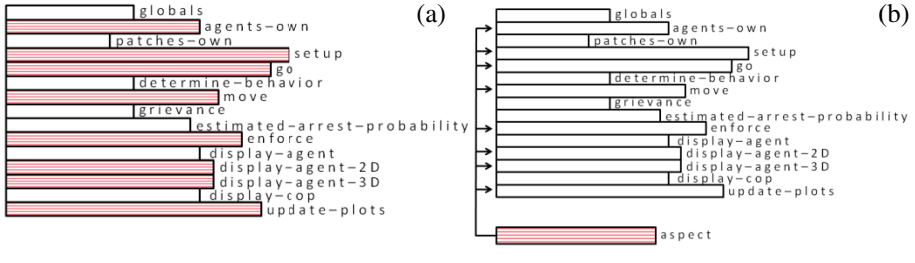


Fig. 3. (a) Each column of the figure represents an existing function in the Rebellion's model. A possible change in the way in which jail term is defined impacts on 53% of the functions (hatched columns). (b) The aspect defines the behaviors of the agent, and through the Pointcuts are defined locations of the code which the aspect will interfere.

Aspect

```
aspect jail-term{
pointcut set-jail-zero : execution setup && target 'display-agent'
pointcut check-jail-breed : execution go && target 'ask turtle'
pointcut decrease-jail : execution go && target 'ask agents [ display-agent ]'
pointcut no-agents-here : execution move && target 'let targets neighborhood with'
pointcut set-jail-random : execution enforce && target 'set active? false'
pointcut check-jail-to-set-color : execution display-agent-2D && target '[ set color red ]'
pointcut check-jail-to-set-shape : execution display-agent-3D && target '[ set shape "person active" ]'
pointcut declare-jailed-count : execution update-plots && target 'set-current-plot "Active agents"'
```

Pointcuts

Advices

```
before : set-jail-zero
        set jail-term 0
before : check-jail-breed [
                if (breed = agents and jail-term = 0) or breed = cops
                        [ move ]
                if breed = agents and jail-term = 0
                        [ determine-behavior ]
                if breed = cops [ enforce ]]
after : decrease-jail
        [ if jail-term > 0 [ set jail-term jail-term - 1 ] ]
after : no-agents-here
        [not any? cops-here and all? agents-here [jail-term > 0]]
after : set-jail-random
        set jail-term random max-jail-term
after : check-jail-to-set-color
        [ ifelse jail-term > 0
        [ set color black + 3 ]
        [ set color scale-color green grievance 1.5 -0.5 ] ]
after : check-jail-to-set-shape
        [ ifelse jail-term > 0
        [ set shape "person jailed" ]
        [ set shape "person quiet" ] ]
before : declare-jailed-count
        let jailed-count count agents with [jail-term > 0]

}
```

Fig. 4. Definition of the jail-term aspect in Rebellion's model using AspectNetLogo language

5 Conclusion and Future Work

This article presented an aspect-oriented language for modeling Multi-Agent Systems in Social Simulations – AspectNetLogo. Its compiler – AspectNetLogoCompiler – was also presented a case study in social simulation using this system. The AspectNetLogo has provided NetLogo the ability to use aspects for modeling Multi-Agent Systems offering the possibility of modeling agents with complete separation of responsibility.

Although it was necessary to make changes in some parts of the syntax and grammar of AspectJ and AspectC++, most of the concepts of these languages were preserved. Therefore it is expected that users with experience in AspectJ (and / or AspectC++) and NetLogo users can become familiar with the AspectNetLogo without great effort.

The use of aspects-oriented techniques combined with NetLogo language is unknown and was not found in the literature any domain-specific language for Social Simulation. Most certainly, this highlights the innovative character of the present work.

As further work, we propose the realization of a detailed study on existing models of social agents, therefore believe this study could help identify new types of aspects for multi-agent systems. Extensions of this work contemplate the construction of a aspect-oriented system for social simulations to describe more friendly simulations making them semi-transparent to the user the details of computational and artificial intelligence techniques used.

References

1. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall (1995)
2. Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley (1999)
3. Neto, L., de Buarque, F., Pita, M.R.S., Filho, B., Serrano, H.: Hybrid and Evolutionary Agent-Based Social Simulations Using the PAX Framework. In: Nineth International Conference on Intelligent Systems Design and Applications, ISDA, Pisa, Italy (2009)
4. North, M., Howe, T., Collier, N., Vos, J.: The repast simphony runtime system. In: Proceedings of the Agent 2005 Conference on Generative Social Processes, Models and Mechanisms (2005)
5. Wilensky, U.: NETLOGO itself: NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999), `http://ccl.northwestern.edu/netlogo/`
6. Pita, M., Neto, L., de Buarque, F., Filho, B., Serrano, H.: Impact of Communication on Agent-Based Social Simulations Using PAX Framework. In: IEEE International Conference on Systems, Man, and Cybernetics, Texas, USA (2009)
7. Pavón, J., Arroyo, M., Hassan, S., Sansores, S.: Agent-based modelling and simulation for the analysis of social patterns. Pattern Recogn. Lett. 29, 1039–1048 (2008)
8. Giácomo, B.N., Berger, L.M., Borenstein, D.: A Multiagent Method Applied to the Economic Analysis of Criminal Law. Economic Analysis of Law Review 1, 161–173 (2010)
9. Kiczales, G., et al.: Aspect-Oriented Programming. In: Aksit, M., Auletta, V. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)
10. Braga, D.S., Alves, F.O.M., Lima Neto, F.B., Menezes, L.C.S.: AspectNetLogo: Uma Proposta de Linguaguem Orientada a Aspectos para a Modelagem de Sistemas Multi-Agentes em Simulações Sociais. In: X Congresso Brasileiro de Inteligência Computacional, Fortaleza, CE. Sessão Técnica, 28 (Interfaces e Ferramentas) (2011)
11. Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., Griswold, W.G.: An Overview of AspectJ. In: Lee, S.H. (ed.) ECOOP 2001. LNCS, vol. 2072, pp. 327–353. Springer, Heidelberg (2001)
12. Gal, A., Schröder-Preikschat, W., Spinczyk, O.: AspectC++: Language Proposal and Prototype Implementation. In: Proceedings of the OOPSLA 2001 Workshop on Advanced Separation of Concerns in Object-Oriented Systems, Tampa, Florida (2001)
13. Wilensky, U.: NetLogo Rebellion Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (2004), `http://ccl.northwestern.edu/netlogo/models/Rebellion`
14. Epstein, J.M.: Modeling Civil Violence: An Agent-Based Computational Approach. Proceedings of the National Academy of Sciences 99, 7243–7250 (2002)

Parallel k-Most Similar Neighbor Classifier for Mixed Data

Guillermo Sanchez-Diaz[1], Anilu Franco-Arcega[2], Carlos Aguirre-Salado[1], Ivan Piza-Davila[3], Luis R. Morales-Manilla[4], and Uriel Escobar-Franco[4]

[1] Universidad Autonoma de San Luis Potosi,
Av. Dr. Manuel Nava no. 8, Zona Universitaria,
San Luis Potosi, SLP, Mexico, C.P. 78290
guillermo.sanchez@uaslp.mx
[2] Universidad Autonoma del Estado de Hidalgo,
Carr. Pachuca-Tulancingo Km. 4.5, Zona Universitaria,
Pachuca, Hgo., Mexico, C.P. 42084
[3] Instituto Tecnologico y de Estudios Superiores de Occidente,
Periferico Sur Manuel Gomez Morin 8585,
Tlaquepaque, Jal. Mexico, C.P. 45604
[4] Universidad Politecnica de Tulancingo,
Ingenierias 100, Col. Huapalcalco,
Tulancingo, Hgo., Mexico, C.P. 43629

Abstract. This paper presents a paralellization of the incremental algorithm *inc-k-msn*, for mixed data and similarity functions that do not satisfy metric properties. The algorithm presented is suitable for processing large data sets, because it only stores in main memory the k-most similar neighbors processed in step t, traversing only once the training data set. Several experiments with synthetic and real data are presented.

Keywords: K-most similar neighbor, K-nearest neighbor, classification, parallel algorithms.

1 Introduction

The k-nearest neighbor classifier (k-NN) [1] has been widely used as a non-parametric technique in Pattern Recognition due to its simplicity and good performance. The classical k-NN algorithm classifies a new object storing the whole training set in main memory and computing the distance of the training objects with the new object. Then, the algorithm sorts the obtained distances and finally, it obtains the k-objects with the lowest distance. The assigned class to the new object will be the majority class of the k-most similar objects.

Since this paradigm is used in different applications, several alternatives of k-NN classifiers have been developed. Some of them are presented in related work as fast k-NN classifiers [2, 3]. These algorithms have been developed to process large datasets over several problems, such as on-line value analysis, air traffic control or intrusion detection. However, some of these problems are defined by

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 586–593, 2012.

datasets with high dimensionality, where a comparison function can be very expensive, so it is recommended to decrease the number of comparisons with the training objects [4]. To solve different classification problems, it is necessary to process large training sets, which in some cases storing them in main memory is not feasible. Examples of problems with large training sets are: hyperspectral images of high resolution with at least 256 bands [5], banking or company transactions groups [6] and so on. To process this kind of problems, several incremental and static k-NN algorithms have been developed [7, 8]. However, if data are updated, these k-NN algorithms must be executed again to process the new whole training set. This drawback limits the use of static algorithms when updates are inevitable. For example, in sampling methods for data mining, it is required to have algorithms that allow updates in both, the sample and the original data, in order to evaluate the accuracy of the sample [9]. As an alternative to solve this problem, it has been developed a k-NN algorithm that allows updates over the dataset in an incremental way [10]. In environments where datasets are not static, k-NN algorithms can do the sample periodically; therefore they provide the accuracy of the data using an efficient way.

Nevertheless, most of these proposed k-NN classifiers have been designed to process only numeric data, which use metric functions. In this way, these methods apply metric properties to reduce the number of comparisons between objects. However, many real applications are described by numeric and categorical attributes (mixed data) [11] and in some cases, the comparison function does not satisfy these metric properties. For this reason, it is not always feasible to apply the developed k-NN classifiers to process objects with mixed descriptions.

In order to solve this problem, it has been developed a fast k-Most Similar Neighbor algorithm (*k-MSN*), which works with mixed data and uses a tree structure [12]. Although this algorithm gives a solution to the drawbacks presented in this section, *k-MSN* does not allow updates in the training set because, due to its static construction, it has to generate again the tree for including a new object. Recently, an incremental k-most similar algorithm (denoted by *inc-k-msn*) which works with mixed data and processes large data sets has been proposed [13]. This algorithm allows updates in the training set. However, when the size of the data set is very large or the dimentionality of the attributes is medium or high, the processing time of this algorithm increases significantly. Different applications in data mining require *k-NN* and *k-MSN* algorithms that allow updating the training set. Besides, it is important to avoid storing the whole dataset in main memory, especially when the set is very large. In the area of learning and information processing, when a significant amount of data is processed, in order to make more efficient the construction of a model, several algorithms have been parallelized [14, 15]. Some of them use multiprocessor computer architecture [16].

In this work, it is presented a paralellization of *inc-k-msn*, denoted by *par-inc-k-msn*, which gives a solution to some of the above problems, with the following contributions:

- The proposed paralellization of *inc-k-msn* allows processing large and very large mixed datasets, using a processing time significantly less than the incremental algorithm *inc-k-msn*. Besides, *par-inc-k-msn* allows inserting new objects to the training set, obtaining the k-most similar neighbors doing only the comparison of the inserted objects with the object to be classified, without processing again the whole updated training set. In addition, to classify a new object, it only traverses once the training set.
- *par-inc-k-msn* is feasible to be executed in personal computers or workstations with multi-core processors using multi-thread programming.

2 Definition of the Problem

Let U be an object universe, which it is not necessarily finite. Each object $o_i \in U$ is described by an attribute set $R = \{x_1, x_2, ..., x_n\}$, and the objects are distributed in d-classes $\{S_1, S_2, ..., S_d\}$. Each attribute $x_i \in R$ can takes values of a set M_i, $i = 1, ..., n$, which determines the nature of the attribute, numeric or categorical. Let $TM = \{o_1, o_2, ..., o_m\}$, $TM \subseteq U$ be the training set of the objects belonging to U. When mixed data are handled, a fundamental concept is the analogy or similarity between objects, which can be formalized by a function of similarity or dissimilarity [17].

A comparison criterion $C_i : M_i \times M_i \rightarrow L_i$ is associated to each attribute x_i, $i = 1, ..., n$, where $C_i\,(x_i(o), x_i(o)) = \min\{y\}$, $y \in L_i$ if C_i is a dissimilarity criterion between values of the attribute x_i or $C_i\,(x_i(o), x_i(o)) = \max\{y\}$, $y \in L_i$ if C_i is a similarity criterion between values of the attribute x_i. C_i is an evaluation of the similarity (or dissimilarity) degree among any two values of the attribute x_i, $i = 1, ..., n$, where L_i is a totally ordered set. A magnitude can be computed between each pair of objects in U. This magnitude is obtained applying a similarity function FS, which can be defined by any subset of R. When mixed data are handled, there are similarity functions that do not satisfy the triangle inequality [18].

The problem in this work is about supervised classification, from a training set TM, the algorithm allows assigning to a set of objects $o_t \in U$, $o_t \notin TM$ a specific class S_i, $i = 1, ..., d$.

3 The Proposed Algorithm

To perform the classification task, the proposed paralellization generates several threads running in different cores, assigning to each one a portion of the training data set. In the same way that *inc-k-msn*, each thread generated by *par-inc-k-msn* processes one by one the assigned portion of objects of the training set, even if new objects are added to it. *par-inc-k-msn* does not store the whole training set in main memory. Each generated thread only keeps the object to be classified and their k-most similar neighbors computed until step t (i.e. when the object t of the training set has been processed).

For each object of the training set that *par-inc-k-msn* processes, the k-most similar neighbors are obtained in a partial way, distributed in the spawned processes. When all objects in training set are processed, the proposed algorithm computes the same k-most similar neighbors than the classical algorithm, joining the partial k-msn calculated by each thread, but with the difference that *par-inc-k-msn* can continue adding new objects, following the same processing philosophy used with the first training objects. This fact guarantees that the accuracy of the classifier does not decrease. The principal difference of the proposed algorithm and static algorithms reported in related work is that these latter need the whole training set to obtain the k-most similar neighbors. On the other hand, *par-inc-k-msn* generates the k-most similar neighbors between each object of the training set and the object to be classified. In this way, at step $t+1$, the algorithm allows using the k-neighbors generated with the objects processed until step t, with the remaining training objects. This procedure allows to the proposed algorithm handling new added objects to the training set, processing them as any other training object. This characteristic makes different the proposed algorithm from the developed static algorithms, which have to process again the whole training set when it is updated.

In a general way, *par-inc-k-msn* works as follows: first, the algorithm calculates the number of objects assigned to each process shall, dividing the size of initial training set, by the number of threads to generate. Second, each thread is created and the portion of objects to be processed by it is assigned. Third, each thread executes the *inc-k-msn* algorithm, but working only over its portion of assigned objects, and when each process terminates, then the thread sends to the main process the list of the generated *k-msn*. Finally, the main process generates the list of *k-msn*. To generate the resultant list, each element is taken incrementally of the partial lists to add or delete it to the current list of k-neighbors. It may be that some elements previously stored being displaced by other elements that are more similar neighbors than they. If new objects are added to training set or there are new objects to be classified, the same procedure is applied. The diagram of the proposed paralellization is shown in Figure 1.

4 Experiments

In this section, in order to show the performance of *par-inc-k-msn*, a comparative table with some training sets reported in [12] is presented. Besides, it is shown the results of the proposed algorithm and the *inc-k-msn* algorithm over large training sets to compare their behavior. In the first example, to verify the behavior of the proposed algorithm using 2, 3 and 4 threads, it was taken only the tests that show the processing time of the *fast k-msn* algorithm, reported as the most efficient algorithm in [12]. This work uses 4 synthetic datasets with: 2000 (1800 training objects and 200 test objects), 3000 (2700 training objects and 300 test objects), 4000 (3600 training objects and 400 test objects) and 7200 (6480 training objects and 720 test objects) objects. In addition, it is shown the results obtained by *inc-k-msn* algorithm. All datasets reported in [12] have 2 attributes and a value of $k = 1$. Similarly, other dataset was generated with the

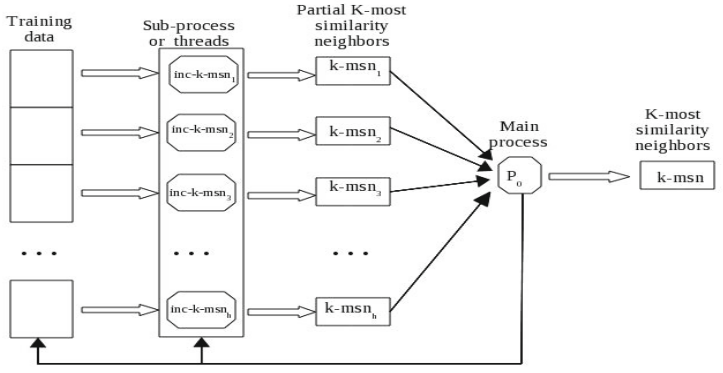


Fig. 1. Diagram of the proposed paralellization, usign h-threads

previous characteristics, in order to do a reliable comparison between processing times. These results are shown in table 1. *par-inc-k-msn* algorithm was designed in Java using multi-thread programming. In [12] was reported that tests were performed in a PC Intel Pentium Processor 3.0 Ghz, with 4 cores and 1 GB of RAM memory. In the other hand, all previous tests for *inc-k-msn* and *par-inc-k-msn* were performed in a PC with Intel Pentium Processor 3.3 Ghz, with 4 cores, 8 GB of RAM memory and using Mandriva Linux 2010.

Table 1. Processing time in seconds, obtained by *fast k-msn*, *inc-k-msn* and *par-inc-k-msn*, generating 2, 3 and 4 threads, with $k = 1$

Objects to process	Training objects	Test objects	Processing time (in seconds) fast k-msn	inc-k-msn	algorithm paralellization 2 process	3 process	4 process
2000	1800	200	0.485	0.4	0.4	0.2	0.2
3000	2700	300	0.812	0.6	0.6	0.3	0.3
4000	3600	400	1.139	0.8	0.8	0.8	0.4
7200	6480	720	7.63	4.32	2.16	2.16	0.72
726480	6480	720000	7630 (2.1 hr)	4320 (1.2 hr)	2160 (0.6 hr)	2160 (0.6 hr)	720 (0.2 hr)

In the second example, several synthetic datasets were used, each one with the following characteristics: a) 6 sets, from 500,000 to 3,000,000 of objects, with increments of 500,000 objects, described by 3 features, with $k = 5$, and classifying one object. Results of these experiments are shown in figure 2.

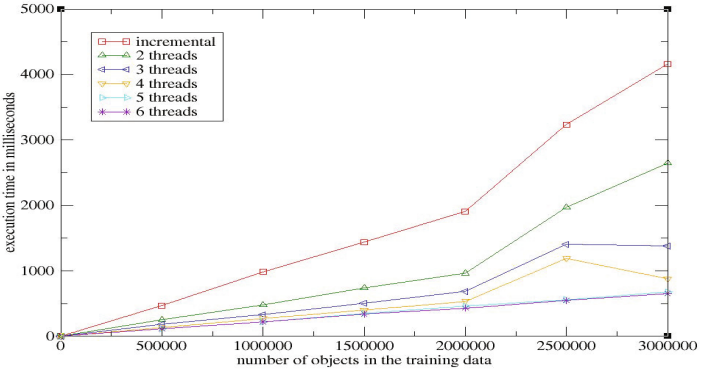


Fig. 2. Processing time in milliseconds, obtained by *inc-k-msn* and *par-inc-k-msn*, generating 2, 3, 4, 5 and 6 process, with $k = 5$

Table 2, shows the value of the ratio between the processing time execution of the incremental algorithm and their paralellization with 2, 3, 4, 5 and 6 processors, using as reference the values shown in figure 2. Besides, in this table, are incorporate the: minimum, maximum, average, and ideally ratio values, as a result of dividing the processing time between the number of used processors. These test were performed in a PC with AMD processor 3.5 Ghz, with 6 cores, 8 Gb of RAM memory and using Mandriva Linux 2010.

Table 2. Ratio between processing times of the incremental algorithm and their paralellization

Objects processed	2 process	3 process	4 process	5 process	6 process
500000	1.84	2.55	3.54	4.10	4.17
1000000	2.4	2.95	3.62	4.44	4.44
1500000	1.95	2.83	3.57	4.12	4.23
2000000	1.98	2.78	3.55	4.15	4.41
2500000	1.64	2.30	2.71	5.76	5.90
3000000	1.57	3.01	4.72	6.14	6.35
Minimum value	1.57	2.30	2.71	4.10	4.17
Maximum value	2.04	3.01	4.72	6.14	6.35
Average	1.83	2.73	3.61	4.78	4.91
Ideally	2	3	4	5	6

4.1 Discussion

In table 1, it can be noticed that proposed *par-inc-k-msn* algorithm has a higher behavior over *fast k-msn* and *inc-k-msn* algorithms, since it was designed to process large datasets (millions of data) and its application over this kind of datasets is adequate (including large training and/or test sets). When *par-inc-k-msn* uses a greater number of threads running in the cores of processor, the processing time decreases significantly. Besides, it is important to highlight that

algorithms like *fast k-msn* do not have the property of adding new training objects without the necessity of processing again the whole training set, property included in the proposed algorithm. In figure 2, it can be observed a higher behavior of the proposed algorithm over the *inc-k-msn* algorithm. When used 5 and 6 threads, *par-inc-k-msn* maintains its processing time stable, even when the number of objects in the training set increases significantly. As we can see in table 2, the average ratio obtained with the parallelizing algorithm is less and near than the ideal ratio, for all tests. Therefore, *par-inc-k-msn* is a good option to process large datasets. In addition, the following formula was obtained using the estimated time of how long it takes the algorithm to process a data set, using n-processors: $T/(k * n)$, where T is the processing time used by *inc-k-msn* and $k = 0.9$ was estimated, according to the results obtained.

5 Conclusions

This work introduces a paralellization of *inc-k-msn*, called *par-inc-k-msn*, which can handle mixed data and similarity functions that do not satisfy metric properties. The proposed algorithm allows processing datasets without storing them in main memory, besides new objects can be added to the training set, without processing again the whole training set. Besides, *par-inc-k-msn* advantages the resources provided by computers with multi-core processors. The proposed algorithm is able to process several similarity functions as well as distances to generate the k-most similar neighbors. However, the algorithm is not designed to process functions that are computed with all training objects, like the median. This is because the algorithm does not store all the comparisons with the training objects and it does not sort all the computed similarities. Other limitation of the proposed algorithm is that it cannot work with non-symmetric similarity functions.

In this work, there is not a specific application over a real problem using the proposed algorithm. However, there are several real classification problems for which the application of the proposed algorithm is appropriated, because they increase the number of the objects from the original dataset. Some applications of this kind are: *a*) detection and classification of epidemics in population, since the original sample of infected patients may increase with different mutations of the virus; *b*) fraud detection with credit card payments, where different ways of frauds are increased as they are detected; and so on.

Acknowledgements. This work was supported by PROMEP Mexico, under modality of NPTC call 2012. Project Number: PROMEP/103-5/11/3671.

References

[1] Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. Transactions on Information Theory (13), 21–27 (1967)
[2] Ramasubramanian, V., Paliwal, K.: Fast nearest-neighbor search based on approximation-elimination search. Pattern Recognition (33), 1497–1510 (2000)

[3] Yong-Sheng, C., Yi-Ping, H., Chiou-Shann, F.: Fast and versatile algorithm for nearest neighbor search based on lower bound tree. Pattern Recognition Letters 2(40), 360–375 (2007)
[4] Adler, M., Heeringa, B.: Search Space Reductions for Nearest-Neighbor Queries. In: Agrawal, M., Du, D.-Z., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 554–567. Springer, Heidelberg (2008)
[5] Sone, I., Olsen, R., Sivertsen, A., Eilertsen, G., Heia, K.: Classification of fresh Atlantic salmon (Salmo salar L.) fillets stored under different atmospheres by hyperspectral imaging. Journal of Food Engineering 109(3), 482–489 (2012)
[6] Chen, H., Yang, B., Wang, G., Liu, J., Xu, X., Wang, S., Liu, D.: A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method. Knowledge-Based Systems 24(8), 1348–1359 (2011)
[7] Xia, C., Lu, H., Ooi, B., Hu, J.: Gorder: an eficient method for knn join processing. In: Proc. of the 30th International Conference on Very Large Data Bases, pp. 756–767 (2004)
[8] Yu, C., Cui, B., Wang, S., Su, J.: Eficient index-based knn join processing for high-dimensional data. Inf. Softw. Technol. 4(49), 332–344 (2007); basado en el algoritmo incremental
[9] Bohm, C., Krebs, C.F.: The k-nearest neighbor join: turbo charging the kdd process. Knowledge Information Systems 6(6), 728–749 (2004)
[10] Yu, C., Zhang, R., Huang, Y., Xiong, H.: High-dimensional kNN joins with incremental updates. Geoinformatica (14), 55–82 (2010)
[11] Ruiz-Shulcloper, J.: Pattern recognition with mixed and incomplete data. Pattern Recognition and Image Analysis 18(4), 563–576 (2008)
[12] Hernandez-Rodriguez, S., Martinez-Trinidad, J., Carrasco-Ochoa, A.: Fast k most similar neighbor classifier for mixed data (tree k-MSN). Pattern Recognition (43), 873–886 (2010)
[13] Sanchez-Diaz, G., Escobar-Franco, U., Morales-Manilla, L.R., Piza-Davila, I., Aguirre-Salado, C., Franco-Arcega, A.: Incremental k most similar neighbor classifier for mixed data. Submitted to Revista Facultad de Ingenieria, Universidad de Antioquia
[14] Walkowiak, K., Woźniak, M.: Modeling of Network Computing Systems for Decision Tree Induction Tasks. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 759–766. Springer, Heidelberg (2009)
[15] Jin, Y., Gao, Y., Shi, Y., Shang, L., Wang, R., Yang, Y.: P^2LSA and P^2LSA+: Two Paralleled Probabilistic Latent Semantic Analysis Algorithms Based on the MapReduce Model. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 385–393. Springer, Heidelberg (2011)
[16] Barua, S., Alhajj, R.: Parallel Wavelet Transform for Spatio-temporal Outlier Detection in Large Meteorological Data. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 684–694. Springer, Heidelberg (2007)
[17] Ruiz-Shulcloper, J., Abidi, M.: Logical combinatorial pattern recognition: A review. Transworld Research Network, Kerala, India (2002)
[18] Hernández-Rodríguez, S., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: Fast k Most Similar Neighbor Classifier for Mixed Data Based on Approximating and Eliminating. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 697–704. Springer, Heidelberg (2008)

Feedback Linearization with a Neural Network Based Compensation Scheme

Josiane M.M. Fernandes, Marcelo C. Tanaka, Raimundo C.S. Freire Júnior, and Wallace M. Bessa*

Universidade Federal do Rio Grande do Norte, Dep. de Engenharia Mecânica
Campus Universitário Lagoa Nova, CEP 59072-970, Natal, RN, Brasil
josiane.eng.mec@gmail.com,ybrtanaka@yahoo.com.br,
{freirej,wmbessa}@ufrnet.br
http://www.posgraduacao.ufrn.br/ppgem

Abstract. This paper presents a nonlinear controller for uncertain single-input–single-output (SISO) nonlinear systems. The adopted approach is based on the feedback linearization strategy and enhanced by a Radial Basis Function neural network to cope with modeling inaccuracies and external disturbances that can arise. An application of this nonlinear controller to an electro-hydraulic actuated system subject to an unknown dead-zone input is also presented. The obtained numerical results demonstrate the improved control system performance.

Keywords: Electro-Hydraulic Systems, Feedback Linearization, Neural Networks, Nonlinear Control, Radial Basis Functions.

1 Introduction

Due to its simplicity, feedback linearization scheme is commonly applied in industrial control systems, specially in the field of industrial robotics. The main idea behind this control method is the development of control law that allows the transformation of the original dynamical system into an equivalent but simpler one. Although feedback linearization represents a very simple approach, an important drawback is the requirement of a perfectly known dynamical system, in order to ensure the exponential convergence.

Due to the adaptive capabilities of the artificial neural networks, it has been largely employed in the last decades to both control and identification of dynamical systems. In spite of the simplicity of this heuristic approach, in some situations a more rigorous mathematical treatment of the problem is required. Recently, much effort has been made to combine artificial neural networks with nonlinear control methodology [2–4, 6]

In this paper, a nonlinear controller is proposed to deal with uncertain single-input-single-output (SISO) nonlinear systems. The adopted approach is based on the feedback linearization method, but enhanced by a neural network compensation scheme to cope with modeling inaccuracies and external disturbances.

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 594–601, 2012.

Radial basis functions are used as activation functions and the related tracking error as input. Numerical simulations are carried out in order to demonstrate the improved performance of the proposed control scheme.

2 Control Scheme

Consider a class of n^{th}-order nonlinear systems:

$$x^{(n)} = f(\mathbf{x}, t) + b(\mathbf{x}, t)u + d \tag{1}$$

where u is the control input, the scalar variable x is the output of interest, $x^{(n)}$ is the n-th time derivative of x, $\mathbf{x} = [x, \dot{x}, \ldots, x^{(n-1)}]$ is the system state vector, d represents external disturbances and unmodeled dynamics, and $f, b : \mathbb{R}^n \to \mathbb{R}$ are both nonlinear functions.

Let us now define an appropriate control law that ensures the tracking of a desired trajectory $\mathbf{x}_d = [x_d, \dot{x}_d, \ldots, x_d^{(n-1)}]$, *i.e.* the controller should assure that $\tilde{\mathbf{x}} \to 0$ as $t \to \infty$, where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d = [\tilde{x}, \dot{\tilde{x}}, \ldots, \tilde{x}^{(n-1)}]$ is the related tracking error. On this basis, assuming that the state vector $\mathbf{x}$ is available to be measured and the functions f and b are well known, with $|b(\mathbf{x}, t)| > 0$, the following control law:

$$u = b^{-1}(-f + x_d^{(n)} - k_0\tilde{x} - k_1\dot{\tilde{x}} - \cdots - k_{n-1}\tilde{x}^{(n-1)} - d) \tag{2}$$

guarantees that $\mathbf{x} \to \mathbf{x}_d$ as $t \to \infty$, if the coefficients k_i $(i = 0, 2, \ldots, n-1)$ make the polynomial $p^n + k_{n-1}p^{n-1} + \cdots + k_0$ a Hurwitz polynomial.

The convergence of the closed-loop system could be easily established by substituting the control law, Eq. (2), in the nonlinear system, Eq. (1). The resulting dynamical system could be rewritten by means of the tracking error:

$$\tilde{x}^{(n)} + k_{n-1}\tilde{x}^{(n-1)} + \ldots + k_1\dot{\tilde{x}} + k_0\tilde{x} = 0 \tag{3}$$

where the related characteristic polynomial is Hurwitz.

However, since d is unknown the control law in Eq. (2) is not sufficient to ensure the exponential convergence of the tracking error to zero. On this basis, we propose the adoption of a neural network within the control law, in order to estimate d and to enhance the feedback linearization performance. Considering $\hat{d}$ the output of the neural network, the control law becomes:

$$u = b^{-1}\big(-f + x_d^{(n)} - k_0\tilde{x} - k_1\dot{\tilde{x}} - \cdots - k_{n-1}\tilde{x}^{(n-1)} - \hat{d}(\tilde{\mathbf{x}})\big) \tag{4}$$

Therefore, the related closed-loop system is:

$$\tilde{x}^{(n)} + k_{n-1}\tilde{x}^{(n-1)} + \ldots + k_1\dot{\tilde{x}} + k_0\tilde{x} = \tilde{d} \tag{5}$$

with $\tilde{d} = \hat{d} - d$.

Due to its simplicity and fast convergence feature, radial basis functions (RBF) are used as activation functions and the related tracking error as input. In this case, the output of the network is defined as:

$$\hat{d}(\tilde{\mathbf{x}}) = \sum_{i=1}^{M} w_i \cdot \varphi_i(||\tilde{\mathbf{x}} - \mathbf{t}||) \tag{6}$$

where $\varphi_i(\cdot)$ are the activation functions and $\mathbf{t}$ a vector containing the coordinates of the center of each activation function.

Now, the signal $\nu = \tilde{x}^{(n)} + k_{n-1}\tilde{x}^{(n-1)} + \ldots + k_1\dot{\tilde{x}} + k_0\tilde{x} - \hat{d}$ is used to train the neural network and the weights of the output layer are adjusted using the pseudo-inverse matrix.

Considering a training set $T = \{(\tilde{\mathbf{x}}, d)_1, (\tilde{\mathbf{x}}, d)_2, \ldots, (\tilde{\mathbf{x}}, d)_p\}$ and

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1M} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{p2} & \varphi_{p2} & \cdots & \varphi_{pM} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{bmatrix} \quad \therefore \quad [\varphi]\{w\} = \{d\} \tag{7}$$

the RBF weights are computed with the pseudo-inverse $[\varphi]^+$

$$\{w\} = [\varphi]^+\{d\} \tag{8}$$

and approximation error, E, by the euclidean norm

$$E = ||\{d\} - [\varphi]\{w\}|| \tag{9}$$

3 Illustrative Example: Electro-Hydraulic System

Electro-hydraulic actuators play an essential role in several branches of industrial activity and are frequently the most suitable choice for systems that require large forces at high speeds. Their application scope ranges from robotic manipulators to aerospace systems. Another great advantage of hydraulic systems is the ability to keep up the load capacity, which in the case of electric actuators is limited due to excessive heat generation.

However, the dynamic behavior of electro-hydraulic systems is highly nonlinear, which in fact makes the design of controllers for such systems a challenge for the conventional and well established linear control methodologies. In addition to the common nonlinearities that originate from the compressibility of the hydraulic fluid and valve flow-pressure properties, most electro-hydraulic systems are also subjected to hard nonlinearities such as dead-zone due to valve spool overlap.

In order to design the neural network feedback linearization controller, a mathematical model that represents the hydraulic system dynamics is needed. Dynamic models for such systems are well documented in the literature [5].

The electro-hydraulic system considered in this work consists of a four-way proportional valve, a hydraulic cylinder and variable load force. The variable load force is represented by a mass–spring–damper system. The schematic diagram of the system under study is presented in Fig. 1.

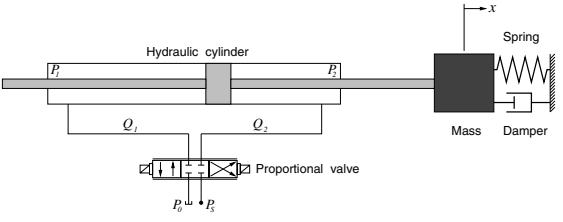


Fig. 1. Schematic diagram of the electro-hydraulic servo-system

The balance of forces on the piston leads to the following equation of motion:

$$F_g = A_1 P_1 - A_2 P_2 = M_t \ddot{x} + B_t \dot{x} + K_s x \tag{10}$$

where F_g is the force generated by the piston, P_1 and P_2 are the pressures at each side of cylinder chamber, A_1 and A_2 are the ram areas of the two chambers, M_t is the total mass of piston and load referred to piston, B_t is the viscous damping coefficient of piston and load, K_s is the load spring constant and x is the piston displacement.

Defining the pressure drop across the load as $P_L = P_1 - P_2$ and considering that for a symmetrical cylinder $A_p = A_1 = A_2$, Eq. (10) can be rewritten as

$$M_t \ddot{x} + B_t \dot{x} + K_s x = A_p P_L \tag{11}$$

Applying continuity equation to the fluid flow, the following equation is obtained:

$$Q_L = A_p \dot{x} + C_{tp} + \frac{V_t}{4\beta_e} \dot{P}_L \tag{12}$$

where $Q_L = (Q_1 + Q_2)/2$ is the load flow, with Q_1 and Q_2 as the flow in each chamber, C_{tp} the total leakage coefficient of piston, V_t the total volume under compression in both chambers and β_e the effective bulk modulus.

Considering that the return line pressure is usually much smaller than the other pressures involved ($P_0 \approx 0$) and assuming a closed center spool valve with matched and symmetrical orifices, the relationship between load pressure P_L and load flow Q_L can be described as follows

$$Q_L = C_d \omega \bar{x}_{sp} \sqrt{\frac{1}{\rho} \left[P_s - \mathrm{sgn}(\bar{x}_{sp}) P_L \right]} \tag{13}$$

where C_d is the discharge coefficient, ω the valve orifice area gradient, $\bar{x}_{sp}$ the effective spool displacement from neutral, ρ the hydraulic fluid density, P_s the supply pressure and $\text{sgn}(\cdot)$ is defined by

$$\text{sgn}(z) = \begin{cases} -1 \text{ if } & z < 0 \\ 0 \text{ if } & z = 0 \\ 1 \text{ if } & z > 0 \end{cases} \tag{14}$$

Assuming that the dynamics of the valve are fast enough to be neglected, the valve spool displacement can be considered as proportional to the control voltage (u). For closed center valves, or even in the case of the so-called critical valves, the spool presents some overlap. This overlap prevents from leakage losses but leads to a dead-zone nonlinearity within the control voltage, as shown in Fig. 2.

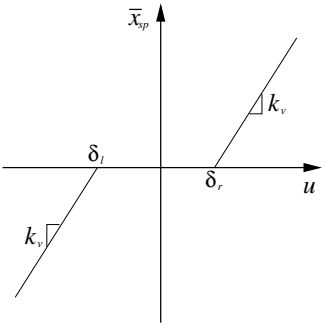


Fig. 2. Dead-zone nonlinearity

The dead-zone nonlinearity presented in Fig. 2 can be mathematically described by:

$$\bar{x}_{sp}(t) = \begin{cases} k_v \left(u(t) - \delta_l\right) & \text{if } \quad u(t) \leq \delta_l \\ 0 & \text{if } \quad \delta_l < u(t) < \delta_r \\ k_v \left(u(t) - \delta_r\right) & \text{if } \quad u(t) \geq \delta_r \end{cases} \tag{15}$$

where k_v is the valve gain and the parameters δ_l and δ_r depends on the size of the overlap region.

For control purposes, as shown by [1], Eq. (15) can be rewritten in a more appropriate form:

$$\bar{x}_{sp}(t) = k_v[u(t) - d] \tag{16}$$

where $d(u)$ can be obtained from Eq. (15) and Eq. (16):

$$d = \begin{cases} \delta_l & \text{if } \quad u(t) \leq \delta_l \\ u(t) & \text{if } \quad \delta_l < u(t) < \delta_r \\ \delta_r & \text{if } \quad u(t) \geq \delta_r \end{cases} \tag{17}$$

Combining equations (11), (12), (13), (16) and (17) leads to a third-order differential equation that represents the dynamic behavior of the electro-hydraulic system:

$$\dddot{x} = -\mathbf{a}^{\mathrm{T}}\mathbf{x} + bu - bd \tag{18}$$

where $\mathbf{x} = [x, \dot{x}, \ddot{x}]$ is the state vector with an associated coefficient vector $\mathbf{a} = [a_0, a_1, a_2]$ defined according to

$$a_0 = \frac{4\beta_e C_{tp} K_s}{V_t M_t} \quad ; \quad a_1 = \frac{K_s}{M_t} + \frac{4\beta_e A_p^2}{V_t M_t} + \frac{4\beta_e C_{tp} B_t}{V_t M_t} \quad ; \quad a_2 = \frac{B_t}{M_t} + \frac{4\beta_e C_{tp}}{V_t}$$

and

$$b = \frac{4\beta_e A_p}{V_t M_t} C_d w k_v \sqrt{\frac{1}{\rho}\left[P_s - \mathrm{sgn}(u)\left(M_t\ddot{x} + B_t\dot{x} + K_s x\right)/A_p\right]}$$

In this way, based on Eq. (4), the following nonlinear controller can be proposed to deal with the dynamic model presented in Eq. (18).

$$u = b^{-1}(\mathbf{a}^{\mathrm{T}}\mathbf{x} + \dddot{x}_d - 3\lambda\ddot{\tilde{x}} - 3\lambda^2\dot{\tilde{x}} - \tilde{x}) + \hat{d}(\tilde{x}, \dot{\tilde{x}}, \ddot{\tilde{x}}) \tag{19}$$

In order to evaluate the control system performance, numerical simulations were carried out. These simulation studies were performed with sampling rates of 500 Hz for control system and 1 kHz for dynamic model. The differential equations of the dynamic model were numerically solved with the fourth order Runge-Kutta method.

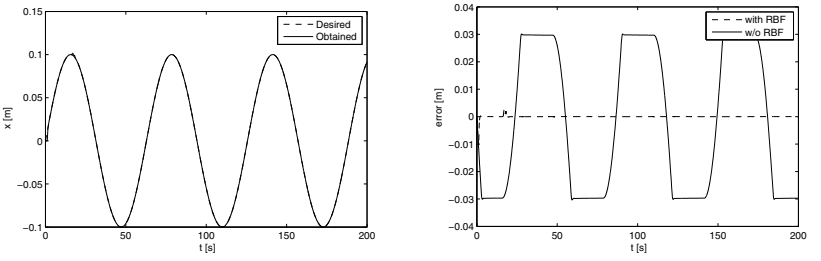
The adopted parameters for the electro-hydraulic system were $P_s = 7$ MPa, $\rho = 850$ kg/m^3, $C_d = 0.6$, $\omega = 2.5 \times 10^{-2}$ m, $A_p = 3 \times 10^{-4}$ m^2, $C_{tp} = 2 \times 10^{-12}$ m^3/(s Pa), $\beta_e = 700$ MPa, $V_t = 6 \times 10^{-5}$ m^3, $M_t = 250$ kg, $B_t = 100$ Ns/m, $K_s = 75$ N/m, $\delta_l = -0.5$ V, $\delta_r = 0.5$ V and $\lambda = 8$.

The dead-zone parameters are considered as unknown for the controller design and its effects should be compensated with adopted the neural network.

Regarding the RBF network, the number of neurons in the intermediate layer and the type of the activation functions, as well as how they are distributed over the input space, could be heuristically defined to accommodate designer's experience and experimental knowledge. On this basis, 21 neurons were adopted for the intermediate layer and the activation functions were chosen as of the gaussian type. Figure 3 shows the results obtained with $x_d = 0.5\sin(0.1t)$ m.

As observed in Fig. 3, despite the unknown dead-zone input, the proposed control scheme allows the electro-hydraulic actuated system to track the desired trajectory with a small tracking error. Through the comparative analysis shown in Fig. 3(b), the improved performance of the proposed controller over the uncompensated counterpart can be easily ascertained.

Since noise contamination is unavoidable in experimental data acquisition, it is important to evaluate its effect on control procedures. In order to simulate experimental noisy data sets, a white Gaussian noise was introduced in the signal:

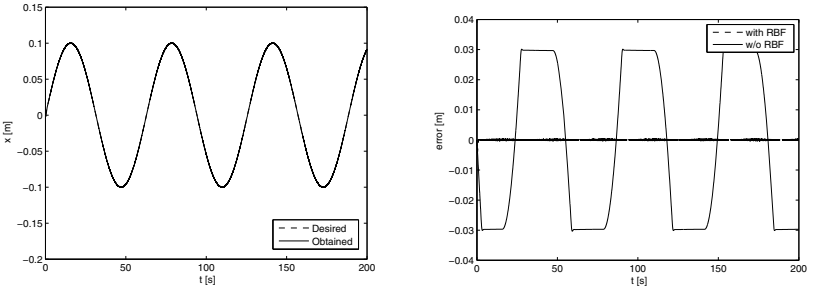


(a) Tracking performance (b) Tracking error

Fig. 3. Tracking of $x_d = 0.1\sin(0.1t)$ m

$$x_R(t) = x(t) + \xi$$
$$\dot{x}_R(t) = \dot{x}(t) + \xi$$
$$\ddot{x}_R(t) = \ddot{x}(t) + \xi$$

where x_R, $\dot{x}_R$ and $\ddot{x}_R$ represent the noisy measured state variables, x, $\dot{x}$ and $\ddot{x}$ the related clean signals, and ξ the white Gaussian noise. Here, noise level was parametrized using the signal-to-noise ratio (SNR). Figure 4 show trajectory tracking with SNR = 60.

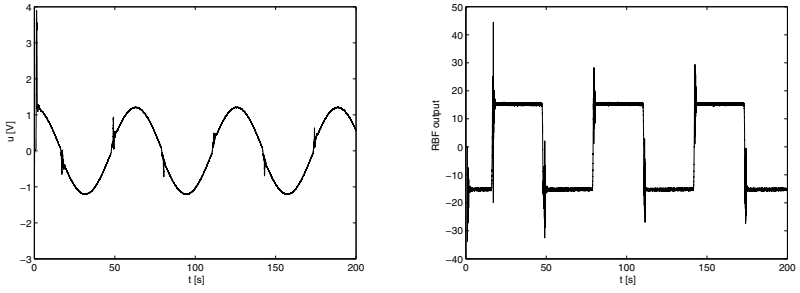


(a) Tracking performance (b) Tracking error

Fig. 4. Tracking of $x_d = 0.1\sin(0.1t)$ m with SNR = 60

As observed in Fig. 4, even in the presence of noisy measured signals and a dead-zone input the proposed controller is able to provide trajectory tracking. By comparing the tracking error obtained with and without the neural network compensation scheme, Fig. 4(b), the superior performance of the proposed controller is noticeable.

The improved performance of proposed scheme is due to the ability of the RBF network to recognize and compensate for external disturbances and modeling imprecisions. Figure 5 shows the control signal and the output of the network considering SNR = 60.



(a) Control voltage u (b) RBF output $\hat{d}$

Fig. 5. Control signals for the tracking of $x_d = 0.1\sin(0.1t)$ m with SNR = 60

4 Concluding Remarks

The present work addressed the problem of controlling uncertain nonlinear systems with a feedback linearization approach, but enhanced with a RBF neural network. In order to illustrate the controller design method, the proposed scheme is applied to an electro-hydraulic actuated system. The control system performance is confirmed by means of numerical simulations. The adoption of a RBF network provides an smaller tracking error due to its ability to compensate for uncertainties with respect to dynamic model, as well as for noisy signals.

Acknowledgments. The authors would like to acknowledge the support of the Brazilian National Research Council (CNPq), the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES), the Brazilian National Agency of Petroleum, Natural Gas and Biofuels (ANP) and the German Academic Exchange Service (DAAD).

References

1. Bessa, W.M., Dutra, M.S., Kreuzer, E.: An adaptive fuzzy dead-zone compensation scheme and its application to electro-hydraulic systems. Journal of the Brazilian Society of Mechanical Sciences and Engineering 32(1), 1–7 (2010)
2. Boutalis, Y.S.: Neural network approaches for feedback linearization. Control Engineering and Applied Informatics 6(1), 15–26 (2004)
3. Braake, H.A.B., Can, E.J.L.V., Scherpen, J.M.A., Verbruggen, H.B.: Control of nonlinear chemical processes using neural models and feedback linearization. Computers & Chemical Engineering 22(7–8), 1113–1127 (1998)
4. Poursamad, A.: Adaptive feedback linearization control of antilock braking systems using neural networks. Mechatronics 19, 767–773 (2009)
5. Walters, R.B.: Hydraulic and Electro-hydraulic Control Systems, 2nd edn. Kluwer Academic, London (2000)
6. Yeşildirek, A., Lewis, F.L.: Feedback linearization using neural networks. Automatica 31(11), 1659–1664 (1995)

Local Fuzzy Pattern: A New Way for Micro-pattern Analysis

Raissa Tavares Vieira, Carlos Eduardo de Oliveira Chierici, Carolina Toledo Ferraz, and Adilson Gonzaga

Department of Electrical Engineering - EESC/USP,
Av. Trabalhador São Carlense, 400 São Carlos, SP, Brazil
raissa@ieee.org,
{chierici,caroltoledoferraz}@gmail.com, agonzaga@sc.usp.br

Abstract. The aim of this paper is to introduce a new methodology for micro-pattern analysis in digital images. The gray-level pixels' structure in an image neighborhood describes a spatial specific context. Edge, line, spot, blob, corner or texture can be described by this structure. The gray-level values of the image pixel are interpreted as a fuzzy set, and each pixel gray-level as a fuzzy number. A membership function can be defined to describe the membership degree of the central pixel to the others in an image neighborhood. We have called this method the Local Fuzzy Pattern (LFP). If a sigmoid membership function is used, the proposed methodology describes the texture very well, and if a symmetrical triangular membership function is applied, the LFP is better for edge's detection. The results were compared to the Local Binary Pattern (LBP), for texture classification getting the better hit-rate. Our proposed formulation for the LFP is a generalization of previously published techniques, such as Texture Unit, LBP, FUNED, and Census Transform.

Keywords: micro-pattern analysis, fuzzy numbers, texture analysis, edge detection.

1 Introduction

Visual patterns are ambiguous by nature. Digital images are often corrupted by noise or distorted by the acquisition process. Objects in scenes are not always well-defined, and the knowledge about the objects in the scene is described in vague terms. Fuzzy Sets theory and Fuzzy Logic constitute a quite appropriate alternative to deal with such uncertainties, in compensation for the conventional systems, based on the traditional logic (crisp) built on answers of the true or false type. In some image processing techniques, the neighborhood of a pixel is considered for analyzing the image characteristics. A pixel's neighborhood can describe the spatial context of the image, such as edge, line, spot, blob, corner and texture. This spatial pixel distribution in a pixel's neighborhood is called micro-pattern. The topographical characteristics of micro-patterns are more robust to shift, scale, and changes in illumination. In the computer vision literature, some

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 602–611, 2012.

approaches were developed to extract these characteristics. The Scale Invariant Feature Transform (SIFT) [1] is a method for micro-pattern analysis that extract some points called key features. The Local Binary Pattern (LBP) [2] was first applied for texture analysis [3], but was later extended to background modeling [4], face detection [5], facial expression analysis [6] and face recognition [7]. The micro-pattern representation was also extended to Gabor magnitude features [8] and Gabor phase features [9] to increase the discrimination capacity. The Fuzzy Local Binary Pattern (FLBP) extends the LBP approach by incorporating fuzzy logic in the representation of local patterns of texture [10] [11]. This methodology assumes that a local neighborhood can be partially characterized by more than one binary pattern as a result of noise-originated uncertainty in the pixel values. The results show that the FLBP leads to improvement in texture classification compared with the original LBP [12]. The problem of using fuzzy logic is the computational cost.

In contrast with the FLBP that uses fuzzy logic for Binary Pattern fuzzification, the objective of this paper is to model the gray-level distribution of an image micro-pattern as a fuzzy set, and based on membership functions generate fuzzy-codes that represents the membership degree of each neighborhood pixel to the central one. We call the proposed methodology the Local Fuzzy Pattern (LFP). For micro-pattern analysis, we have tested two membership functions: the sigmoid curve and the symmetrical triangular distribution.

The rest of this paper comprises four sections. Section 2 presents the proposed LFP methodology. Section 3 describes the application to texture analysis. Section 4 refers to the experimental evaluation and the classification results obtained. The conclusions are provided in the last section.

2 Methodology

For our proposal, we considered the pixels' gray-levels as fuzzy numbers, thus, the inherent variability of image gray values were incorporated, providing a more powerful approach for the treatment of digital images compared with the classic treatment that is based on an analytic formulation.

A fuzzy set is a pair (U, μ) where U is a set and $\mu : U \rightarrow [0, 1]$. For each $x \in U$, the value $\mu(x)$ is called the membership degree of x in (U, μ). For a finite set $U = \{x_1, x_2, ..., x_n\}$, the fuzzy set (U, μ) is denoted by $\{\mu(x_1)|x_1, ..., \mu(x_n)|x_n\}$. x is called a fuzzy member if $0 < \mu(x) < 1$. The function $\mu(x)$ is called the membership function of the fuzzy set (U, μ).

Fuzzy numbers are constituted by fuzzy sets defined in discreet or continuous discourse universes, which allow quantifying the uncertainty and imprecision associated with some specific information. A fuzzy number is a convex, normalized fuzzy set $A \subseteq \mathbb{R}$ represented by a membership function, whose discourse universe is the real straight line. The concept of fuzzy numbers, as fuzzy subsets of real numbers, is a powerful paradigm for representing imprecision in numerical information. In many aspects, fuzzy numbers depict the physical world more realistically than single-valued numbers. The concept takes into account the fact that all phenomena in the physical universe have a degree of inherent uncertainty.

Let the gray-levels of an image be considered fuzzy numbers. For each $g(i,j)$ image pixel, the membership is calculated with regard to a specific region, considering the neighbors with gray-levels near $g(i,j)$.

The definitions of different membership functions may be based on the properties of the micro-pattern neighborhood $W \times W$ of a central pixel $g(i,j)$ in a digital image. We propose that the membership degree of the central pixel $g(i,j)$ to the micro-pattern defined by the neighborhood $W \times W$ should be determined by Equation 1.

$$\hat{\mu}_{g(i,j)} = \frac{\sum_{k=1}^{W}\sum_{l=1}^{W}(f_{g(i,j)}P(k,l))}{\sum_{k=1}^{W}\sum_{l=1}^{W}P(k,l)}, \tag{1}$$

where, $f_{g(i,j)}$ is the membership function and $P(k,l)$ is a weighting matrix for the neighborhood $W \times W$ with the same dimension.

Through Equation 1, it is possible to derive some previously published approaches for micro-pattern analysis. The Fuzzy Number Edge Detector (FUNED) [14] can be obtained by using the triangular symmetric membership function shown in Equation 2.

$$f_{g(i,j)} = max(0, 1 - \frac{|g(i,j) - A(k,l)}{\delta}), \tag{2}$$

where, $A(k,l)$ are the pixels in the $W \times W$ neighborhood, δ is the fuzzy number span, and the weighting matrix has the central value equal to 0 and the other elements equal to 1.

The Local Binary Pattern can be derived from Equation 1 by using a crisp function as the Heaviside Step Function shown in Equation 3.

$$f_{g(i,j)} = H[A(k,l) - g(i,j)], \tag{3}$$

where,

$$H[n] = \begin{cases} 0, \text{ se} & n < 0, \\ 1, \text{ se} & n \geq 0. \end{cases}$$

Taking into account the basic LBP with a neighborhood of pixels, the weighting matrix will be:

$$P(k,l) = \begin{bmatrix} 1 & 2 & 4 \\ 128 & 0 & 8 \\ 64 & 32 & 16 \end{bmatrix}$$

The LBP_{code} (values between 0 and 255) will be obtained by Equation 4.

$$LBP_{code} = \hat{\mu}_{g(i,j)} \sum_{k=1}^{W}\sum_{l=1}^{W} P(k,l). \tag{4}$$

The LBP_{code}, thus, is a particular case of the LFP approach.

The Census Transform (CT) proposed by Zabih and Woodfill [15] differs from the LBP by the order of the bit string. Similarly, the CT can be inferred from the LFP by using an appropriate weighting matrix.

The Texture Unit proposed by He and Wang [16] can also be derived from the LFP. The membership function must be as in Equation 5.

$$f_{g(i,j)} = 1 + sgn[A(k,l) - g(i,j)], \tag{5}$$

where,

$$sgn(x) = \begin{cases} -1, \text{ if } x < 0, \\ 0, \text{ if } x = 0, \\ 1, \text{ if } x > 0 \end{cases}$$

and

$$P(k,l) = \begin{bmatrix} 1 & 3 & 9 \\ 2187 & 0 & 27 \\ 729 & 243 & 81 \end{bmatrix}$$

By using the right membership function, our proposed methodology could extract specific features from the image, based on the micro-pattern processing.

We propose in this work the analysis of two micro-pattern descriptor based on the LFP methodology, and comparing then with the LBP approach. The first one is a smooth approximation to the step function by a logistic function, that is, a common sigmoid curve membership function as in Equation 6.

$$f_{g(i,j)} = \frac{1}{1 + e^{\frac{-[A(k,l) - g(i,j)]}{\beta}}}, \tag{6}$$

where β is the curve slope. We named this descriptor as LFP-s and the second is a symmetrical triangular membership function as stated by Equation 2, the LFP-t.

Because the LBP method uses the "crisp version of the sigmoid curve" for texture analysis, we have performed a comparative performance evaluation of our approach.

3 Micro-pattern for Texture Analysis

For the performance evaluation of our method, we used a texture database named Brodatz's album [17], and we tested the proposed approach by using 111 images of synthetic and natural textures. Brodatz's photo album is a well-known benchmark database for evaluating texture recognition algorithms. Each texture image is considered to be a class, with a dimension of 640×640 pixels. We randomly extracted ten samples with a size of 50×50 pixels from each class.

To analyze the LFP approach performance, the 1110 random samples were compared with the LBP descriptor, either for LFP-s as for LFP-t. We generated

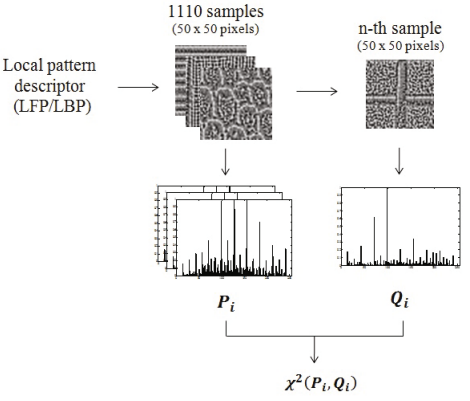


Fig. 1. Discrimination method

the histogram from each sample and used the Chi-square distance (Equation 7) for histogram comparison, as illustrated in Fig. 1.

Then, we choose the best 40 images having the best results for the LFP-t. This was done because that, LFP-t is the same formulation than FUNED, that is, an edge detector, and LFP-s is a smoothed LBP, widely applied for texture analysis. From these 40 images, we have extracted randomly 10 samples of each one, generating an image set of 400 samples. Thus, we evaluated the best performance for different membership functions, and we analyzed the type of texture each one works better.

LFP has its values in the 0-1 interval, so, for each sample we generated the LFP histogram converting it to an 8-bit gray level scheme by multiplying each membership degree value by 255 and rounding the result. Therefore, each image sample histogram has 256 bins (integer values from 0 to 255), the same as the original LBP approach.

The LFP-s needs to define the curve slope (β), and the LFP-t needs to specify the fuzzy number span δ. Fig. 2 shows the sigmoid's curve slope (β) performance for Brodatz's album.

We choose $\beta = 1.005$ as the best-trained value, and the weighting matrix as:

$$P(k,l) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

This weighting matrix represents the non prevalence position for any pixel turning the LFP, which is rotation invariant.

The fuzzy number span was empirically established δ=50 after testing some values for the Brodatz's album. The weighting matrix for the LFP-t was adopted the same as the FUNED approach,

$$P(k,l) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

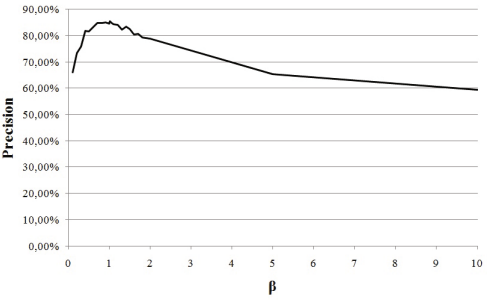


Fig. 2. Slope of sigmoid curve

By leave-one-out cross-validation, we compared the histograms from each sample with the rest of those from the sample set (1109 samples) by using the Chi-square distance [18] as shown in Equation 7.

$$\chi^2(P_i, Q_i) = \frac{1}{2}\sum_{i=1}^{255}\frac{(P_i - Q_i)^2}{(P_i + Q_i)}, \tag{7}$$

where, Q_i are the gray-level frequencies of the query sample and P_i are the gray-level frequencies of the compared sample from the set. One query sample is correctly classified if it has the lowest distance value to one of the 9 samples of the same class.

4 Results

We generated confusion matrices with True Positives (TP), or the number of correctly classified samples, and False Negatives (FN), or the wrongly classified samples, for all of the query samples. The Hit-rate or Sensitivity was calculated as shown in Equation 8.

$$S = \frac{TP}{TP + FN}. \tag{8}$$

The LFP performance was first evaluated for texture analysis by using the 1110 samples. The Hit-rate is shown in Table 1, considering the two compared methodologies. The LFP-s surpassed the original LBP by nearly 5%. Most likely, this occurs because the LBP is a crisp version of the LFP-s, and thus, some texture was better represented as a micro-pattern by our methodology. The LFP-t is not appropriate for general texture analysis. Considering all the 1110 images, the LFP-t had a badly performance showing a hit-rate of only 60.72%.

However, by analyzing the confusion matrix, it has been verified that some samples were better classified by the LFP-t than LFP-s or LBP. Some of these samples are shown in Fig. 3. The general performance of the LFP-s is outstanding considering the other two compared methods. Some images' samples showed a superior performance as observed in Fig. 4. Otherwise, when the LBP has a better performance than the LFP-s and LFP-t, as shown in Fig. 5, the number of correct

Table 1. Results' comparison for all the 111 textures from Brodatz's album

1110 image's samples	
Method	**Hit-rate**
LBP	80.18%
LFP-s	85.50%
LFP-t	60.72%

classifications is not higher than 2. That is, the performance comparison showed that some types of texture were very well evaluated with the LFP-s approach, and some others can be analyzed either by the LBP or LFP. The LFP-t is better for texture, or micro-patterns, containing visible edges. These results are in accordance with the Fuzzy Number Edge Detector (FUNED), as published.

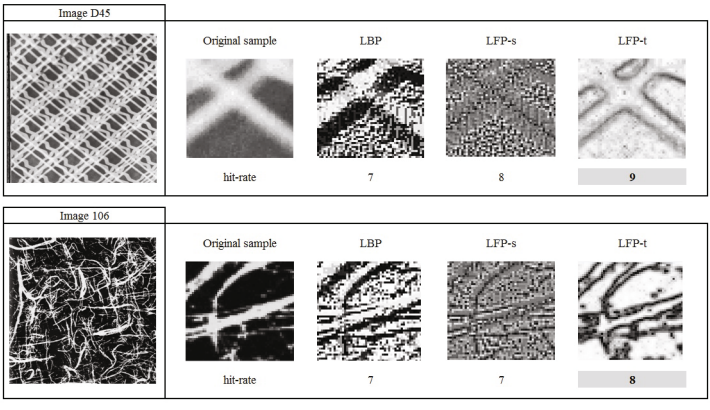


Fig. 3. Samples with better performance for the LFP-t

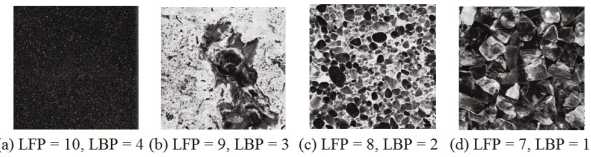
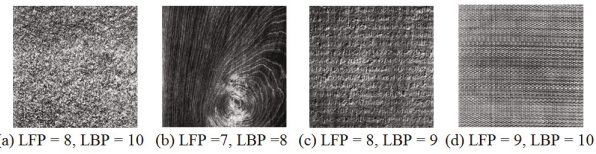


Fig. 4. Better performance for the LFP-s versus the LBP

Fig. 6 shows a processed texture without the predominance of edges. For the most kind of texture, considering all the 111 textures of the Brodatz's album, the LFP-s had success in the classification task. For the image in this figure, the LFP-s has classified all the ten samples, whereas the other two have hit only seven. Qualitatively it is easy to see that this texture doesn't have predominance of edges, and the "crispness" of the LBP doesn't allow capturing the gray-level nuances.



(a) LFP = 8, LBP = 10 (b) LFP =7, LBP =8 (c) LFP = 8, LBP = 9 (d) LFP = 9, LBP = 10

Fig. 5. Better performance for the LBP versus the LFP-s

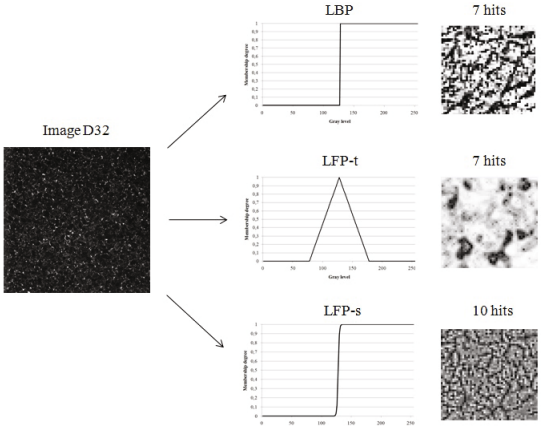


Fig. 6. Example of processed samples

Table 2. Results' comparison for 40 textures selected from Brodatz's album

400 image's samples	
Method	**Hit-rate**
LBP	92.50%
LFP-s	95.25%
LFP-t	90.50%

Aiming to evaluate the membership function and its potential for texture discrimination, we took the best 40 classified samples by the LFP-t, that is, the worst performance showed for the 111 images, among the three tested methods. The results are shown in Table 2. As expected, the micro-patterns of these samples are those with predominance of edges, so, the LFP-t is a good tool for this kind of texture analysis. However, the performance for the LBP and LFP-s have increased, showing that the best membership function for general texture's analysis is the sigmoid curve.

5 Conclusion

In this paper, we have shown a new methodology for micro-pattern analysis. We have formulated a new equation based on membership functions of fuzzy

numbers. We have assumed that the neighborhood of a pixel in digital images should be modeled as a fuzzy set, taking in account the pixels gray levels. The Local Fuzzy Pattern (LFP) has been defined, and it has been proven that it is a generalization of some previously published methods. By using a sigmoid (LFP-s) and a symmetrical triangular (LFP-t) function for calculating the membership degree of a central pixel of a neighborhood, we have applied our approach, for texture analysis. Brodatz's album with 111 classes was randomly sampled, generating 10 samples of each class. After processing these images by the LFP-s, LFP-t and LBP approaches, we have compared the hit-rate reached by using the Chi-square distance. Our results showed that the LFP-s surpasses the LBP by 5% in terms of correct classifications. This result is justified because the LBP-s is a particular case of the LFP; that is, if we adopt a crisp membership function for the LFP formulation, we obtain the LBP approach.

The LFP-t is the same for the FUNED [13] [14] approach. It was proposed recently for edge detection. We used the symmetrical triangular membership function, and we concluded that if a texture has the predominance of edges, LFP-t could be a good texture descriptor.

The proposed methodology is a general and robust method for micro-pattern analysis. By using the correct membership function, it is possible to codify and to note the representative feature of the analyzed pixel neighborhood. Therefore, micro-patterns, such as texture, edges, and corners should be better extracted by the LFP methodology.

Acknowledgments. The authors would like to thank the So Paulo State Foundation for Supporting Research (FAPESP) for their financial support of this research.

References

1. Lowe, D.G.: Object Recognition from Local Scale-Invariant Features. In: 7th IEEE International Conference on e-Science, pp. 1150–1157 (1999)
2. Ojala, T., Pietikäinen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recognition 29, 51–59 (1996)
3. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 971–987 (2002)
4. Heikkilaa, M., Pietikäinen, M.: A Texture-Based Method for Modeling the Background and Detecting Moving Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 657–662 (2006)
5. Hadid, A., Pietikäinen, M., Ahonen, T.: A Discriminative Feature Space for Detecting and Recognizing Faces. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 797–804 (2004)
6. Zhao, G., Pietikäinen, M.: Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 915–928 (2007)

7. Ahonen, T., Hadid, A., Pietikäinen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 2037–2041 (2006)
8. Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition. In: IEEE International Conference on Computer Vision, vol. 1, pp. 786–791 (2005)
9. Zhang, B., Shan, S., Chen, X., Gao, W.: Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition. IEEE Transactions on Image Processing 16, 57–68 (2007)
10. Iakovidis, D.K., Keramidas, E.G., Maroulis, D.E.: Fuzzy Local Binary Patterns for Ultrasound Texture Characterization. In: Campilho, A., Kamel, M.S. (eds.) ICIAR 2008. LNCS, vol. 5112, pp. 750–759. Springer, Heidelberg (2008)
11. Iakovidis, D.K., Keramidas, E.G., Maroulis, D.: Fusion of fuzzy statistical distributions for classification of thyroid ultrasound patterns. Artificial Intelligence in Medicine 50, 33–41 (2010)
12. Keramidas, E., Iakovidis, D., Maroulis, D.: Fuzzy Binary Patterns for uncertainty-aware texture representation. Electronic Letters on Computer Vision and Image Analysis 10, 1577–5097 (2011)
13. Boaventura, I.A.G., Gonzaga, A.: Border Detection in Digital Images: An Approach by Fuzzy Numbers. In: Seventh International Conference on Intelligent Systems Design and Applications, pp. 341–346. IEEE Computer Society Press (2007)
14. Boaventura, I.A.G., Gonzaga, A.: Edge detection in digital images using fuzzy numbers. International Journal of Innovative Computing and Applications 2, 1–12 (2009)
15. Zabih, R., Woodfill, J.: Non-parametric Local Transforms for Computing Visual Correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994, Part II. LNCS, vol. 801, pp. 151–158. Springer, Heidelberg (1994)
16. He, D.-C., Wang, L.: Texture Unit, Texture Spectrum, And Texture Analysis. IEEE Transactions on Geoscience and Remote Sensing 28, 509–512 (1990)
17. Brodatz, P.: Textures: A Photographic Album for Artists and Designers. Dover Publications, New York (1999)
18. Pele, O., Werman, M.: The Quadratic-Chi Histogram Distance Family. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 749–762. Springer, Heidelberg (2010)

Evolving Fuzzy Classifier Based on the Modified ECM Algorithm for Pattern Classification

Maurílio J. Inácio[1], Renato D. Maia[2], and Walmir M. Caminhas[3]

[1] Computer Engineering Dept., FACIT, Montes Claros, MG, Brazil
Computer Sciences Dept., UNIMONTES, Montes Claros, MG, Brazil
maurilio@femc.edu.br
[2] Automation and Control Engineering Dept., FACIT, Montes Claros, MG, Brazil
Computer Sciences Dept., UNIMONTES, Montes Claros, MG, Brazil
Institute of Agricultural Sciences, UFMG, Montes Claros, MG, Brazil
renato.dourado@unimontes.br
[3] Electronic Engineering Dept., UFMG, Belo Horizonte, MG, Brazil
caminhas@cpdee.ufmg.br

Abstract. Nowadays, online and real-time pattern classification applications are required in many areas. Most classification algorithms are suitable only for off-line applications. Using the concept of evolving intelligent systems, this paper proposes an evolving fuzzy classifier capable of creating the rule base in online mode and real-time. The proposed evolving fuzzy classifier is based on a new clustering algorithm that consists of an improved version of the Evolving Clustering Method (ECM). Experiments with well-known benchmark classification problems indicated that the proposal is promising.

Keywords: Pattern classification, fuzzy classifier, clustering algorithm, evolving intelligent systems.

1 Introduction

Nowadays, many daily activities rely directly or indirectly on pattern classification methods use. Examples of this are fingerprint recognition in security systems, handwriting recognition on computers' touchscreen displays, DNA sequences identification on medical diagnosis software or faults diagnosis in industrial machines. Pattern classification consists of, from a set of "raw" data, the capability to perform an action based on the category of each pattern [1]. One of the challenges in pattern classification problems is to classify precisely each sample data, i.e., assign a data pattern to its category or class. For this task, classes must be known a priori or obtained by some learning method [1], [2]. However, most pattern classification algorithms are only suitable for off-line applications, i.e., requiring that the entire data set to be available to perform training. However, in many real applications, for example, industrial machines fault diagnosis from data obtained in real time, the classification process must be performed online, receiving patterns in a data stream.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 612–621, 2012.

A new paradigm called Evolutionary Intelligent Systems (EIS) [3] [4] has been employed in developing new algorithms for unsupervised learning. Evolving Intelligent Systems are those capable of, from input dada acquired online – and many times in real time –, gradually determine both their structure and their parameters. This feature allows EIS to solve problems in dynamic and non-stationary environments as typically found in industrial, medical, military, robotics, and others areas [4]. Several published works in recent years demonstrate the growth of EIS application and its success handling complex real-world problems in modeling, control, classification or prediction [5].

Therefore, this paper proposes an evolving fuzzy classifier based on the EIS concept. In this classifier, prior knowledge of the class number is not necessary, neither is the existence of the entire data set in order to perform pattern classification. The proposed evolving fuzzy classifier is able to create a fuzzy rule base in online mode and in real time, so that the system learns to classify patterns as input data is received. With this feature, the proposed classifier can be applied to complex real-world problems where other types of classifiers are not suitable. The proposed evolving fuzzy classifier is based on a new evolving clustering algorithm, which is an improved version of the Evolving Clustering Method (ECM) proposed by Kasabov and Song [6]. In this version, the procedures for updating the centers and radii have been modified in order to obtain a better representation of the clusters.

The remainder of this paper is organized as follows: Section 2 presents the basic concepts and models of evolving intelligent systems; Section 3 describes the proposed evolving fuzzy classifier and the new evolving clustering algorithm; Section 4 presents and discusses the experimental results; Section 5 presents the conclusions and suggestions for future work.

2 Evolving Intelligent Systems

EIS are evolving systems that gradually determine both its structure and its parameters, using input data acquired in online mode and often in real time. Such systems are different from adaptive systems, which are only able to adapt their internal parameters to the problem in question [7]. EIS are also different from systems that employ evolutionary algorithms, in which a process based on selection, crossover and mutation operators is capable of evolving, from a population of individuals, a better adapted individual [8]. Adaptive and evolutionary systems are more suitable for applications where changes in the environment are relatively slow and usually these systems adapt/evolve in off-line mode [6].

It is important to highlight that different theoretical and practical approaches can be used to implement EIS, such as systems based on artificial neural networks, fuzzy rules, intelligent agents or hybrid systems [5]. But, regardless of the approach to be used, the main features of evolving intelligent systems are: (1) its structure is neither fix nor defined a priori; it grows (expands or shrinks) naturally as the system evolves; (2) its parameters are adjusted (adapted) as the system evolves; and (3) its operation is continuous, as the learning process occurs online and, if necessary, in real time.

EIS adopt as model a natural form of individuals' evolution, particularly of humans. In nature, individuals evolve during their life, gradually and continuously, from the learning and the use of accumulated knowledge. Experience allows individuals to become more and more apt to overcome difficulties imposed by environmental changes and thus able to solve increasingly difficult problems. The generic structure of EIS is illustrated in Figure 1. The structure's main component is the intelligent system itself (fuzzy system, for example), responsible for reasoning and decision-making tasks. These tasks are performed from the input data and using the information accumulated in the knowledge base. A mechanism for online learning is used to update the knowledge base, giving the system evolving characteristics, enabling it to acquire new knowledge from experience and, therefore, to solve new problems.

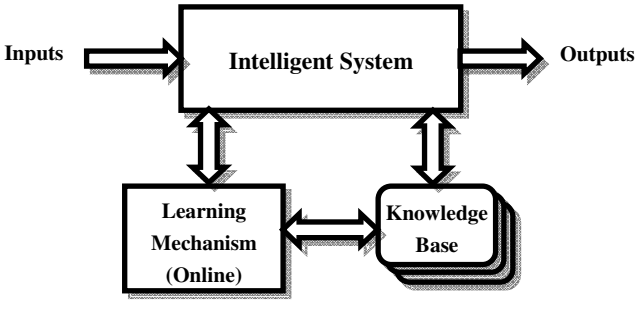


Fig. 1. Generic structure of evolving intelligent systems

The EIS paradigm emerged from studies whose objective were to discover new methods and techniques to transform existing intelligent systems, specially artificial neural networks, fuzzy systems, or the combination of both – neuro-fuzzy networks –, in systems whose structure were expansible (evolvable), allowing its adjustment to problems at hand, aiming better performance. The first published works were based on artificial neural networks or neuro-fuzzy networks [9]. In these models of evolving artificial neural networks, the network structure is flexible and changes during the evolution process. These early works on evolving intelligent systems were later improved, resulting in other models based on evolving artificial neural networks [10].

Papers published in the mid-2000s proposed intelligent systems based on evolving flexible rules models [11]. In such models, a flexible set of fuzzy rules performs the nonlinear mapping between inputs and outputs, using Mamdani or Takagi-Sugeno fuzzy models. A learning algorithm performs the online model identification, which means that both its structure (rule base) and its parameters are modified during the evolving process. In subsequent years, other works proposed improvements on intelligent systems based on evolving flexible rules models [12] and systems using new learning methods [13], [14], [15].

In recent years, many other works on EIS were published. However, despite the great developments achieved and several cases of successful application of these systems, there is still much demand for applications in various areas. Thus, EIS research field continues growing and it is not saturated [5].

3 Evolving Fuzzy Classifier Based on the Modified ECM Algorithm

One of the pattern recognition stages is classification. For this task, classification algorithms are used, among which may be highlighted the classification algorithms based on fuzzy rules, which have been applied to tasks such as decision making, fault diagnosis and image classification, due to their advantages over other classification algorithms [1]. This classifier type has good prediction performance and good transparency in form of linguistic rules that meaningfully represent the dependence between patterns' characteristics [16].

The architecture of a typical single-output fuzzy classifier consists on a set of fuzzy rules defined as [17]:

$$\text{Rule}_{\text{i}}: \ \text{IF } x_1 \text{ IS } \mu_{i1} \text{ AND } ... \text{ AND } x_p \text{ IS } \mu_{ip} \text{ THEN } y = L_i \,, \tag{1}$$

where p is the dimensionality of the input space, $\{x_1,...,x_p\}$ are the input variables/patterns, $\{\mu_{i1},...,\mu_{ip}\}$ are the fuzzy sets antecedents of the i-th fuzzy rule, y is output, L_i is the crisp output corresponding to a class label set $\{1,...,K\}$, with K being the number of classes. The classification of each new input x is achieved by assigning to it the corresponding class label to the rule of highest degree of activation, defined as:

$$y = L_{i^*}, \text{ with } i^* = \arg\max_{1 \le i \le R} \tau_i \tag{2}$$

where R is the number of fuzzy rules and τ is the i-th rule activation degree, defined by a t-norm, usually expressed as a product operator:

$$\tau_i = \prod_{j=1}^{p} \mu_{ij}\left(x_j\right) \tag{3}$$

where μ_{ij} are the membership functions for the fuzzy sets, defined by Gaussian functions:

$$\mu_{ij} = e^{-\frac{1}{2}\frac{\left(x_j - c_{ij}\right)^2}{\sigma_{ij}^2}} \tag{4}$$

where c_{ij} is the center and σ_{ij} the standard deviation of the membership functions.

The EIS can be applied in classification problems, through the automatic development of a fuzzy classifier from the input data acquired online and often in real time (measurement signals, data, images, etc.). In this case, it is necessary to employ an evolving clustering algorithm for unsupervised learning in order to create and to update the fuzzy rules online by means of partitioning the input space. This approach is different from that used in traditional fuzzy classifiers, which requires some type of off-line performed training (usually supervised).

Among the evolving clustering algorithms proposed in literature, the Evolving Clustering Method (ECM) has been applied in evolving fuzzy system model called Dynamic Evolving Neuro-Fuzzy Inference Systems (DENFIS) to create and to update all the fuzzy rules during a "one pass" training process [10]. The ECM is a fast online clustering algorithm based on maximum distance, which implements a partitioning of

the input space for purposes of creating and updating fuzzy inference rules. This algorithm doesn't use any optimization process to dynamically estimate the number of clusters in a data set and to find their centers in the input space. In any cluster, the maximum distance between its center and a data sample *MaxDist* is less than a threshold value *Dthr*, which should be set as a parameter and affects the number of clusters to be estimated. In the clustering process, samples are provided in an input data stream and, early in the process, the set of clusters is empty. When a new cluster is created, its center *Cc* is defined as the position of the current sample and its radius *Ru* is initially set to zero. As more data samples are provided, prior clusters are up-dated by changing its center positions and by increasing its radius.

In ECM, clusters centers are updated only when a new data sample that does not belong to any cluster is included in a particular cluster. If the new data sample already belongs to an existing group, no cluster center is updated. Thus, the clusters centers convergence deteriorates. In addition, in this algorithm the clusters are defined as hyperspherical regions, since the same radius is calculated for each direction/dimension. Thus, the input space partitioning is inflexible and the clusters representation can be incorrect.

Based on the ECM algorithm, a new Evolving Clustering Algorithm (ECA) is being proposed in this paper, in order to improve centers' convergence and to achieve better clusters representation. In ECA, the clusters' centers are always updated, either by new sample data that do not belong to an existing group or by new sample data that is included in a particular group. Groups are defined by hiperellipsoids regions, because different cluster radii are defined for each dimension. It is also important to note that in the ECA, both cluster centers and radii are calculated incrementally according to a learning rate that decreases in function of the data samples number in each group. The steps of the ECA are:

1. Create the first cluster C_1, taking the position of the first input data sample x_1 as the cluster center C_{C1}, and set to zero the center radius Ru_{1j} in each dimension, j=1, ...,p, where p is the dimensionality of the input space. Set the number of data samples of the first cluster nk_1 equal to 1 and set the learning rate η on the first center as the initial learning rate η_{init}, chosen in the interval [0, 1], usually defined as 0.5.
2. If all the input data samples were processed, the algorithm ends, otherwise, obtain the current data sample x_k and calculate the distance D_i between the sample and the centers of all n existing clusters centers Cc_i , where:

$$D_i = \left\| x_k - Cc_i \right\|, \quad i = 2,...,n \tag{5}$$

3. Find the cluster C_a from the all n existing clusters centers, choosing the cluster center Cc_a with the minimum distance:

$$D_{ia} = \min\left(D_i\right), \quad i = 2,...,n \tag{6}$$

4. If for at least one cluster C_i, i = 1,2, ..., n, the following condition is satisfied:

$$\sum_{j=1}^{p} \frac{\left(x_j - Cc_{ij}\right)^2}{Ru_{ij}^2} \leq 1, \; i = 1,...,n, \; j = 1,...,p \tag{7}$$

then, it means that the current data sample x_k belongs to an existing group. In this case, the cluster center Cc_a is updated as:

$$\begin{aligned} Cc_a^{new} &= Cc_a + n_a\left(x_k - Cc_a\right) \\ nk_a &= nk_a + 1 \\ \eta_a &= \frac{\eta_{init}}{nk_a} \end{aligned} \tag{8}$$

and the algorithm returns to step 2. Otherwise, the algorithm proceeds to the next step.

5. If the minimum distance D_{ia} is greater than $2\times D_{thr}$, the data sample x_k does not belong to any existing cluster. In this case, a new cluster must be created, as described in step 1, and the algorithm returns to step 2.
6. If the minimum distance D_{ia} is not greater than $2\times D_{thr}$, the sample data is included in an existing cluster. The cluster centers Cc_a and radii Ru_{aj} are updated as:

$$\begin{aligned} Cc_a^{new} &= Cc_a + n_a\left(x_k - Cc_a\right) \\ Ru_{aj}^{new} &= Ru_{aj} + n_a\left(x_{kj} - Cc_{aj}\right)^2, \;\; j = 1,\ldots, p \\ nk_a &= nk_a + 1 \\ \eta_a &= \frac{\eta_{init}}{nk_a} \end{aligned} \tag{9}$$

Return to step 2.

By using these procedures, ECA has the ability to extract from the input data set hiperellipsoids clusters in positions defined by centers obtained with higher precision, determining a partition of the input space that better represents the data set. An example of this effect is illustrated in Figure 2, in which an artificial data set with two clusters was partitioned employing the ECM and ECA algorithms. This result demonstrates the superiority of the ECA over the ECM algorithm.

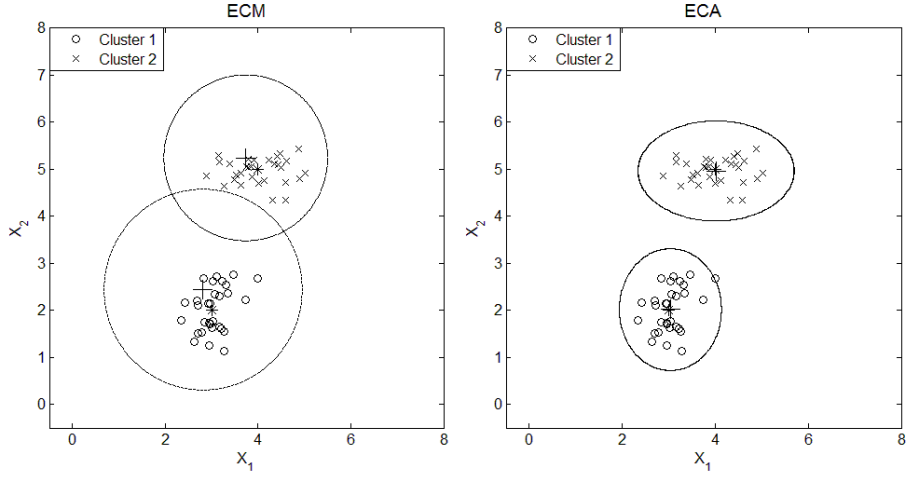


Fig. 2. Data clustering employing ECM and ECA algorithms

An Evolving Fuzzy Classifier (EFC) can be elaborated from the ECA for data pattern classification in online mode and potentially in real time. In the EFC, a fuzzy rule base is created and updated while the ECA performs partitioning the input data space, using the cluster centers and radii as parameters of the fuzzy rules antecedents (Eq. 1). To each new input data sample, the sample's class label is determined by the fuzzy rule with higher degree of activation (Eq. 2). An important feature of the EFC is that both the number of fuzzy rules as the number of classes is determined during the evolving process and does not need to be defined a priori. EFC steps are as follows:

1. Initialize the classifier setting the fuzzy rule number and the number of classes to 0.
2. Obtain a new input data sample and perform partitioning the input space using the ECA.
3. If a new cluster is created, then create a new fuzzy rule and set its antecedents as the parameters of this cluster. Otherwise, update the antecedents of fuzzy rule corresponding to the cluster to which the sample belongs or in which the data sample will be included.
4. Determine the class label of the data sample.
5. If all the input data samples were processed, the algorithm ends, otherwise, return to step 2.

4 Evaluation and Results

The proposed evolving fuzzy classifier was tested on three well-known benchmark problems. It should be noted that, despite the clearly off-line nature of these benchmark problems, they were used to test the proposed classifier in order to allow some comparison with other classification techniques. For the online mode simulation, input data were supplied "sample per sample" to the classifiers. The results obtained with evolving fuzzy classifier using both the ECM and the ECA algorithm were compared with those obtained by a classifier based on Multilayer Perceptron (MLP) Artificial Neural Networks (ANN) and a fuzzy classifier based on Fuzzy C-Means (FCM) algorithm.

The Iris data set has three classes that represent three types of Iris plants: Iris Setosa, Iris Verginica, and Iris Versicolor. The four attributes represent sepal and petal lengths and widths in centimeters. The data set consists of 50 samples for each species [18]. The test results for this data set are summarized in Table 1.

Table 1. Results for Iris problem

Classifier	Mode	Rules	Rate
ANN / MLP	off-line	-	97.7%
Fuzzy / FCM	off-line	3	89.3%
Evolving / ECM	online	3	88.0%
Evolving / ECA	online	3	92.7%

The Wine Recognition data set comes from the chemical analysis of wines grapes grown in Italy. There are three classes, 178 samples, with thirteen continuous numerical features [18]. The test results for this data set are summarized in Table 2.

Table 2. Results for Wine Recognition problem

Classifier	Mode	Rules	Rate
ANN / MLP	off-line	-	99,1%
Fuzzy / FCM	off-line	3	68.5%
Evolving / ECM	online	3	85.4%
Evolving / ECA	online	3	95.5%

The Wisconsin Breast Cancer data set contains information about clinical cases from University of Wisconsin Hospitals, reported by Dr. William Wolberg. There are ten attributes, and two classes: benign or malignant. The data set consists of 699 samples, with 458 for benign class and 251 for malignant class [18]. The test results for this data set are summarized in Table 3.

Table 3. Results for Wisconsin Breast Cancer problem

Classifier	Mode	Rules	Rate
ANN / MLP	off-line	-	97,6%
Fuzzy / FCM	off-line	2	60.2%
Evolving / ECM	online	2	90.4%
Evolving / ECA	online	2	96.3%

Comparing the results of the four classifiers, one can observe that the Evolving Fuzzy Classifier (EFC) with the ECA algorithm achieved a better performance than the other fuzzy classifiers, but worse than those obtained by the classifier with ANN. However, one should take into account that the EFC performs the classification in online mode, while the ANN performs classification in off-line mode. This means that the EFC begins with an empty rule base and evolves it from the input data stream, without prior training. This feature is essential in applications where the learning must be performed online and in real-time and difference in accuracy (rate classification) between the EFC and the ANN may be acceptable in most of these applications.

5 Conclusions and Future Work

In this paper, an Evolving Fuzzy Classifier (EFC), whose main feature is to perform pattern classification online and possibly in real-time, was proposed. In the EFC, a new Evolving Clustering Algorithm (ECA) based on Evolving Clustering Method (ECM) algorithm was employed. The ECA algorithm has more efficient mechanisms for updating the clusters centers and radii, improving the classifier performance. Experiments with well-known benchmark problems (Iris, Wine Recognition and

Wisconsin Breast Cancer data sets) have been performed and the results indicate that EFC achieve classification rates comparable to those obtained by off-line pre-training classifiers. Thus, the proposed EFC is suitable for online and real-time applications, such as classification of streaming data patterns or fault diagnostics, with the advantage of allowing integration of new data to the existing knowledge base. In future work, other procedures, such as an optimization strategy to determine the number of rules and an automatic adjusting strategy for the distance threshold will be investigated.

Acknowledgments. This work was supported by FAPEMIG, CAPES, CNPq and FACIT.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, New York (2000)
2. Jain, A.K.: Data clustering: 50years beyond K-means. Pattern Recognition Letters 31(8), 651–666 (2010)
3. Kasabov, N., Filev, D.: Evolving Intelligent Systems: Methods, Learning, & Applications. In: 2006 International Symposium on Evolving Fuzzy Systems, pp. 8–18 (2006)
4. Angelov, P., Kasabov, N.: Evolving Intelligent Systems, eIS. IEEE Systems, SMC News Letter 15, 1–13 (2006)
5. Angelov, P., Filev, D., Kasabov, E. (eds.): Evolving Intelligent Systems: Methodology and Applications. IEEE Press Series on Computational Intelligence, 484 p. John Willey and Sons (2010)
6. Kasabov, N., Song, Q.: DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction. IEEE Trans. on Fuzzy Systems 10(2), 144–154 (2002)
7. Aström, K., Wittenmark, B.: Adaptive Control, 2nd edn. Addison Wesley, Reading (1995)
8. Abraham, A., Grosan, C.: Engineering Evolutionary Intelligent Systems: Methodologies, Architectures and Reviews. In: Engineering Evolutionary Intelligent Systems. SCI, pp. 1–22. Springer, Germany (2008)
9. Kasabov, N., Song, Q.: Dynamic Evolving Fuzzy Neural Networks with'm-out-of-n' Activation Nodes for On-line Adaptive Systems. Technical Report TR99/04, Department of Information Science, University of Otago (1999)
10. Kasabov, N., Song, Q.: DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction. IEEE Trans. on Fuzzy Systems 10(2), 144–154 (2002)
11. Angelov, P.: Evolving rule- based models: a Tool for Design of Flexible Adaptive Systems. Springer, London (2002)
12. Angelov, P., Filev, D.: Simpl_ets: as implified method for learning evolving takagi-sugeno fuzzy models. In: The14th IEEE International Conference on Fuzzy Systems, FUZZ 2005, pp. 1068–1073 (2005)
13. Leng, G., McGinnity, T.M., Prasad, G.: An approach for on-line extraction of fuzzy rules using a self-organizing fuzzy neural network. Fuzzy Sets & Systems 150, 211–243 (2005)
14. Lughofer, E.: FLEXFIS: A Robust Incremental Learning Approach for Evolving Takagi-Sugeno Fuzzy Models. IEEE Transactions on Fuzzy Systems 16(6), 1393–1410 (2008)

15. Lima, E., Hell, M., Ballini, R., Gomide, F.: Evolving fuzzy modeling using participatory learning. In: Angelov, P., Filev, D., Kasabov, N. (eds.) Evolving Intelligent Systems: Methodology and Applications. John Willey and Sons (2010)
16. Angelov, P., Lughofer, E., Zhou, X.: Evolving fuzzy classifiers using different model architectures. Fuzzy Sets and Systems 159(23), 3160–3182 (2008)
17. Lughofer, E.: Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications. STUDFUZZ, vol. 266. Springer, Heidelberg (2011)
18. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets.html

Overhead-Controlled Routing in WSNs with Reinforcement Learning

Leonardo R.S. Campos, Rodrigo D. Oliveira,
Jorge D. Melo, and Adrião D. Dória Neto

Departamento de Engenharia de Computação e Automação,
Universidade Federal do Rio Grande do Norte (UFRN)
59.078-900, Natal, RN, Brazil
{leofields,rodrigodutra,jdmelo,adriao}@dca.ufrn.br

Abstract. The use of wireless sensor networks in industry has been increased past few years, bringing multiple benefits compared to wired systems, like network flexibility and manageability. Such networks consist of a possibly large number of small and autonomous sensor and actuator devices with wireless communication capabilities. The data collected by sensors are sent — directly or through intermediary nodes along the network — to a base station called sink node. The data routing in this environment is an essential matter since it is strictly bounded to the energy efficiency, thus the network lifetime. This work investigates the application of a routing technique based on reinforcement learning's Q-learning algorithm to a wireless sensor network by using an NS-2 simulated environment. Several metrics like routing overhead, data packet delivery rates and delays are used to validate the proposal comparing it with another solutions existing in the literature.

Keywords: wireless sensor networks, routing, reinforcement learning.

1 Introduction

Wireless Sensor Networks (WSNs) have gained much attention over past few years by both industries and researches around the world because of its various features, like scalability, self-organization, low complexity, cost and size of nodes, being applicable to multiple scenarios, including many industry ones. A WSN consists of a possible large number of small and autonomous sensor devices with wireless communication capabilities whose objective is to deliver the data collected from the environment to a special device called sink [1]. Like any other network, they need a mechanism to forward data packets among nodes: the routing protocol. Since WSNs have energy restrictions concerning battery-powered nodes, prolonging the autonomous lifetime of the network is a challenging task that must be accomplished mostly by designing energy-efficient routing protocols [2].

Primary routing techniques were designed to optimize only one goal at time such as routing path length, load balance or link reliability. Adaptive approaches

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 622–629, 2012.

using reinforcement learning (RL) emerged to provide a way to use several optimization objectives into a single function.

A major concern when designing routing schemes is the overhead imposed by the protocol control packets to the network data traffic. Performance studies [3,4] have shown that most routing protocols barely achieve the minimum rate of one control packet per actual data packet in the network. In this paper we present an improvement to the RL-based routing with Q-learning [5] that decreases the routing overhead maintaining all benefits of this routing technique.

In the following, Section 2 reviews related work. Section 3 provides a basic background on general reinforcement learning concepts and how to turn the routing problem into a RL problem. Implementation details are presented in Section 4, followed by the results gathered during our experiments in Section 5. For last, Section 6 ends with conclusions and future work.

2 Related Work

The application of reinforcement learning to routing in networks was firstly introduced by Boyan and Litman [6], who used Q-learning in static packet switched network. They have shown that the Q-learning based routing approach is able to compete with the shortest path algorithms, without prior knowledge regarding the network topology. Following their prior work, Q-learning is applied to routing in ad-hoc networks [7] and optimizing the trade-off between routing and compression [8]. The AdaR protocol [9] uses a Least Squares Policy Iteration (LSPI) algorithm that is shown to be superior to the basic Q-learning in some aspects, although nothing is discussed about the routing overhead performance.

There are many others non-RL based routing protocols worth mentioning. One of the most widely deployed is ad hoc on-demand distance vector routing (AODV) [10], that was firstly applied in mobile ad hoc networks (MANETs) and then ported to be used in WSNs. Directed diffusion [11] is a data-centric and application-aware paradigm that aims at diffusing data through sensor nodes by using a naming scheme for the data gathered by the sensors.

3 Theoretical Basis

3.1 Reinforcement Learning Background

Reinforcement learning (RL) is a paradigm of machine learning applicable to Markov Decision Processes (MDP) where an agent can learn control policies based on experience and rewards [12]. A MDP models an agent acting in an environment with a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is a set of actions. $P(s'|s,a)$ is the transition model that describes the probability of entering state s' after executing action a at state s. $\mathcal{R}(s,a,s')$ is the reward obtained when the agent executes a at s and enters s'. Within this framework, the quality of action a at state s can be denoted by a Q-value $Q(s,a)$. The goal of solving an MDP is to find an optimal policy, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, that maps states

to actions such that the cumulative reward is maximized, which is equivalent to finding the following Q-fixed points (Bellman equation):

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a) \max_{a'} Q^*(s',a') \ , \tag{1}$$

where the first term on the right side represents the expected immediate reward of executing action a at s, which is calculated as $r(s,a) = \sum_{s'} \mathcal{P}(s'|s,a)\mathcal{R}(s,a,s')$, and the second term is the maximum expected future reward. The discount factor γ models the fact that immediate reward is more valuable than future reward.

3.2 RL-Based Routing with Q-Learning

In the context of learning routing strategy, each sensor can be considered as a state s, and for each of its neighbors s', there is a corresponding action a ($\mathcal{P}(s'|s,a) = 1$). Executing action a at s means forwarding the packet to the corresponding neighbor s'. The fixed Q-values of s can be regarded as its routing table, telling to which neighbor forward the data. With this routing table's aid, the optimal routing path can be trivially built from a sequence of table look-up operations. Thus, the task of learning optimal routing strategy is equivalent to finding the Q-fixed points (1).

If the underlying transition model $\mathcal{P}$ and the reward mechanism $\mathcal{R}$ are known, the Q-fixed points can be solved deterministically. In most cases, such as WSNs however, it is fairly hard to determine the underlying model, which calls for stochastic reinforcement learning techniques. Q-learning is an off-policy temporal difference (TD) control algorithm that directly approximates the optimal action-value function (1). Each time an action a is executed, the agent receives an immediate reward and the expected long term reward to update the Q-values, which in turn influences future action selection. Its simplest form, one-step Q-learning is defined as:

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') \right] \ , \tag{2}$$

where α is the learning rate, which models the rate of updating Q-values.

As a model-free RL technique, Q-learning requires no knowledge about the underlying reward or transition mechanisms, thus applicable to the problem of learning routing strategy in sensor networks.

4 Implementation Details

4.1 Basic Q-Routing

The use of reinforcement learning in packet routing has the objective to indicate which neighbor the packet will be sent to from where it is. The policy will be evaluated accordingly the total packet delivery time. There is no parameter to evaluate or improve the policy until a packet reaches its destiny. However, by

using reinforcement learning, the policy can be updated faster using only local information.

Let $Q_x(d, y)$ be the time estimated by a node x to deliver a packet to the sink d (including any time this packet may spend in x nodes queue) through y, where $y \in \mathcal{N}_x$ and $\mathcal{N}_x$ represents the set of all neighbors of x. This scheme is depicted in Figure 1. By sending the packet to y, x immediately obtain the estimative t of the time that will be spent from y to the rest of the path until d:

$$t = \min_{z \in N_y} Q_y(d, z) \; . \tag{3}$$

If a packet spends q units of time in x node's queue and s units of time in the transmission from x to y, then x can update its estimative as follows:

$$\Delta Q_x(d, y) = \alpha \left(q + s - t + Q_x(d, y) \right) \; , \tag{4}$$

where α is the learning rate ranging between 0 and 1. Equations (3) and (4) combined will form the general Q-learning equation (2), where the maximization is replaced by a minimization, that is:

$$Q_x(d, y) = (1 + \alpha) Q_x(d, y) + \alpha \left[r - \min_{z \in N_y} Q_y(d, z) \right] \; , \tag{5}$$

where $r = q + s$ is the immediate reward obtained by sending the packet to y.

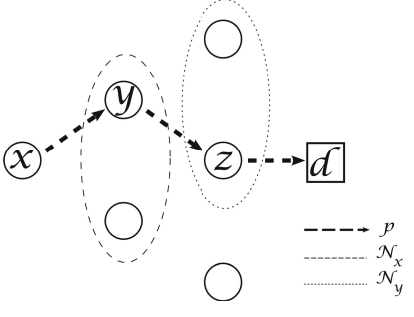


Fig. 1. Organization of Q-routing scenario

4.2 Improved Q-Routing

In practice, to obtain the immediate reward and the estimated future reward is required a feedback packet sent from the neighbor who has just received a packet. Therefore, the protocol would introduce an overhead from at least one routing packet per actual data packet. In order to reduce this overhead, we managed a solution using a packet accumulation and averaging scheme, described as follows.

Each data packet contains a timestamp in its common header set when it is created. When receiving a data packet, a node can use its current timestamp

Table 1. Simulation parameters

Hardware Configuration	
Communication Channel	Channel/WirelessChannel
Propagation Model	Propagation/TwoRayGround
Queue type	Queue/DropTail/PriQueue
Queue size	50
Physical layer	Phy/WirelessPhy/802_15_4
MAC layer	Phy/WirelessPhy/802_15_4
CSThresh	6.25e-17
RXThresh	36.25e-17
Frequency	2.4GHz

Topology Parameters	
Terrain Area	500m x 500m
Topology	Mesh
Number of devices	100 (sensors) + 1 (sink)
Routing algorithm	AODV, Q-routing, Improved Q-routing
Data Traffic	
Traffic type	Constant Bit Rate (CBR)
Data rates	0.1, 0.2, 1, 5, 10 packets per second (pps)
Packet size	90 bytes
Simulation time	1000 s

to calculate $(q + s)$. Each node keeps a table containing the number of packets p received from each other and the average receiving time $avg(q + s)$ as well, updating both for each new data packet received. The average receiving time is updated by the following expression:

$$\frac{p[avg(q+s)](q+s)}{p+1} \,. \tag{6}$$

Every node also defines a timer which expires periodically, causing the node to send a control packet to all neighbors in its table whose $p > 0$. This only routing packet defined by this solution contains the average receiving time calculated and the best estimative from the remaining time to destination used to update nodes estimative.

The expire time λ is set as a parameter from the algorithm along with the learning rate α and the exploration rate ϵ. The last one is used to speedup exploration through a ϵ-greedy policy, i.e., with probability $1 - \epsilon$, one takes action with the best estimative, and with probability ϵ one chooses an action at random.

The behavior of λ according the data traffic on the network is crucial for decreasing the routing overhead. In fact, if x is the average packet rate, a function $\lambda = f(x) = a/x$ can be defined, where a is a constant which represents the decreasing factor of the routing overhead.

5 Experiments, Results and Discussion

The proposed algorithm and evaluating experiments were implemented in the widely recognized network simulator NS-2 version 2.34. The Q-routing and its improved version were included in the simulator as new routing protocols and tested in a WSN scenario. AODV routing already in the simulator was selected for comparison. The following Table 1 presents the general configuration of the experiments.

5.1 Number of Experiments

The number of simulations executed was statistically determined by using the standard deviation of the packet delivery ratio. This metric represents the ratio

between the number of packets received and the number of packets sent. As depicted by Figure 2, from 25 simulations on, the changes on standard deviation are negligible.

5.2 Convergence

Although Q-learning algorithm is proved to be convergent since the policy guarantees visiting all possible states (satisfied by ϵ-greedy policy), the convergence of proposed modification cares to be evaluated.

As Figure 3 shows, about 40 episodes are necessary to achieve convergence in both solutions. We can also see that the modification in the original Q-learning doesnt affect the convergence speed.

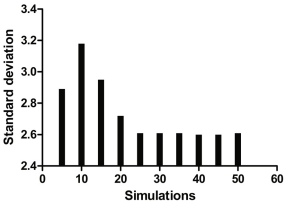


Fig. 2. Statistical determination of the number of experiments

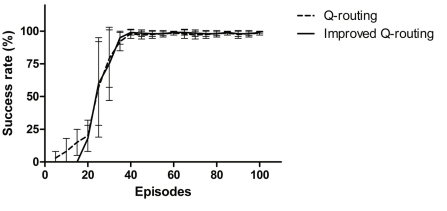


Fig. 3. Success rate of routing by number of episodes

5.3 Packet Delivery Ratio

The packet delivery ratio (PDR) measures the amount of packets successfully delivered to its destination divided by the total number of packets sent. This metric is important to show how effective the network is. The performance of the three considered protocols relative to this metric is shown at Figure 4.

We can see that AODV decreases its performance slowly but continuously as the traffic load grows, while Q-routing and its improved version present similar performance, fluctuating around 90%. This behavior of AODV can be explained by the higher overhead imposed by the several control packets used in this protocol.

5.4 Average End-to-End Delay

The transaction time of passing a packet to a one-hop neighbor, including time of all necessary processing, backoff as well as transmission, and averaged over all successful end-to-end transmissions within a simulation run.

Again, as seen in Fig. 5, reinforcement learning based techniques show similar results, with average end-to-end delay around 100 ms, while AODV increases latency time until reach more than 200 ms at the rate of 10 pps.

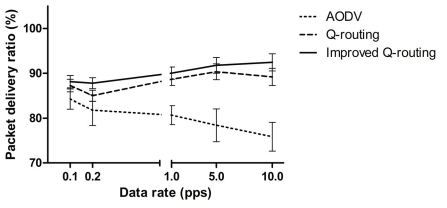


Fig. 4. Packet delivery ratio of analyzed protocols

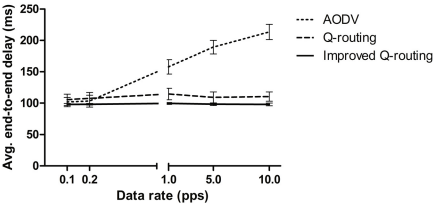


Fig. 5. End-to-end delay averaged in all successful transmissions

5.5 Routing Overhead

This metric is defined as the relation between total data packets and total control packets sent. As we observe in Figure 6, all the Q-learning based techniques have a fixed value for the overhead regardless the data rate.

Furthermore, by controlling the accumulation of data packets through the constant a, it is shown the significant reduction of the overhead.

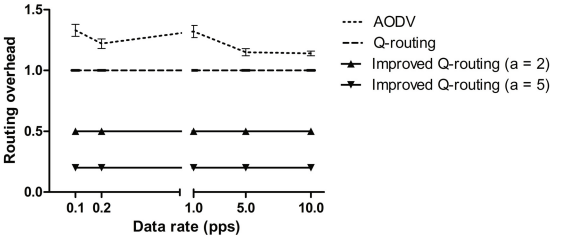


Fig. 6. Routing overhead of analyzed solutions

6 Conclusion

In this paper we discussed an improvement to the reinforcement learning based routing algorithm for WSNs that decreases the routing protocol overhead. Results have shown that the proposed technique maintains the performance of other existing protocols regarding another metrics like packet delivery ratio and average end-to-end delay. The technique also has the advantage of using the own data packets to build the solution, instead of creating several control packets as other protocols.

References

1. Verdone, R., Dardari, D., Mazzini, G., Conti, A.: Wireless Sensor and Actuator Networks: Technologies, Analysis and Design. Academic Press (2008)

2. Al-karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: A survey. IEEE Wireless Communications 11(6), 6–28 (2004)
3. Zheng, J., Lee, M.J.: A comprehensive performance study of ieee 802.15.4. IEEE Press, New York (2004)
4. Dwivedi, A.K., Kushwaha, S., Vyas, O.P.: Performance of Routing Protocols for Mobile Adhoc and Wireless Sensor Networks: A Comparative Study. International Journal of Recent Trends in Engineering 2(4), 101–105 (2009)
5. Watkins, C., Dayan, P.: Q-learning. Machine Learning, 279–292 (1992)
6. Boyan, J.A., Litman, M.L.: Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach. In: Cowan, J., Tesauro, G., Alspector, J. (eds.) Advances in Neural Information Processing Systems, pp. 671–678. Morgan Kaufmann (1994)
7. Silva, D.R.C., Doria, A.D., Melo, J.D.: Uso do Algoritmo Q-Learning para o Roteamento em Redes Ad-hoc. In: Simpósio Brasileiro de Redes Neurais, São Luiz. Anais do VIII SBRN (2005)
8. Beyens, K.S.P., Peeters, M., Nowe, A.: Routing with compression in wireless sensor networks: a q-learning approach. In: Proc. of AAMAS (2005)
9. Wang, P., Wang, T.: Adaptive Routing for Sensor Networks using Reinforcement Learning. In: Proc. of IEEE ICCIT, p. 219 (2006)
10. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proc. of IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100 (1999)
11. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proc. of ACM MobiCom (2000)
12. Sutton, R., Barto, A.: Reinforcement Learning: An introduction. MIT Press, Cambridge (1992)

Metaheuristic GRASP with Path-Relinking to the Solution of the Graph Partitioning Problem with Capacity and Connexity Constraints

Nádia Mendes Santos, Gustavo Silva Semaan, and Luiz Satoru Ochi

Instituto de Computação - Universidade Federal Fluminense,
Passo da Pátria Street 156, Bloco E – 3° andar, São Domingos,
CEP.: 24210-240 Niterói, RJ, Brasil
nadiaphb@gmail.com, {gsemaan,satoru}@ic.uff.br

Abstract. This work proposes new hybrids algorithms based in the Greedy Randomized Adaptative Search Procedure (GRASP) metaheuristic and Path-Relinking (PR) procedure for the solution of the Clustering Problem with Capacity and Connectedness Constraints. Computational results show the proposed methods to be competitive in relation to instances in the literature and to existing techniques.

Keywords: GRASP, Path-Relinking, Clustering Problem, Graph Partitioning Problem.

1 Introduction

In general, obtaining the solution to a problem of clustering corresponds to grouping the elements of a database in such a way the groups formed, or clusters, represent a configuration in which each element has a higher similarity with any element of the same cluster rather than with other clusters elements. More specifically, it deals with the problem of clustering applied to graphs, known as graph partitioning problem, which is a problem in class Complete-NP.

Formally, the problem can be described as, given a set of n objects $X=\{x_1, x_2, ..., x_n\}$ in which each object x_i has p attributes $x_i = \{x_i^1, x_i^2,...,x_i^n\}$ that measure the characteristics of the object, partitions k and C_i clusters must be constructed from the set X, in compliance with the following restrictions [Baum, 1986][Hruschka and Ebechen, 2001][Dias and Ochi, 2003]:

$$C_i \neq \emptyset \text{ for } i = 1,...,k \quad (1)$$

$$C_i \cap C_j \neq \emptyset \text{ for } i, j = 1,...,k \text{ and } i \neq j \quad (2)$$

$$\bigcup_i^k C_i = X \quad (3)$$

The value of k can be known or not previously. If the value of k is given as a parameter to the solution, the problem is referred to in the literature as k-clustering problem

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 630–641, 2012.

or Clustering Problem (PC) [Fasulo, 1999]. The complexity of this problem stems from the large number of alternative solutions that exist, thereby hindering, to obtain an optimal solution or even form solutions of good quality. The amount of solutions on the PC can be determined by the equation (4) [Cole, 1998]:

$$N(n,k) = \sum_{k=1}^{n} \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n \quad (4)$$

This work is divided into four sections. In Section 2 we have the modeling of the problem (APOND and APAs), where the case study of this work is described in details. In Section 3 we report the GRASP metaheuristic and GRASP with Path-Relinking algorithm. It is concluded with the computational results obtained through the implementation of metaheuristics.

2 Modeling the Problem

A reasonable amount of real applications, as the geographical regionalization and the division of telecommunications network, can be mapped in a problem of clustering with spatial connectivity constraints. In this type of problem, we aim to partition a set of n objects that comprise a database in cluster k which are internally homogeneous and further objects in the same cluster are connected to the associated graph. This assembly problem appears, for example, in situations where the geographical regionalization is necessary, i.e., in case of censuses and socio-economic survey [Openshaw, 1977].

In the context of the geographic census, for example, the objects can be linked to households, to census tracts, to area weighting and to municipalities. Considering the particular case of the weighting areas, for the formation of clusters are also considered the connectivity constraints, so that these areas are comprised of geographically neighboring regions and homogeneity, according to set of p attributes linked to the characteristics of population and infrastructure known as indicators of weighting areas.

This section discusses in detail a particular real application of regionalization know as APAs formation problem. For a better understanding of the problem the APONDs, the APAs and the case study of this work will be introduced, successively.

2.1 APOND and APA

The APOND (area weigh) corresponds to a geographic unit formed by mutually exclusive groupings of census tracts, which are, in turn formed by groups of households. These areas are used to estimate information to the public. Thus, the size of these areas, in terms of number of household and/or population, can't be greatly reduced under penalty of loss of precision of estimates. The APONDs also represent the most detailed geographical level of the operating base, being defined in order to meet the demands for information on smaller geographic levels than municipalities [Censo Demográfico, 2000][Silva *et. al*, 2004]. For the formation of APONDs are still

considered criteria of contiguity, so that these areas are formed by p series of variables associated with changes in population and the known infrastructure. These variables we represent by x^s, $s=1, ..., p$ are called indicators of weighting areas. For the formulation of the problem treated in this work, it's possible to associate the information on the proximity of the regions, and the information of the totals associated with each attribute p and the distance $d_{ij,}$ the graph $G= (V, E)$. Each vertex $i \in V$ of the graph corresponding to a region and its attributes, including the capacity, used to validate the capacity restriction.

The Equation (5) describes d_{ij} representing the degree of homogeneity between the aggregate to be APONDs where i and j neighboring APONDs:

$$d_{ij} = \sqrt{\sum_{s=1}^{p} \left(x_i{}^s - x_j{}^s\right)^2} \tag{5}$$

The APAs (areas weighted aggregate) are formed by groups of mutually exclusive APONDs, respecting two criteria of creation feasibility:

- Contiguity: the APONDs aggregated on each one of the APAs must be neighboring or must be able to leave an APOND A and arrive at an APOND B, both in the same APA, passing by only other APONDs that are also in the same APA;
- Total associated to one of the variables: provided a variable, the population of each area, for example, the sum of this variable in each APA must meet a predetermined limit.

2.2 Case Study

Based on the previous sections information, it can be seen that the problem of formation of aggregate weight areas can be mapped on a clustering graph problem with the restrictions of connectedness (contiguity) and capacity (total given variable).

So, it's possible to associate both the information relating to contiguity (connectedness) of APONDs and the information capacity of the $G = (V, E)$. Each vertex $i \in V$ of the graph corresponds to an APOND and contains the value associated with the variable that determines its capacity. Furthermore, if two APONDs i and j are neighboring, there is an edge $e = (i, j)$ which contains the value of the distance d_{ij}. Considering the connectedness restriction of the graph G, a natural solution to the problem consists of constructing a minimal tree generator $T = (V, E^* \subset E)$ obtained from G, considering lower values of d_{ij}. Provided the tree T and a number of clusters k (APAs numbers to be formed), $(k - 1)$ edges of T are removed, defining a set of k subtrees T_j, $j = 1,... , k$, which are also connected. Each of these subtrees corresponds to a possible APA.

The property of connectedness, observed for each one of the subtrees makes possible the immediate execution of the contiguity restriction at each one of the APAs. So, one possible approach to solving the problem consists on partitioning T in k subtrees T_j, $j = 1,..., k$, associated with APAs that satisfy the capacity constraint and resulting the smallest possible value of a particular objective function which measures the degree of similarity in APAs. In the case of partitioning, to assess the quality of APAs

obtained, considering the capacity and contiguity criteria are respected, the authors proposed the following objective function described by the Equation (6):

$$f = \sum_{j=1}^{m}\sum_{i=1}^{m}(x_{ij} + \bar{x}_j)^2, \text{ in which } x_j \sum_{i=1}^{n}\frac{x_{ij}}{n} \text{ to } m \text{ variable, } n \text{ areas} \quad (6)$$

For the formation of two new groups (APAs), the value of *f* is evaluated, considering the removing of each one of the edges associated with the previous groups. The lower the value of *f,* the better the quality of the solution.

The solutions proposed in this paper refer to the construction of Minimum Spanning Tree *T* (AGM) through the application of Kruskal algorithm. The advantage of using a AGM is that the spatial adjacency (contiguity) required by the problem is met normally, once a tree represents a connected graph and any edge removed from it produces two subtrees which are also related. This tree is constructed from a graph *G* that contains the information of APONDs. Then, to generate the *k* APAs (clusters) is necessary to remove *(k – 1)* edges of *T*. In other words, in each iteration *j (j=1,..., k – 1)*, the edge e_j is removed in a subtree T_{j-1}, producing two new subtrees (two new clusters) T_j^1 and T_j^2 *(j=1,..., k – 1).* This procedure corresponds to a strategy of hierarchical division in which initially all APONDs belong to one cluster.

The criterion for selection of the subtree (cluster) was to be partitioned based on the highest value obtained in Equation 6, ie, the each iteration selects the subtree (cluster) which contains the mean quadratic deviation of the nodes (APONDs) that comprises the cluster, considering the variables of *p* APONDs.

Once selected subtree, the choice of the edge to be removed consists of a greedy procedure. Therefore, the possibility of removing edges in the selected cluster are performed, aiming at the minimization of the objective function $Cost_e$ in Equation 7.

$$Cost_e = f(T_i) - (f(T_i^1) + f(T_i^2)) \quad (7)$$

The value of the cost of removing the edge ($Cost_e$) corresponds to the difference between the objective function of the current cluster $f(T_i)$ and a sum of the objective function of each one of the clusters $(f(T_i^1) + f(T_i^2))$. The more homogeneous the two new clusters are, the greater the value of $Cost_e$.[Semaan, 2010].

In the next Section, is the description of metaheuristics and procedures used in this study.

3 Metaheuristics and Procedures Implemented

The metaheuristics are generic heuristics that have gained popularity in recent decades and represent a family of optimization techniques that get great solutions in the elucidation of various problems of high computational complexity (NP-Complete and NP-Hard problems). Some metaheuristics can be seen as a simply repetition of two simple steps: (Ibaraki, et.al., 2005) generating solutions and improvement of solutions through local search.

In the scientific context of this work, a metaheuristic this class: the GRASP, is of special importance for good results. Another procedure that can be coupled with the

very successful metaheuristics is the Path-Relinking. Both will be detailed following: the GRASP and the Path-Relinking.

3.1 GRASP

GRASP (*Greedy Randomized Adaptative Search Procedure*) proposed by Feo & Resende in 1995, is a metaheristic that has been used successfully in solving combinatorial optimization problems in several areas [Silva & Ochi, 2012] [Silva et al., 2012], [Ferreira & Ochi, 2012]. It's in an iterative sequence metaheuristic, where each interaction consists of two phases: the constructive phase (Figura1- ConstrutionAlgorithm) and the local search phase (Figure1- LocalSearchAlgorithm). After each interaction, one local optimal is found and the best solution of all iterations is returned as the final solution. The GRASP combines the greedy and random approaches. The construction phase builds a feasible solution by using a function of randomness greedy. The greedy part of the function aims to generate a feasible better cost solution (low or high cost, depending on application). The random component is included to explore various regions of the solution space and is a key to effectiveness of the GRASP. The best optimal local among all local searches is returned as a solution of the metaheuristic [Feo & Resende, 1995]. Pseudocode GRASP in the Figure 1, next:

```
(1) BestSolution ← 0
(2) repeat
(3) InitialSolution ← ConstructionAlgorithm
(4) GoodSolution ← LocalSearchAlgorithm (InitialSolution)
(5) until stopping criterion satisfied
(6) return BestSolution
```

Fig. 1. Pseudocode of the GRASP

3.2 Path-Relinking

The Path-Relinking procedure was initially proposed to compose the diversification phase of metaheuristic Scatter Search [Glover and Laguna, 1997]. And has the strategy to generate new solutions to explore the path that connects solutions of good quality. The Path-Relinking starts at the choice of a solution called initial or base solution, and later it is built a way through the solution space that connects to another extreme solution called Solution Guide, thus leading to new paths called intermediate solutions to each movement performed. For the movement is performed are inserted attributes or characteristics of the Solution Guide to Intermediate Solution [Glover and Laguna, 1997].

The Path-Relinking has the objective to find good quality solutions, known as elite solutions, which have characteristics to be shared. And through a series of solutions generated, which derive from then, can find a neighboring solution better than the ones known, in other words, they can find a great place never before explored and probably a global optimal or a local optimal of the best quality [Ho and Gendreau, 2006].

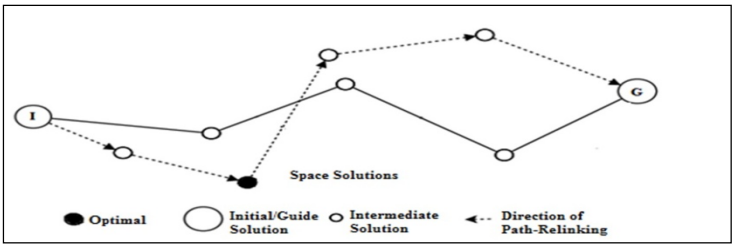


Fig. 2. Adaptation of Path-Relinking [Andrade, *et. al*, 2009]

The Figure 2 illustrated the solution space between the Initial Solution (I) and the Guide Solution (G), in which the solid line indicates the intermediate solutions found in movements in the common local search. Since the dashed line indicates the path created by the Path-Relinking, it was possible to find a better intermediary solution. The arrows indicate the path, starting from Initial Solution (I) to Guide Solution (G) showing the solution space noticeably larger.

3.3 GRASP with Path-Relinking

GRASP with Path-Relinking procedure is a hybrid algorithm which has produced excellent results when applied to various problems of high computational complexity [Silva & Ochi 2010], [Penna et. al., 2012]. The hybrid version proposed here uses the same implementations the construction phase and local search phase used by GRASP in its standard form, and no change were needed in these structures. Path-Relinking used as a strategy of backward optimization technique, where the Path-Relinking is applied from the best final solution (Initial Solution) to the other extreme solution (Solution Guide). And as intensification strategy, being applied between each solution generated by the local search phase and a selected solution from the elite set [Andrade, 2009].

The pseudocode of the GRASP with Path-Relinking is showed in Figure 3, next [Frinhani, 2011]:

```
(1) BestSolution ← 0;
(2) repeat
(3)  InitialSolution ← ConstructionAlgorithm;
(4)  GoodSolution ← LocalSearchAlgorithm(InitialSolution);
(5)  IntermediateSolution ←Path-Relinking(GoodSolution, EliteSolution);
(6)  if (IntermediateSolution > GoodSolution)
(7)     BestSolution = IntermediateSolution;
(8)  else
(9)     BestSolution = GoodSolution;
(5) until stopping criterion satisfied;
(6) return BestSolution;
```

Fig. 3. Pseudocode GRASP with Path-Relinking

The Construction Algorithms used in this work in both the GRASP and GRASP with Path-Relinking generate clusters by the partitioning of an AGM *T*, obtained from the graph G associated with the problem, in subtree *k* (clusters). In other words, the *k* clusters are defined considering the removal of exactly *k – 1* edges of *T*, and the removal of each new edge produces two subtrees which shall correspond to that defined by two new clusters T_i^1 and T_i^2.

Thus, to the ideal of partitioning the AGM were implemented three versions of the constructor procedures of initial solution which will be described in the next section.

3.4 Procedures Implemented

The procedures developed in this work which deals with the representation of the solution, of the constructors of the initial solutions and local search procedures are described below.

3.4.1 Representation of the Solution

Before the application of the algorithm is necessary to define a structure for representing the solutions. The good representation is extremely important, being able to be a determinative factor in the performance of the algorithm, that is, for fast convergence and quality of the solutions obtained by the same.

The representation of the solution structure adopted in this study was the group-number. In this structure each index vector represents the vertice of the graph and its contents, a positive integer between *1* and *k*, represents the cluster which belongs to the vertice [Trindade and Ochi, 2006][Dias, 2004]. Still representing the solution, the algorithms developed in this work allow this value to be entered as input parameter or the algorithm itself calculate the capacitance value by using Equation 8. In this Equation, |V| represents the amount of objects to be grouped, x_{il} is the value of the *i*-th attribute associated with the *i*-th object, *k* is the number of clusters and β is the adjustment factor:

$$Capacity_{Min} = (\beta / k).\sum_{i=1}^{|V|} x_{il} \quad (8)$$

Besides this vector, the data structure used to represent the solutions also contains an attribute relating to the capacity of the cluster, obtained by summing the attribute associated with the ability of the vertices of the cluster. So, it's possible to verify which clusters are penalized, in other words, having a value below the predetermined limit and, consequently, the solution has many penalties [Semaan, 2010].

3.4.2 Construction of Initial Solution

Three versions for construction of the initial solutions had been used: construction procedure 1, construction procedure 2 and construction procedure 3.

The construction procedure 1 consists, basically, of two steps, executed *k – 1* times to obtain *k* clusters: selecting the cluster to be partitioned and defining the edge to be removed. The objective to enable the diversification of solutions, this procedure was adapted for not having a greedy behavior. Thus, the algorithm is able to obtain solutions potentially interesting, but would be ignored by the greedy procedure. Using the idea of the GRASP construction procedure, instead of selecting

always the edge of bigger cost, opted for the construction of a Restricted Candidates List (RCL) of the GRASP. One of the edges of RLC is randomly selected and removed from the corresponding cluster, forming two new clusters.

The second version of the construction procedure works in an attempt to form cluster that meet the minimal restriction capacity. The first step is the definition of two vertices as being extremity from the main tree (A and B vertices). Next, it determines if the vertices that will belong to this main tree, i.e., the vertices are on the path between A and B existing in the AGM, by applying of the Equation 9, in the distance between the vertices *s* and *t* as the existing path AGM. The other vertices will be grouped to vertice belonging to the main tree closer to them. In this way, the graph will be similar to a "cordon" and the vertices belonging to the main tree will possess the sum of the capacities of vertices grouped them.

$$\text{Dist}(A, B) - (\text{Dist}(A, x) + \text{Dist}(B, x)) = 0 \tag{9}$$

The next step is to partition the main subtree forming clusters. For this purpose, from one of the selected vertices Then, from one of the vertices selected for the formation of subtree main (A and B vertices), this procedure adds the groups of vertices of subtree main adjacent to this vertices. In addition, it's necessary to verify if the sum of its capacity exceeds the lower limit predetermined. So, when this minimal restriction capacity must be satisfied, the cluster is formed and the new cluster is initiated from the remaining vertices in the subtree main.

And like the construction procedure 2, the third version of the constructor procedure works specifically in an attempt to form valid solutions in which, Besides of the connectedness restriction, minimal restriction capacity must be satisfied too. While this version possess another objective in relation to the construction procedure 1, its algorithms are similar.

In this version, the RCL is formed by α edges that, if removed, it's possible to generate solutions of α solutions that meet the minimal restriction capacity and, at the same time, maximizing the capacity of the clusters.

3.4.3 Local Search

Six versions for construction of the local search procedures had been used: local search procedure 1, local search procedure 2, local search procedure 3, local search procedure 4, local search procedure 5 and local search procedure 6.

The local search procedure 1 has as main focus the elimination of penalties. To that end, it uses a list of edges removed of the AGM and a list of clusters penalized. As from the elimination of the use of the list, the procedure verifies the possibility of to execute migrations of vertices between clusters. In cases where the migration possibility exists, the vertice of the edge removed which belongs cluster isn't penalized will migrate to the cluster penalized. This operation doesn't guarantee that the cluster penalized that it becomes a viable cluster, that the migrations of vertices makes this the cluster of origin is penalized and not even improving the quality of the solution. However, its systematic application, this may assist in the production of valid solutions. Like local search procedure 1, the local procedure 2 has as main focus the elimination of penalties too, independent of improving the solution. However, in this

version, it's executes migrations of vertices between clusters considering the original graph and not only the built AGM.

The local search procedure 3 it has the sole purpose of at minimizing ability of the solution. For this purpose, like local search procedure 2, it's executes migrations of vertices between clusters considering the original graph and the built AGM. Therefore, it's again becomes necessary to execute the procedure verifies if of the connectedness restriction is satisfied when there is migration of vertices.

The local search procedure 4 works with the edges removed in the formation of clusters. This version aims the search for better solutions through of the union and repartitioning clusters initially constructed. The clusters which it's edges removed of vertices belong, it's randomly selects, it can be joined in cluster to, after that, this cluster can be partitioned again by applying to construction procedure 1. The local search procedure 5 is similar to local search procedure 4, but it is applied with the objective to obtain new solutions are not penalised, i.e., that meet the minimal restriction capacity. And finally, the local search 6 objectives at minimizing ability of the solution and, for this purpose, it carries through migrations of vertices between clusters, on the basis of the original graph.

4 Computational Results

Section presents the computational results from the implementation of the algorithms proposed here, in C language. Using the NetBeans 7.0.1 Development Environment, all algorithms were executed on a computer with 2GB *RAM* memory, Intel Core *2* Duo of 64 bits with 1,67Hz processor, Windows 7 Ultimate Operating Disk.

The instances used to evaluate the proposed algorithms in this work refer to data obtained from 2000 Population Census sample data [Censo Demográfico, 2000],[Silva *et. al.*, 2004]. For the instances describes in Table 1 were used two text files: the first containing information of neighborhood between the APONDs objects associated to the Federation Unit (Espírito Santo (ES), Rondônia (RO), Amazonas (AM) and Bahia (BA)) and the second file containing the identification of the weighting area and the variables used to calculate distances: total number of houses, homes, people, the sum of income, the total households in APOND, years of study of the head of household, income per capita of households and average number of years of schooling of the head of household. Since the attribute related to the capacity used in this study corresponds to the total number of individuals. The following Table 1:

Table 1. Instances used in experiment

UF	Nº of Vertices (APONDs)	Nº of Edges
ES	73	350
RO	14	46
AM	61	286
BA	409	2020

The algorithms were implemented in total of 10 (ten) iteration each one and the GRASP alpha parameter was 10 (ten) . In addition to these settings, it's necessary to inform algorithms which the amount of clusters *k*.

GRASP using *AGM*, using Construction Algorithm 1, 2 and 3 respectively, and using the Local Search Algorithm (1, 4 and 5) generated the following versions: G1AGM1, G2AGM2 and G3AGM3. GRASP using the Original Graph, using the Construction Algorithm 1, 2 and 3 respectively, and using o the Local Search Algorithm (2, 3 and 6) generated the following versions: G1, G2 and G3. GRASP with Path-Relinking using the *AGM*, using Construction Algorithm 1, 2 and 3 respectively, and using the Local Search Algorithm (1, 4 e 5) generated the following versions: GPR1AGM1, GPR2AGM2 and GPR3AGM3. GRASP with Path-Relinkg using the Original Graph, using the Construction Algorithm 1, 2 e 3 respectively, and using the Local Search Algorithm (2, 3 and 6) generated the following versions: GPR1, GPR2e GPR3.

The Path-Relinking module used in this study was the backward reconnection from the best solution applied (Initial Solution) to the worst solution (Guide Solution). The optimization strategy used was intensification, applied input solution generated local search phase of GRASP and a selected solution (Solution Elite = distinct set of best solutions generated by GRASP, where *h* is an input data).

The experiment considered the application of algorithms version of GRASP and GRASP with Path-Relinking using *AGM* and Original Graph from instance presented in Table 1 for obtained of 3 clusters. From the results obtained are presented analysis evaluating algorithms in relation the quality of the solutions. It was possible to classify the solutions of good quality according to the following categories: Excellent equal to Gap using Equation 10 to 0%, Reasonable equal to Gap greater than 0% and less than or equal to 10% and Rubbish equal to Gap above 10%.

$$Gap = 100 * \frac{solution - solution_{best}}{solution_{best}} \tag{10}$$

Only valid solutions are considered, ie, those in which the restriction of beyond the minimum capacity and connectedness for cluster were made

We tested versions of GRASP and GRASP with Path-Relinking for all instances. It was observed that most versions have Gaps above 10%, i.e., not good results. The Tables 2 presents the best Gap obtained by GRASP and GRASP with Path-Relinking. GRASP with Path-Relinking, version GPR3, stands in relation to GRASP for the quality of the solutions, but consumed processing time more in all instances.

Table 2. Best Gap (0%) and Time (seconds) obtained with GRASP and GRASP with Path-Relinking for 3 *clusters*

	BEST GAP ($Gap = 0\%$) and Time (s)			
Instance	**GRASP Version G3**		**GRASP with PATH-RELINKING - Version GPR3**	
	Gap	**T(s)**	**Gap**	**T(s)**
ES	0%	10	0%	14
RO	0%	1	0%	8
AM	0%	7	0%	16
BA	0%	395	0%	424

The Table 3 presents, versions of GRASP and GRASP with Path-Relinking, the best solution ($solution_{best}$) obtained for the configuration 3 clusters.

Table 3. Best solution ($solution_{best}$) obtained with GRASP and GRASP with Path-Relinking for 3 *clusters*

	BEST SOLUTION ($solution_{best}$)	
Instance	**GRASP Version G3**	**GRASP with PATH-RELINKING - Version GPR3**
ES	119,91	98,25
RO	2,37	1,06
AM	56,99	42,93
BA	733,11	601,80

The Conclusion from this is that the module Path-Relinking helps considerably the performance of GRASP.

References

[Andrade, et. al, 2009] Andrade, A.V., Assis, L.P., Caires, L.F.V.: Aplicação do Método Path-Relinking na resolução do Problema Roteamento de Veículos com Coleta e Entrega Simultâneas. XLIISBPO (2009)

[Baum, 1896] Baum, L.O., Ochi, L.S., Macambira, E.M.: GRASP with Path Relinking for the SONET Ring Assignment Problem. In: Proc.of the 5th International Conferente on Hybrid Inteligente Systems (HIS 2005), pp. 239–244 (2005)

[Censo Demogr Censo Demográfico 2000. Primeiros Resultados da Amostra, Parte I, IBGE/CDDI (2001)

[Chiou and Lan, 2001] Chiou, Y.C., Lan, L.W.: Genetic Clustering Algortims. European Journal of Operational Research 135 (2001)

[Cole, 1998] Cole, R.M.: Clustering with Genetic Algorithms. Master's thesis, Departament of Computer Science, University of Western Australia (1998)

[Dias and Ochi, 2003] Dias, C.R., Ochi, L.S.: Efficient Evolutionary Algorithms for the Clustering Problems in Directed Graphs. In: Proc of the IEEE Congress on Evolutionary Computation (IEEE-CEC), Canberra, Austrália, pp. 983–988 (2003)

[Dias, 2004] Dias, C.R.: Algoritmos Evolutivos para o Problema de Clusterização de Grafos Orientados – Desenvolvimento e Analise Experiemental. Dissertação de Mestrado, Universidade Federal Fluminense (UFF), Niterói – RJ (2004)

[Fasulo, 1999] Fasulo, D.: An Analysis of Recent Work on Clustering Algorithms. Technical Report, Dept. of Computer Science and Engineering, Univ. of Washington (1999)

[Feo and Resende, 1995] Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133 (1995)

[Ferreira & Ochi 2012] Ferreira, C.S., Ochi, L.S., Parada, V., Uchoa, E.: A GRASP based approach to the generalized minimum spanning tree problem. Expert Systems with Applications 39(30), 3526–3536 (2012)

[Glover, F. and Laguna, 1997] Glover, F., Laguna, M.: Tabu Search. Kluwer, Boston (1997) ISBN 0-7923-9965-X

[Ho, S. and Gendreau,, 2006] Ho, S.C., Gendreau, M.: Path relinking for the vehicle routing problem. Journal of Heuristics 12(1), 55–72 (2006)

[Ibaraki et.al, 2005] Ibaraki, T., Nonob, E. K., Yagiura, M.: Metaheristics: Progress as real problem solvers. Springer (2005)

[Openshaw, 1977] Openshaw, S.: A geographical solution to scale and aggregation problems in regionbuilding, partitioning and spatial modelling. Transactions of the Institute of British Geographers (New Series) 2, 459–472 (1977)

[Semaan et al., 2008] Semaan, G.S., Ochi, L.S., Brito, J.A.M., Montenegro, F.: An Efficient Evolutionary Algorithm for the Aggregated Weighting Areas Problem. In: Proc. of the International Conference on Engineering Optimization (2008)

[Semaan et al., 2009] Semaan, G.S., Brito, J.A.M., Ochi, L.S.: Um algoritmo evolutivo hibrido aplicado ao problema de clusterização em grafos com restrições de capacidade e contiguidade. Anais do IX Congresso Brasileiro de Redes Neurais e Inteligência Computacional (IX CBRN), Ouro Preto/MG (2009)

[Semann, 2010] Semaan, G.S.: Algoritmos Heurísticos para o Problema de Particionamento de Grafos com Restrições de Capacidade e Conexidade. Dissertação de Mestrado, Universidade Federal Fluminense (UFF), Niterói – RJ (2010)

[Silva et al., 2004] Silva, A.N., Cortez, B.F., Matzenbacher, L.A.: Processamento das Áreas de Expansão e Disseminação da Amostra no Censo Demográfico 2000, Textos para Discussão, número 17, IBGE/DPE/COMEQ (2004)

[Silva & Ochi, 2010] da Silva, A.R.V., Ochi, L.S.: Hybrid Heuristics for Dynamic Resource-Constrained Project Scheduling Problem. In: Blesa, M.J., Blum, C., Raidl, G., Roli, A., Sampels, M. (eds.) HM 2010. LNCS, vol. 6373, pp. 73–87. Springer, Heidelberg (2010)

[Silva et al., 2012] Silva, M.M., Subramanina, A., Ochi, L.S., Vidal, T.: A simple and effective heuristic for the Minimum Latency Problem.To appear in European Journal of Operational Research, EJOR (2012)

[Trindade and Ochi, 2006] Trindade, A.R., Ochi, L.S.: Um algoritmo evolutivo híbrido para a formação de células de manufatura em sistemas de produção. Pesquisa Operacional 26(2) (2006)

A Self-organizing Genetic Algorithm for UWB Microstrip Antenna Optimization Using a Machine Learning Technique

Sinara R. Martins, Hertz W.C. Lins, and Cláudio R.M. Silva

Communication Engineering Department,
Federal University of Rio Grande do Norte – UFRN, Natal, Brazil
{sinaramartins,hertzw,claudio.rmsilva}@gmail.com

Abstract. This paper presents an application of a machine learning technique to enhance a multi-objective genetic algorithm to estimate fitness function behaviors from a set of experiments made in laboratory to analyze a microstrip antenna used in ultra-wideband (UWB) wireless devices. These function behaviors are related to three objectives: bandwidth, return loss and central frequency deviation. Each objective (modeled as dependent of an antenna slit dimensions Ls and Ws) is used inside an aggregate adaptive weighted fitness function that estimates the multi-objective behavior in the algorithm. The final results were compared with the ones obtained with a similar antenna modeled in a simulator program and with the ones of a real prototype antenna built from the optimal values obtained after the optimization. The final comparison has shown a promising gain for the designed antenna in the analyzed frequencies.

Keywords: Machine Learning, UWB Antenna Design, Evolutionary Algorithm, Multi-objective Optimization, Genetic Algorithm.

1 Introduction

The world has become easier with the advent of wireless communications. In many situations, it is not necessary anymore to worry about connectors and short or disrupted wires to access a computer network using a notebook, a tablet or even a smartphone. This has been also a reality inside our homes, since increasingly home devices are shipped with computer network access functionalities to wireless local area networks (WLANs) and wireless personal area networks (WPANs). Unfortunately, WLAN and WPAN characteristics are not enough to meet the needs of the emerging consumer electronic devices [1]. The next generation of WPANs, for example, aims to boost performance with higher data rates, better throughput, lower power consumption, and lower interference. These characteristics make UWB technology one of the main candidates to fit for next-generation WPANs. Some UWB chipset manufacturers estimate that its products will have a range between 9 and 10 meters with data rates around 450 Mbps. Besides, UWB offers some other benefits worth

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 642–649, 2012.

considering, since the UWB transceivers are extremely easy and cheap to build compared to typical spread spectrum transceivers. UWB systems also consume less power, and this makes them feasible for smaller devices. Moreover, the relatively wide spectrum that UWB utilizes significantly minimizes the interference from other systems.

The consequent widespread of ultra-wideband systems has aroused interest in the subject of ultra-wideband antenna design. In the case of UWB antenna, the design task may has several, possibly conflicting, objectives. Typically, a balance has to be found between the various competing objectives. In terms of optimization, multiple objectives usually involve multiple fitness functions that have to be considered simultaneously to solve a specific problem. These diverse considerations has made UWB antenna design an interesting problem to computational intelligence methods. One of these methods is the evolutionary algorithms (EA). The main advantage of EA, when applied to solve multi-objective optimization problems is the fact that they typically optimize sets of solutions, allowing computation of an approximation of the entire Pareto front in a single algorithm run. These evolutionary algorithms may be self-organizing or not [2]. In general, when they are self-organizing methods, the composition of the several objectives involved is adjusted by the algorithm without the need of expertise knowledge. The EA is also known as Evolutionary Computation (EC) and its developments and applications have been one of the fastest growing fields in computing science. Moreover, research into enhancing the EC algorithms with machine learning (ML) techniques has played an important role in the literature. ML alone is already one of the most promising and salient research areas in artificial intelligence, since it has become a powerful tool in a wide range of applications [3].

This work explores one of these applications of ML. It presents a self-organizing multi-objective genetic algorithm based on a technique that adopts a machine learning process to estimate fitness function behaviors from a set of experiments made in laboratory to analyze and optimize a microstrip antenna used in ultra-wideband wireless devices. These function behaviors are related to three objectives: bandwidth, return loss and central frequency deviation. Each objective (modeled as dependent of an antenna slit dimensions Ls and Ws) is modeled inside an aggregate adaptive weighted fitness function that estimates the multi-objective behavior in the algorithm.

The organization of the paper is as follow: section 2 provides the terminology used in a UWB microstrip antennas design process; section 3 introduces some background about the multi-objective genetic algorithms behavior and machine learning aspects; sections 4 describes the design and implementation aspects related with the prototype algorithm developed; section 5 discuss the results obtained with this prototype; and section 6 presents the main conclusions of the authors about the work.

2 Antenna Design

Several factors must be taken into account in order to choose an optimum antenna topology for ultra wideband (UWB) design. It is important to consider the physical profile and geometry to be used, compatibility, impedance bandwidth, radiation

efficiency, return loss, and radiation pattern. The main challenge in UWB antenna design is achieving the very broad bandwidth with high radiation efficiency and small in size. However, construction of microstrip antennas with these characteristics without loss of performance is still a challenge to current research. Several designs approaches have been proposed [4] [6]. But most of these antennas involve complex calculation and sophisticated fabrication process. A typical UWB antenna consists of a thin metal layer of some geometry that acts as a radiator element, separated from its ground plane by a dielectric substrate layer. See Figure 1.

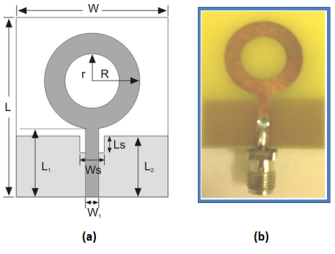


Fig. 1. Geometry of the microstrip antenna (a) physical antenna (b)

This work focuses on UWB planar printed circuit board (PCB) antenna design and analysis. It is based on one of our previous work [1], but it presents now a different approach, where the algorithm is applied to optimize not only one single dimension of the antenna, but two of them (slit dimensions of the ground plane): Ws and Ls. The antenna geometry implemented is known as Ring Monopole and it was designed from the following specifications and dimensions: relative permittivity Substrate: 4.4, Substrate Thickness: 1.57 mm, fed line: 50 Ω, L: 33 mm, L1: 12.5 mm, L2: 12 mm, W: 28 mm, W4: 2.7 mm, Outer radius (R): 8.8 mm, Inner radius (r): 5.0 mm (constant dimensions). The optimization process search for the ideal dimensions of length Ls and width Ws (variable dimensions) of the slit used in the truncated ground plane in order to find optimum values for bandwidth, return loss and fitness center frequency.

3 ML Techniques Enhancing Multi-objective EC Algorithms

Multi-objective optimization is the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints. These objectives may also or not be dependents among themselves [5]. Several real world problems of engineering involve simultaneous optimization of more than one conflicting objective. In these problems, there are a set of optimal solutions where each solution satisfies the objectives at an acceptable level. Their search space is normally difficult to search, since it often has multimodal behavior with discontinuity and some noises in space.

In these environments, often the choice of parameters of an algorithm can have a significant impact on the effectiveness of optimization [8]. Evolutionary Computing (EC) algorithms are popular approaches for solving this kind of optimization, especially those ones using genetic algorithms (GA). GA is one type of EC algorithm.

On the other side, ML refers to the use of formal structures (machines) to make inference (learning). This includes the construction of models proposing mathematical expressions that encapsulate the mechanism by which a physical process gives rise to observations [3]. ML can be used in various contexts and applications, and one of them is the EC. EC algorithms use computational models of natural processes of evolution and natural selection in order to solve real world problems. These algorithms depend on probabilistic factors, such as during the startup phase of the population and choosing parents. There are a number of computational models proposed, but all of them have in common the concept of simulating the evolution of species through genetic operators such as mutation, selection and reproduction.

Algorithms in evolutionary computation typically produce databases of sufficient size to obtain knowledge and improvement of the algorithm itself, allowing the use of ML techniques. In the literature, these ML techniques have been successfully used in the algorithms of EC, in order to improve the performance of the algorithm. ML techniques may include: statistical methods, interpolation and regression, clustering analysis, principle component analysis, orthogonal experimental design, opposition-based learning, artificial neural networks, support vector machines, case-based reasoning, reinforcement learning, competitive learning, Bayesian network and others [3].

ML techniques, such as interpolation and artificial neural network in a simple step can assist the initial solution of higher quality in the algorithms of CE [3]. Since such algorithms are more likely to start from points closer to the global optimum and can converge in least generations. In the next section show how the first technique could be used to enhance a GA's used for microstrip antenna design optimization.

4 Design and Implementation

The design of the genetic algorithm prototype with machine learning capabilities begins with four important questions. 1) Which parts of a classical genetic algorithm should benefit on the machine learning process? 2) Which ML technique should be used to the application domain under study? 3) How to adapt this technique to develop a variant one that is suitable to the genetic algorithm parts selected to change? And 4) How to put it all together in client-server model, since the whole design environment involves different professionals playing different roles in different places?

The UWB antenna design is typically a mix of mathematical modeling and empirical process. The standard approach is normally based in initial promising antenna geometry and its optimization for one or two of its dimensions using a numerical or an artificial intelligence method to estimate the behavior of the set of objective measure under analysis. Therefore, the answer to the first design question is to change the first two steps of a classical genetic algorithm to benefit of the learning process: The initial population generation step (since it depends of the optimizations dimensions chosen) and the fitness calculation step (since it depends of the objective measures) behavior to be modeled with the machine learning technique. See Figure 2.

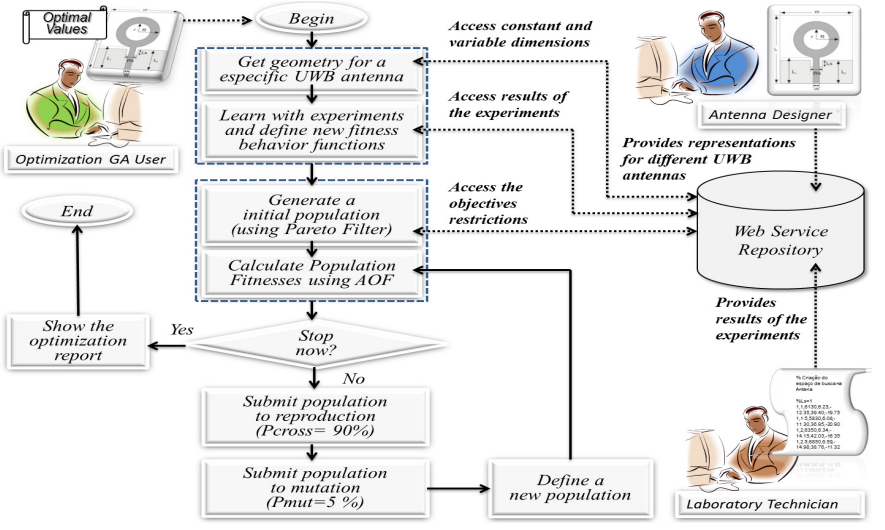


Fig. 2. The genetic algorithm and its learning aspects

In Figure 2, the initial population is generated from the behavior obtained from the experiments and its results for each objective (using a ML technique). After this step, a *Pareto filter* that considers restrictions for each objective (bandwidth>9.0GHz, Return Loss<-20dB and Frequency Deviation<0.37 Hz) is used to restrict the search space. The behaviors estimated by the ML technique (one for each objective) is also used to compose and aggregate objective fitness function (AOF) to evaluate each chromosome in the population. This AOF changes whenever a new result is added to the repository of experiments.

The choice of the appropriate ML technique is not easy task, since there are a great number of them in the literature [3]. As discussed in previous section, interpolation is one of the known possibilities, since it is a fast technique, relatively easy to implement and flexible enough to enable different approaches. Therefore, the second question answer is to consider a variant approach of the Interpolation technique. The choice of the interpolation variant should consider the context of a typical multi-objective genetic objective based in the analysis of a compound and weighted aggregate objective function (considering that all objectives depend of two variables). Therefore, the third design decision should be a variant approach based in an interpolation process not for a line but for a surface associated with the different objectives. But how would it be? In a formal way, interpolation is known as a method of constructing new data points within the range of a discrete set of known data points. This is interesting because in typical antenna design, one often has a number of data points (variable antenna parameters), obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable (fitness function to a specific objective). So, it is often required to interpolate the value of that function for an intermediate value of the independent variable. This may be achieved by curve fitting or regression analysis. The method

used here is based in a special case of curve fitting with a variant of *Spline interpolation*. Spline interpolation is a form of interpolation where the interpolant is a type of piecewise polynomial named *spline*. While polynomial interpolation fits a curve through all the data points at once, spline interpolation approximates a curve between each proximate pair of data points and adds all the curves together to create the final approximation. This is a good strategy if a function can only be approximated well with a polynomial of a very high order over the entire domain, but can be approximated well with a sequence of low-order polynomials for different parts of the domain. Which is the case of UWB antenna design. Spline interpolation techniques include linear, quadratic, and cubic interpolation. In this approach, the answer to the third design question is an interpolation process that extends cubic interpolation (*bicubic interpolation).* It is used for interpolating data points for the objectives used in the optimization on a two dimensional regular grid (formed by antenna variable parameters Ls and Ws). This type of interpolation computes a bicubic spline "s" which interpolates the (xi,yj,zij) points. That means that we have s(xi,yj)=zij for all i=1,..,nx and j=1,..,ny. The resulting spline s is defined by the triplet (x,y,C), where C is the vector of length 16(nx-1)(ny-1) with the coefficients of each of the (nx-1)(ny-1) patches: on [x(i) x(i+1)]x[y(j) y(j+1)]. The coefficients c_{ij} of the bi-cubic patches are defined inside a hyper matrix of ([4,4,nx-1,ny-1],C). So that the coefficient *(k,l)* of the patch *(i,j)* would be in stored in c(k,l,i,j). The final expression for S(x,y) is defined by:

$$S(x,y)= \sum_{k=1}^{4}\sum_{l=1}^{4} c_{ij}(k,l).(x-x_i)^{k-1}.(y-y_i)^{l-1} \quad (1)$$

In the optimization domain considered, each objective is associated with a different s(x,y), where x and y will be the variable antenna parameters Ls and Ws. So that, Sbw(Ls,Ws) represents the interpolation function for bandwidth, Srl(Ls,Ws) represents the interpolation function for return loss and Sfd(Ls,Ws) represents the interpolation function for central frequency deviation. The compound aggregate objective function is shown below with the empirical weights: wi for i=1,2 and 3.

$$AOF(Ls, Ws) = Max[w_1.Sbw(Ls,Ws) + w_2.Srl(Ls,Ws) + w_3.\ Sfd(Ls,Ws)] \quad (2)$$

The answer to the fourth design question involves a web prototype system that is accessed from different places by the professionals using web navigators. The web prototype system is also diagrammatically shown in Figure 2. It presents a web system designed to enable the interaction of three different professionals: the UWB antenna designer (a person that idealize the initial antenna characteristics such as geometry and others), the laboratory technician (a person that choose an specific antenna previously represented in the repository and associates experiments in which different values are assigned to the optimization dimensions to measure the consequent effect for each objective) and a optimization user (a person that desires to choose an specific antenna, previously represented in the repository using a XML syntax, and optimize it, if there exist experiments associated to this specific antenna that enable the optimization). Figure 2 also shows how the classical GA algorithm is modified to enable the inclusion of additional techniques to improve the algorithm

performance. The initial population procedure is now dependent of the XML representation of the antenna, since the codification of a chromosome is a binary representation for each variable dimension considered. The antenna used has two variables parameters: Ls and Ws (slit dimensions) and all the other parameters are constant. The fitness calculation step was modified to fit the spline interpolation process for each objective and its combination inside a weighted aggregate function. The other GA parameters consist of simultaneous binary genetic operations over a population consisting of individuals with two binary parts (one for each antenna variable dimension) with Mutation and Crossover probabilities defined as 5% and 90% respectively.

5 Results

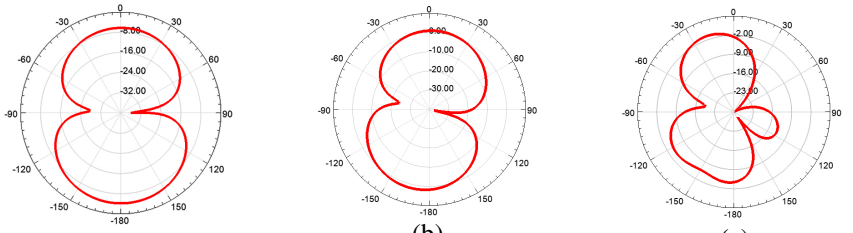
The prototype algorithm implemented was tested for different datasets of experiments considering the antenna shown in Figure 1b and a set of restrictions empirically defined. These datasets contain a different number of experiments in an increasing way to demonstrate that: as the number grows, the prototype algorithm increases its perception on the behavior of the objectives and on the joint behavior of the whole. Each optimization objective is associated with a restriction that must be observed in the optimization process. They are: a) Bandwidth values (BW) be bigger than 9 GHz; b) Return Loss values (RL) must be less than -20 dB; and c) Central Frequency Deviation (CFD) must be less than 0.37 Hz. The optimization results are shown in Table 1.

Table 1. Results of the optimization for w_1= *1.0*, w_2= *-1.0*, w_3= *-1.0*

Datasets	Ls (mm)	Ws (mm)	BW (GHz)	RL (dB)	CFD (Hz)
72	1.897959	4.020408	9.2791638	- 25.21127	0.3571445
100	2.387755	3.612244	9.3125183	- 20.31941	0.3698174
119	2.306122	3.693877	9.3206112	- 20.12885	0.3549482
162	1.000000	5.000000	9.3800000	-29.42000	0.0900000

Observing Table 1, one can notice that in all the situations, the restrictions have been met and that the best results were obtained with the last set of experiments. This datasets contains 162 experiments and presents the best results for all the objectives considered: bandwidth (9.38 GHz), return loss (-29.42 dB) and central frequency deviation (0.09 Hz). These results for Ls (1.000000) and Ws (5.000000) were then compared with the results simulated by *Ansoft HFSS* software. The HFSS obtained the following results for Ls=1.000000/Ws=5.00000: bandwidth (9.16 GHz), return loss (-48.97 dB) and central frequency deviation (0.65 Hz). This confirms the optimization process. The next step was the construction of the physical antenna prototype, it has been analyzed by measurements of the signal propagated in free space and implemented using the vector network analyzer from Rohde & Schwarz brand, model ZVB-14. The simulations were performed by changing the dimensions of the slot ground plane for each objective provided the AG values of the dimensions of the slit respectively combined, bandwidth, return loss and fitness center frequency. Figure 3 shows the behavior of radiation generated of the antenna at frequencies of 3.5 GHz, 6

GHz and 9.5 GHz and their respective gains, varying the polar coordinates θ = 0 ° to 360 ° and θ = φ = 0 °. We can observe an increasing amount of the gain as the frequency rises, especially the frequency of 6.0 GHz and 9.5 GHz.



(a) (b) (c)

Fig. 3. Behavior of the antenna radiation at frequencies 3.5 (a), 6.0 (b) and 9.5 (c) GHz

6 Conclusion

An antenna for UWB operation was constructed and optimized by a self organizing multi-objective genetic algorithm prototype improved with an interpolation ML technique. This ML technique enabled a dynamic estimation of the fitness function used in the algorithm that made it possible to learn with a set of experiments stored in a web system repository. The final optimal values found allowed the construction of a prototype UWB antenna that presented a reasonable gain in the analyzed frequencies of 3.5, 6.0 and 9.5 GHz. Although, the current work have considered only two optimization dimensions, the general approach for "n" dimensions is also feasible, since the spline interpolation described could be perfectly adapted to the general case.

References

1. Prasad, R.: OFDM for Wireless Communication Systems. Artech House universal personal communications series (2004) ISBN 1-58053-796-0
2. Silva, C.R.M., Lins, H.W.C., Martins, S.R., Barreto, E.L.F., D'Assunção, A.G.: A Multiobjective Optimization of a UWB Antenna Using a Self Organizing Genetic Algorithm. In Process of Publication by Microwave and Optical Technology Letters (2012)
3. Zhang, J., et al.: Evolutionary Computation Meets Machine Learning: A Survey. IEEE Computational Intelligence Magazine 6(4), 68–75 (2011)
4. Huang, C.Y., Hsia, W.C.: Planar Elliptical Antenna for Ultra wideband Communications. Electronics Letters 41(6) (2005)
5. Konak, A., Coit, D.W., Smith, A.E.: Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering & System Safety 91(9) (2006)
6. Liang, J., Chiau, C.C., Chen, X., Parini, C.G.: Printed circular disc monopole antenna for ultra-wideband applications. Electronics Letters 40(20) (2004)
7. Oliveira, E., D'Assunção, A.G., Silva, C.R.M.: Optimization of the input impedance of Koch triangular quasi-fractal antennas using Genetic Algorithms. In: 14th Biennial IEEE Conference on Electromagnetic Field Computation (2010)
8. Grefenstette, J.J.: Optimization of control parameters for genetic algorithms. IEEE SMC-16(1), 122–128 (1986)

Error Concealment by Means of Motion Refinement and Regularized Bregman Divergence

Alessandra Martins Coelho[1], Vania V. Estrela[2], Felipe P. do Carmo[3], and Sandro R. Fernandes[4]

[1] Instituto Federal de Educacao, Ciencia e Tecnologia do Sudeste de Minas Gerais (IF SEMG), Rio Pomba, MG, Brazil
alessandra.coelho@ifsudestemg.edu.br
[2] Telecommunications Department, Universidade Federal Fluminense (UFF), CEP 24210-240, Niterói, RJ, Brazil
vestrela@id.uff.br
[3] Colégio Pedro II, Rua Bernardo de Vasconcelos, 941, Realengo, CEP 21710-261, Rio de Janeiro, RJ, Brazil
felcarmo@yahoo.com.br
[4] Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais, Rua Bernardo Mascarenhas, 1283 Fábrica CEP 36080-001, Juiz de Fora, MG, Brazil
sandro.fernandes@gmail.com

Abstract. This work addresses the problem of error concealment in video transmission systems over noisy channels employing Bregman divergences along with regularization. Error concealment intends to improve the effects of disturbances at the reception due to bit-errors or cell loss in packet networks. Bregman regularization gives accurate answers after just some iterations with fast convergence, better accuracy and stability. This technique has an adaptive nature: the regularization functional is updated according to Bregman functions that change from iteration to iteration according to the nature of the neighborhood under study at iteration *n*. Numerical experiments show that high-quality regularization parameter estimates can be obtained. The convergence is sped up while turning the regularization parameter estimation less empiric, and more automatic.

Keywords: Inverse problems, optical flow, Bregman divergences, exponential distributions, error concealment, computer vision, regularization.

1 Introduction

The extraction of motion information from a given video sequence is a major undertaking. In this *milieu*, optical flow (OF) (cf. [10]) is a recurrent concept with numerous applications, e.g., image compression/coding (cf. [11, 14]), automatic movie edition/digitalization (cf. [12]), reconstruction of 3D surfaces by means of depth from stereo (cf. [13]), object recognition and motion estimation (see, for example, [14-16]). OF discontinuities are of particular interest and they should be distinguishable from

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 650–657, 2012.

object borders i.e., large gradients of grayscale values within the projections of moving objects. Furthermore, due to imperfect communication channels, it is nearly impossible to attain reconstructed pictures with suitable visual quality. That is why data protection and error reduction methods such as decoder-based error concealment (EC) algorithms are required. They rely on two types of redundancies: spatial (SEC) and temporal (TEC), requiring no alterations on the bit-stream syntax and transport technology. SEC algorithms: (a) interpolate the lost area using spatially neighboring image data; (b) presuppose statistical correlation between adjacent image blocks; and (c) provide a good approximation for the lost macroblocks (MBs). TEC schemes utilize previously decoded image data to estimate motion vectors (MVs) of the lost MBs to compensate for errors. EC is largely dependent upon the ability of the system to detect errors, since EC operations are applied to corrupted MBs. Because the damaged packetized bit-stream is thrown out and considered missing; we can obtain the MB position where an error occurs by checking the MB address (MBA), which defines the absolute position of the MB. The Bregmanized regularization relies on Bregman divergences constructed with the q-discrepancy functional [3, 4], so that the regularization function does not have to be fixed at each interaction [2, 5]. Information on the discrepancy principle can be used to improve the stopping criterion as well [3, 4]. The connection with exponential distributions allows for entropy-based estimation methods [8, 9] because numerous well-known divergences, such as relative entropy, can be expressed as Bregman divergences on the distribution parameters.

In this paper, the problem of error concealment in transmission of video over noisy channels is addressed. Section 2 states the motion estimation problem used in this text. Section 3 casts the problem in terms of the minimization of a regularization functional term depending on the Bregman divergence. The error concealment algorithm is introduced in Section 4. Experimental results are shown in Section 5. Finally, conclusions are drawn in Section 6.

2 The Motion Estimation Problem

OF is the distribution of apparent velocities of movement for intensities of pixels $I_k(\boldsymbol{r})$ of the k-th frame at location $\boldsymbol{r} = [h, v]^T$ in an image and it requires at least two consecutive frames. The displacement of every pixel in a frame forms the displacement vector field (DVF). We seek the corresponding displacement vector (DV) $\boldsymbol{d}(\boldsymbol{r}) = [d_h, d_v]^T$ at the working point $\boldsymbol{r}$, in the current frame k, in order to minimize the displaced frame difference (DFD) in an area containing the working point and assuming constant image intensity along the motion trajectory. The perfect registration of frames results in $I_k(\boldsymbol{r})=I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r}))$. Then, the DFD can be written as $\Delta(\boldsymbol{r};\boldsymbol{d}(\boldsymbol{r}))=I_k(\boldsymbol{r})-I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r}))$. An estimate of $\boldsymbol{d}(\boldsymbol{r})$, is obtained via minimization of the gradient $\nabla I_{k-1}(\boldsymbol{r}-\boldsymbol{d}^i(\boldsymbol{r}))$ or by determining a linear relationship between these two variables through some model. This can be accomplished by using a Taylor series expansion of $I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r}))$ about the

location ($\boldsymbol{r}$-$\boldsymbol{d}^i(\boldsymbol{r})$), where $\boldsymbol{d}^i(\boldsymbol{r})$ represents a prediction of $\boldsymbol{d}(\boldsymbol{r})$ in the i-th step. This results in, where the displacement update vector is $\boldsymbol{x}=[x_h, x_v]^T = \boldsymbol{d}(\boldsymbol{r}) - \boldsymbol{d}^i(\boldsymbol{r})$, and $e(\boldsymbol{r}, \boldsymbol{d}(\boldsymbol{r}))$ stands for the truncation error resulting from higher order terms and $\nabla=[\delta/\delta_h, \delta/\delta_v]^T$ represents the spatial gradient operator [17]. Considering all points in a neighborhood of pixels around $\boldsymbol{r}$ leads to

$$\boldsymbol{y}=\boldsymbol{H}\boldsymbol{x}+\boldsymbol{\eta}, \tag{1}$$

where the temporal gradients $\nabla I_{k-1}(\boldsymbol{r}-\boldsymbol{d}^i(\boldsymbol{r}))$ have been stacked to form $\boldsymbol{y}$ containing DFD information on the pixels in a neighborhood, $\boldsymbol{H}$ contains the spatial gradient operators at each observation, and the error terms have formed the additive white noise vector $\boldsymbol{\eta}$. the corrected DV is given by

$$d^{i+1}(\boldsymbol{r}) = d^i(\boldsymbol{r}) + \boldsymbol{x}^i(\boldsymbol{r}), \tag{2}$$

at iteration i. The ordinary least squares (LS or OLS) estimate of the update vector is

$$\hat{\boldsymbol{x}}_{LS} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{y}. \tag{3}$$

3 The Motion Recovery Algorithm

Segmenting OF via EM algorithm for mixtures of DVs can be done successfully [15] because it is presumed that there is little or no interference amid individual sample constituents or that all the constituents in the samples are known ahead of time. The Tikhonov regularization functional (TRF) (cf. [1, 2]) associated to Eq. (1) is

$$\begin{aligned} Q(\hat{\boldsymbol{x}}) = & \sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\left[y(i,j) - \sum_{k=-N}^{N}\sum_{l=-N}^{N} b(k,l)\hat{x}(i+k, j+l) \right]^2 \\ & + \alpha \frac{1}{1+q} \sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\left\{ \hat{x}_{i,j}\left[\frac{(\hat{x}_{i,j})^q - (\overline{x}_{i,j})^q}{q} \right] - (\overline{x}_{i,j})^q(\hat{x}_{i,j} - \overline{x}_{i,j}) \right\}, \end{aligned} \tag{4}$$

where α is the regularization parameter, and is the estimate of $\boldsymbol{x}$ obtained with Eq. (4). S corresponds to a family of functions (Bregman divergences) given by

$$S = D_q(\hat{\boldsymbol{x}}, \overline{\boldsymbol{x}}) = \frac{1}{1+q} \sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\left\{ \hat{x}_p\left[\frac{(\hat{x}_{i,j})^q - (\overline{x}_{i,j})^q}{q} \right] - (\overline{x}_{i,j})^q(\hat{x}_{i,j} - \overline{x}_{i,j}) \right\}, \tag{5}$$

$\overline{\boldsymbol{x}}$ stands for a reference value and q is an adjustable parameter (q-discrepancy). The nonlinear system becomes

$$F_{rs}(\hat{\mathbf{x}}) = \frac{\partial Q(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}_{rs}} = 0, \tag{6}$$

for $r,s=0,1,2,...,M-1$. We seek the q and $\hat{\boldsymbol{x}}^t$ that minimize Eq. (6), where t is an iteration counter and $\hat{\boldsymbol{x}}^0$ is the initial estimate of $\boldsymbol{x}$. The Newton-Raphson method yields

$$\hat{\boldsymbol{x}}^{t+1} = \hat{\boldsymbol{x}}^t + \Delta\hat{\boldsymbol{x}}^t, \quad t = 0,1,2,..., \tag{7}$$

A first-order Taylor expansion of Eq. (6) results in

$$F_{rs}(\hat{\boldsymbol{x}}^{t+1}) = F_{rs}(\hat{\boldsymbol{x}}^t + \Delta\hat{\boldsymbol{x}}^t) = F_{rs}(\hat{\boldsymbol{x}}^t) + \sum_{m=1}^{M-1}\sum_{n=1}^{M-1} \frac{\partial F_{rs}}{\partial \hat{\boldsymbol{x}}_{mn}}\bigg|_{\hat{x}^t} \Delta\hat{\boldsymbol{x}}_{mn}^t = \boldsymbol{0}. \tag{8}$$

With the help of the Gauss-Seidel method, we find

$$\Delta\hat{x}_{rs}\Big|^{t,c+1} = -\frac{1}{\left(\partial F_{rs} \big/ \partial \hat{x}_{mn}\right)\Big|_{\substack{t,c \\ m=r \\ n=s}}} \left\{ F_{rs}\Big|_{\hat{x}^{t,c}} + \sum_{\substack{m=0 \\ m\neq r}}^{M-1}\sum_{\substack{n=0 \\ n\neq s}}^{M-1} \frac{\partial F_{rs}}{\partial \hat{x}_{mn}}\bigg|_{\hat{x}^{t,\tilde{c}}} . \Delta\hat{x}_{mn}^{t,\tilde{c}} \right\}, \tag{9}$$

with $\Delta\hat{\boldsymbol{x}}^{t,0} = \boldsymbol{0}$, c is the iteration counter and

$$\tilde{c} = \begin{cases} c+1, & \text{if } (m < r) \text{ or } (m = r \text{ and } n < s) \\ c, & otherwise \end{cases}. \tag{10}$$

Now, the update term becomes

$$\Delta\hat{x}_{rs}\Big|^{t,c+1} = \Delta\hat{x}_{rs}\Big|^{t,c} - \frac{1}{\left(\partial F_{rs} \big/ \partial \hat{x}_{mn}\right)\Big|_{\substack{t,c \\ m=r \\ n=s}}} \left\{ F_{rs}\Big|_{\hat{x}^{t,c}} + \sum_{\substack{m=0 \\ m\neq r}}^{M-1}\sum_{\substack{n=0 \\ n\neq s}}^{M-1} \frac{\partial F_{rs}}{\partial \hat{x}_{mn}}\bigg|_{\hat{x}^{t,\tilde{c}}} . \Delta\hat{x}_{mn}^{t,\tilde{c}} \right\}, \tag{11}$$

Once the corrections $\Delta\hat{\boldsymbol{x}}_{RS}$ are calculated, new estimates $\Delta\hat{\boldsymbol{x}}^{t+1}$ can be obtained from Eq. (11) and with a convergence factor γ, where $0 < \gamma < 1$, Eq. (7) becomes

$$\hat{x}^{t+1} = \hat{x}^t + \gamma\Delta\hat{x}^t, \quad t = 0,1,2,.... \tag{12}$$

The previous expression converges faster than Eq. (7).

Fig. 1. (a) Decoded 14-th frame; and (b) Missing macro block location for the Foreman sequence (inter frame)

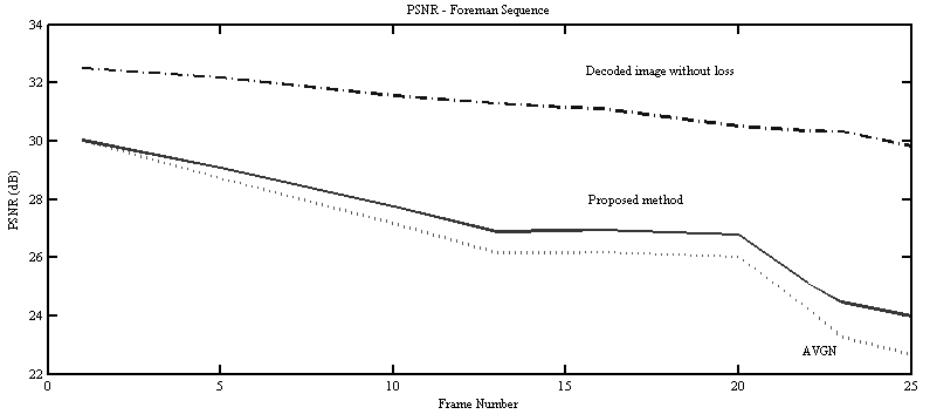


Fig. 2. PSNR plots for the Foreman sequence

Fig. 3. (a) Decoded 8-th frame; and (b) Missing macro block location for the Mother and Daughter sequence (inter frame)

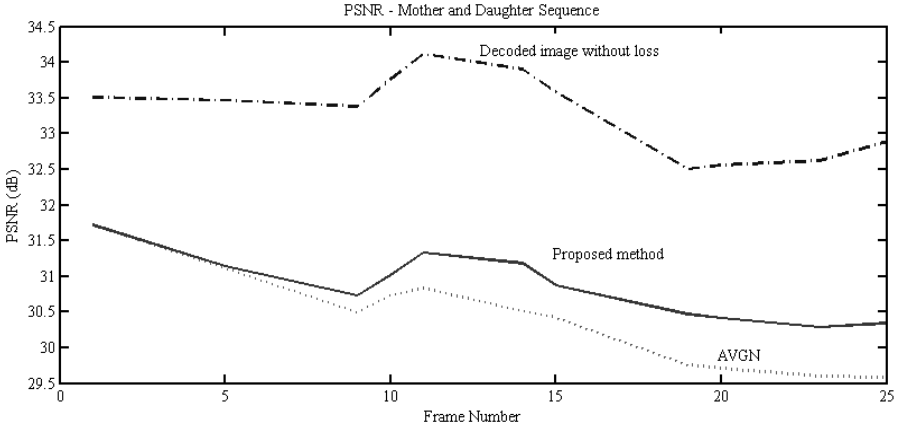


Fig. 4. PSNR plots for the Mother and Daughter sequence

4 Error Concealment

A priori information on the images as well as redundancies in both space (*h-v* directions) and time help to detect and correct errors. Intra frames of compressed video are basic frames that help generating inter frames: if intra frames contain lost data, then this will affect inter frames. This work assumes that missing MVs are correlated to the MVs of their neighbors. The proposed method assumes that a finite number of MV clusters exists. Each cluster corresponds to the displacement of a given region inside a frame (cf. [9, 13]). Once the received information is depacketized, a frame will have its pixels labeled as containing legitimate and erroneous regions. The legitimate ones can be used to calculate MVs with a simple procedure such as the one from Eq. (3). The motion recuperation algorithm from the previous section will be applied to cluster and correct the whole OF. The regularization operator

$$G(\hat{\boldsymbol{x}}) = \frac{1}{1+q}\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\left\{ \hat{x}_{i,j}\left[\frac{(\hat{x}_{i,j})^q - (\overline{x}_{i,j})^q}{q}\right] - (\overline{x}_{i,j})^q(\hat{x}_{i,j} - \overline{x}_{i,j})\right\}$$

from Eq. (4) is critical since it incorporates prior knowledge about the original uncorrupted frame into the recovery problem. The choice of the regularization parameter α also affects the final result.

5 Experimental Results

Two *176×144* QCIF sequences were used: the "Foreman" and "The Mother and Daughter". The peak signal to noise ratio (*PSNR*) was chosen as a measure of performance. For an *8* bit *M×N* image, the *PSNR* in dB is given by

$$PSNR = 10\log\left\{\frac{255^2 MN}{\left\|\boldsymbol{w} - \hat{\boldsymbol{w}}\right\|^2}\right\},$$

where $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$ are, correspondingly, the original and restored images. The experiments considered a 5% cell loss. Fig. 1 and Fig. 3 show examples with a complete and a corrupted frame (i.e., missing macroblocks) for each sequence used. For all experiments, it was assumed that *q=1* and γ=0.8. The algorithm was applied to compute $Q(\boldsymbol{x})$, by means of a constant regularization parameter α and with an optimal α. However, a constant α may yield bad convergence as the maximum number of iterations grows because the curve for $Q(\boldsymbol{x})\times\alpha$ resumes increasing after a few iterations. To some extent, increasing the value of α_0 remediates this behavior. This becomes necessary due to the need to obey other bounds on errors and trends used to assess and to make a decision upon the optimum estimate selection. Multicriteria and an adaptive Bregaminized regularization algorithm give better results, although increasing the computational load. Experiments were also performed with a simple EC algorithm relying on the average of the MVs from neighbors (AVGN) and without loss of cells. Fig. 2 and Fig. 4 show *PSNR* plots when the three algorithms are applied to the "Foreman" and the "Mother and Daughter" sequences. It is important to mention that the use of Bregman Divergence s improved the displacement vectors around borders.

6 Conclusions

This work solves the error concealment problem by means of Bregman divergence s and regularization [5,8,9,10], where the regularization function does not have to be fixed at each interaction any longer. It is important to devise lower and upper values for α^{ρ}, so that convergence and optimization are more efficient. Certainly, the convergence, stability and merit of the estimates are improved when judged against to iterations relying on invariable values of α. The fact that more local information on the image neighborhood is added to the regularization procedure is the main motive for these improvements. Occasionally, Bayesian estimation introduces estimation bias by prior information that may be needless. A possibility is to model estimation as a minimization of an expected Bregman divergence between the unknown and the projected distributions. It has been proven that Bregman iteration also approaches a solution with much less computational complexity than conventional regularization. Furthermore, when stopped according to the discrepancy principle, Bregman iteration is also a choice method for solving compressed sensing problems and they are very suitable for parallel implementations due to its characteristics [14-16]. Bregman divergences [13, 14] can be employed as measures of nearness—these divergences are natural for learning low-rank kernels since they maintain rank as well as positive semi-definiteness. Special cases of the proposed framework yield faster algorithms for several learning problems, and experimental results show that this algorithm can effectively learn both low-rank and full-rank kernel matrices.

References

1. Bregman, M.: The Relaxation Method of Finding the Common Point of Convex Sets and its Application to the Solution of Problems in Convex Programming. USSR Comt. Math and Math. Ph. 7(3), 200–217 (1967)
2. Tikhonov, A.N., Arsenin, V.Y.: Solutions of Ill-Posed Problems. J. Willey & Sons (1977)
3. Stutz, D.: Restauração de Imagens em Escala Nanométrica com Funcional de Regularização de Tikhonov e Computação Paralela. M.Sc. thesis, IPRJ/UERJ, N. Friburgo, RJ, Brazil (2004)
4. Stutz, D., Silva Neto, A.J., Farias, R.C.: Information Weighted Mean Square Error (IWMSE): Uma Medida de Comparação de Imagens Baseada na Percepção, X EMC, N. Friburgo, RJ, Brazil (2007)
5. Galatsanos, N.P., Katsaggelos, A.K.: Methods for Choosing the Regularization Parameter and Estimating the Noise Variance in Image Restoration and their Relation. IEEE Trans. on Im. Proc., 322–336 (1992)
6. Coelho, A.M., Estrela, V.V.: EM-Based Mixture Models Applied to Video Event Detection. In: Principal Component Analysis - Engineering Applications, pp. 102–124. Intech (2012) ISBN 9788563337214, http://www.intechopen.com/books/principal-component-analysis-engineering-applications/em-based-mixture-models-applied-to-video-event-detection
7. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An Iterative Regularization Method for Total Variation Based Image Restoration. Multiscale Modeling Sim. 4, 460–489 (2005)

8. Murata, N., Takenouchi, T., Kanamori, T., Eguchi, S.: Information Geometry of U-Boost and Bregman Divergence. Neural Comput. 16, 1437–1481 (2004)
9. Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S.: An Information Theoretic Analysis of Maximum Likelihood Mixture Estimation for Exponential Families. In: Proc. 21st ICML (2004)
10. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations, 2nd edn. Springer, New York (2006)
11. Hinterberger, W., Scherzer, O.: Models for Image Interpolation Based on the Optical Flow. Computing 66, 231–247 (2001)
12. Grossauer, H.: Inpainting of Movies Using Optical Flow, Math. Models for Registration and Applications to Med. Imaging, Math. Ind., vol. 10, pp. 151–162. Springer, Berlin (2006)
13. Slesareva, N., Bruhn, A., Weickert, J.: Optic Flow Goes Stereo: A Variational Method for Estimating Discontinuity-Preserving Dense Disparity Maps. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 33–40. Springer, Heidelberg (2005)
14. Li, X., Jackson, J.R., Katsaggelos, A.K., Mersereau, R.M.: Multiple Global Affine Motion Model for H.264 Video Coding with Low Bit Rate. In: Proc. SPIE VCIP, San Jose, CA (2005)
15. Coelho, A.M., Estrela, V.V., de Assis, J.T.: Error Concealment by Means of Clustered Blockwise PCA. In: IEEE Picture Coding Symposium, Chicago, IL, USA (2009)
16. do Carmo, F.P., Estrela, V.V., de Assis, J.T.: Estimating Motion with Principal Component Regression Strategies. In: Proc. of IEEE MMSP 2009, Rio de Janeiro, RJ, Brazil (2009)
17. Coelho, A.M., Estrela, V.V.: Data-Driven Motion Estimation with Spatial Adaptation. Intl. J. of Image Proc (IJIP) 6(1), 53–67 (2012)
http://www.cscjournals.org/csc/manuscript/Journals/IJIP/volume6/Issue1/IJIP-513.pdf

Computational Intelligence Applied to Competitiveness Evaluation of Supply Chains: An Adaptive Neuro-Fuzzy Model

Suelene de Jesus do Carmo Corrêa and Antônio Morais da Silveira

Federal University of Pará, Institute of Exact and Natural Sciences,
Augusto Corrêa. 01, 66075-110 PA, Brazil
{suelene,morais}@ufpa.br

Abstract. Technological advances and economic turmoil are some of the factors leading to increased competition in a global scale, which led companies, industries and countries to be concerned to maintain a leadership position in the market for competitive advantage. One way to achieve this goal is to make use of computer technologies to facilitate and accelerate decision-making in these environments of uncertainty. This work aims to show the use of a hybrid approach of Artificial Neural Networks and Fuzzy Logic, an Adaptive Neuro-Fuzzy system, to supply chain competitiveness evaluation. To validate the method is used a case of study based on the supply chain of broilers in Brazil. The results were satisfactory considering the low errors obtained in the validation tests.

Keywords: Neuro-Fuzzy, Competitiveness evaluation, Supply chains, Adaptive learning.

1 Introduction

Technological advances, globalization, mega-mergers and greater environmental awareness have led to changes in the global economy. Such changes imply increased competition, requiring organizations to create more effective competitive strategies, innovative solutions to stay alive in the market. Porter [1] states that competitive strategy is the search for favorable competitive position in an industry, to establish a profitable and sustainable position against the forces that determine the competition. This determination leads to the possibility of business expansion, diversification, or even survival of a company, especially if its action involves competitive markets.

This growing quest for competitive advantage generates the need for speed and efficiency in decision making in managerial environments, which has led to the development and use of increasingly advanced techniques to facilitate decision making. Between these techniques may be cited the Computational Intelligence techniques such as Artificial Neural Networks (ANN) and Fuzzy Logic, which allow the systems analysis and decision making in environments of uncertainty.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 658–669, 2012.

Regarding the problem of competitiveness evaluation in conjunction with the use of computational solutions were found some literature reports that are shown bellow.

Choy [2] presents a tool for intelligent supplies management, using Case Based Reasoning (CBR) and Artificial Neural Network technique to select and compare suppliers during the development of new products. Dall'Agnol [3] carried out an assessment of computer simulation to evaluate supply chain performance through collaboration and improvements in logistics processes. Silva [4] used computational intelligence using Genetic Algorithms Optimization Method and the Ant Colony Optimization for an online scheduling problem logistics. Manoj [13] used fuzzy logic based programming to solve the problem of selection of vendors in a supply chain. Silveira et al [7] performed an essay using Artificial Neural Networks to evaluate the competitiveness of organizations, using the approach similar to competitiveness proposed by [1] with its five competitive forces.

It is observed that competitiveness is a concern to many researchers, however most of the practical studies are focused on punctual issues of the problem, as seen in [2], [3] and [4], and not focused on a more deeply and real competitiveness as that defended by [5] and [6]. Even Silveira [7] which aimed analyzing the competitiveness of organizations considering internal and external factors, still treats the problem in a restricted way, adopting the approach of [1] which is just one component of the model proposed by [5] and [6] which was adopted in the development of the computational solution presented in this study.

Thus, the objective of this study is to present a technique that combines simulation and optimization based on Computational Intelligence, evaluating if it can be used for integration, learning and improvement of production chains, supporting the administration of them to meet their productivity goals and market competitiveness. This computational model must be applicable to various types of supply chains, and is implemented through the combined use of Neural Networks and Fuzzy System, a hybrid system, called Adaptive Neuro-Fuzzy System.

As a case of study for applying the solution it was chosen the supply chain of broilers, due to the importance of this activity in the scenario of national economy, and also because there are several theoretical administrative studies on this chain, which facilitates the analysis and validation of computational solution and also enhances its usefulness. The structure the supply chain of broilers can be seen in [8].

For the development of the generic model were used the drivers described by [9] and [10], which are representative for almost all activities or links of a supply chain, and are consistent with the approach of competitiveness proposed by [5] and [6]: institutional environment, technology, market structures, governance and coordination structures of the chain, company management, inputs, transport and storage.

The training of the model included 50 input and output patterns for each subfactor evaluated, obtained with the aid of specialists working at universities and Federal State of Para, in a total of 42 sub-factors. The model validation included 19 profiles related to the productive chain of broilers of Mato Grosso Brazil, obtained from a researcher who has studied this chain, work that can be seen in [8].

The paper is organized as follows: Section 2 presents the theoretical studies on the problem of evaluating the competitiveness of supply chains that support the computational solution. Sect. 3 presents the basic concepts of Adaptive Neuro-Fuzzy systems. Sect. 4 describes the methodology used to create the computational solution. details of the training system of the Neuro-Fuzzy. In sect. 5 are presented some of the results. Sect. 6 presents conclusions about the work, as well as suggestions for future work.

2 The Problem of Supply Chain Competitiveness Evaluation

Worldwide there are multidisciplinary studies that aim to define best approach for the assessment of competitiveness for countries, industries or companies. Some studies indicate competitiveness as dependent only on internal factors of the company; others only consider external factors [11]. However, experts on the subject agree that these approaches isolated do not allow a consistent evaluation of the competitiveness.

The German Development Institute (DIE), assumes a competitiveness assessment model proposed by Esser et al cited [6]. This model assumes four levels of analysis: Meta, Macro, Meso and Micro. At these levels, [6] joined the competitive model of [1] of the five competitive forces and then proposed a model integrating both by putting the Porter model at the micro level.

Each level of the model of [6] is composed as follows: **Macro level** - variables related to the national congress, national government, state institutions, central bank, judicial organs; **Meta level** - variables related to the orientation of the groups of actors for learning and efficiency, advocacy and self-organization into changeable conditions, ability of social organization and integration, capacity of groups of actors in strategic integration; **Meso level** - intermediate factors in the regional and community space: governments, business associations, labor unions, other private organizations, public and private institutions for research and development; **Micro level** - substitutes, buyers, existing competitors, input and output (Porter's model [1]).

Regarding the Brazilian industry and Kupfer Hasenclever [5] indicated that the competition is divided into three elements: business factors, structural factors and systemic factors. Siqueira and Fusco [12] say that a company must collect and analyze information on a variety of variables, markets, customers, technology, global finance and world economy in transition to formulate and implement their strategies properly.

Based on this scenario, the objective is the creation of a computational solution for evaluating the competitiveness of production chains, assuming it to a broader approach of competitiveness indicated in the model of [6] with its four levels, macro, meta, meso and micro, which also include the approach of [5], because these levels include business factors, structural factors and systemic factors.

3 Adaptive Neuro-Fuzzy System

The Artificial Neural Networks (ANN) are seen as non-linear parametric models. This methodology has the advantage of detecting any implicit non-linear relationship between the response variable and the explanatory variables [14] The neural processing is able to extract relations of input variables directly on the high-dimensional spaces that typically characterize them, making it a valuable tool in processing complex problems of pattern recognition. On the other hand, neural networks can work together with other processing techniques, allowing the use of the knowledge gained in a particular application area [15]. Detailed information about ANN ca been seen in [14].

Fuzzy Logic is a model based on the theory of Fuzzy Sets. In conventional or Boolean logic a proposition has two extremes: either it is completely true or completely false. However, in the fuzzy logic, a premise varies in degree of truth from 0-1, which leads to partially true or partially false. The control performed by the fuzzy logic mimics a behavior based on rules rather than a control strict to mathematical models, such as differential equations. The goal of fuzzy logic is to generate a logic output from a set of inputs not precise, with noise or even missing [16].

According [17] the fuzzy system are indicated when is desirable to automate processes that depend on the experience of human operators, allowing that the behavior of the control system approach to the human form of thinking. Humans are capable of dealing with complex processes based on inaccurate or approximate information. The fuzzy logic comprises a mathematical tool which allows the translation of these information in linguistic values, and the handling of them [18]. A generic fuzzy system can be seen in [19].

The unification of the Fuzzy Systems and Neural Networks in a common structure in the form of an adaptive network is called Adaptive Neuro-Fuzzy Inference System (ANFIS). These systems combine the fuzzy inference method with the learning ability of the ANN, where the ANN helps in generating the set of rules for the Fuzzy System [20].

The best known models of fuzzy inference engine are the Mandami, Sugeno and Tsukamoto model. The Sugeno model is flexible for combining subjective information from the human knowledge with objective information to the rules. This way it is appropriate to the use of adaptive techniques. Detailed information on Neuro-Fuzzy architectures can be seen in [20].

4 The Computational Solution

4.1 Development of the Computational Solution

Following the principles of Software Engineering, according [21] and [22], the development of the computational solution followed the following life cycle: analysis, design, development, training, validation and testing.

In the analysis phase were defined the functionalities of the solution, during the development has occurred the implementation of the functionality defined in the analysis phase. The modeling approach used by the Neuro-Fuzzy system is similar

to many system identification techniques. First, the structure of a parameterized model is created (related inputs, rules for outputs, membership functions and so on). Next, it is collected input and output data to be used in training. The Neuro-Fuzzy system is then used to emulate the training data presented to it , modifying the parameters of the membership function according to a error criterion selected.

The initial parameters were obtained by searching the literature [6], [8], [9], [10], [20], and with the assistance of experts that indicate the best drivers and subfactors that are representative of a supply chains and their importance or weight to make the inference performance.

The developed solution can be customized to suit different production scenarios, by assigning weights to subfactors or even adding subfactors and drivers according to the particular scenario under consideration.

Fig. 1 shows the general structure of the solution Neuro-Fuzzy created, showing the dependence of the stages which must be met by the functionalities oh the computational solution defined in the analysis phase to achieve the ultimate goal, which is the diagnosis and suggestions for improvement of the supply chain.

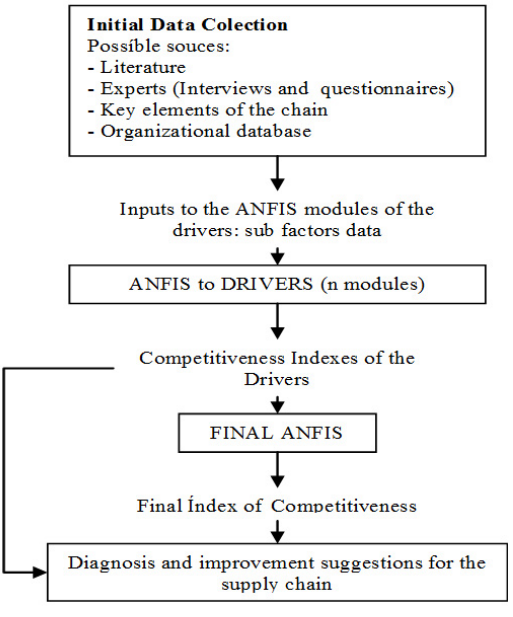


Fig. 1. General structure of the computational solution

The initial implementation of the model was made using the Toobox Neuro-Fuzzy of MATLAB software VR2006a [1], and has been implemented by scripts because the graphical tool of Matlab does not allow the training of several modules ANFIS at the same time, and, as can be seen in Fig.1, the designed model

[1] MATLAB. Simulation Software. The Language of Technical Computing. http://www.mathworks.com/products/matlab

is formed by one ANFIS module for each driver of competitiveness, and the output of these modules are the entries to the Final ANFIS module which defines the final index of competitiveness. The final implementation will be in Java programming language.

After the solution Neuro-Fuzzy be properly tested and validated it must be applied in the evaluation of competitiveness, generating the diagnostic of the status of a pro-ductive scenario, providing the final competitiveness index of each driver, as well as the final competitiveness index, indicating the strengths and weaknesses of that sce-nario. The solution must also provide graphics to facilitate understanding of the diagnosis. From the diagnostics generated by the Final ANFIS, the computational solution must provide appropriate suggestions for improvements the performance of the chain according to the problems identified, taking into account the expert analysis on each driver dimension, analysis that were obtained during the data collection.

For the development of the Neuro-Fuzzy model it were used the drivers described by [9] and [10], which are representative for almost all activities or links of a supply chain. It is Important to detach that each driver has n subfactors and these subfactors can also be subdivided, as shown below.

Institutional Environment. The institutional environment consists of rules imposed by legislation, by society and politics that surround the firm. Includes:

- Macroeconomic conditions: Interest rates; Exchange rate, Inflation.
- Sector programs and policies: Availability of credit; Access to credit; Differentiated interest rate.
- Taxes: Export taxes; Social security taxes; Internal taxes; Corporate income tax.
- Product Security: Health Legislation; Sanitary inspection services; Weather conditions.
- Foreign Trade Policies: Tariff barriers; Non-tariff barriers; Commercial agreements.

Technology. The technology evaluation should consider the methods, processes, facilities and equipments used in operations, aspects related to research and development, adaptability of technology and adoption of technological standards [10]. Includes: Dissemination of key technologies; Investment in R & D; Number of experimental stations; Number of patents.

Coordination and Governance Structures. Governance and coordination structures are the different organizational forms that influence the transactions within a supply chain. [10]. Includes: Contracts between companies; Sector organizations.

Company Management. The efficient management of the firm must use tools to identify market signals and allocate resources optimally [8]. Includes: Use of infor-mation technology, Strategic planning; costing systems; Quality Control, Marketing.

Inputss. This category includes factors related to the processing the raw material and their acquisition. Includes: Prices of inputs, Cost of labor; Price of land, Availa-bility of land.

Storage and Transport. Storage refers to the ability to maintain products available for purchase and transportation and is related to the forms of the production flow. How: Grain storage; Storage of food to chicken; Meat storage; Condition of highways; Capacity of the highways.

4.2 The Training of the Solution

During this phase are presented to the model the patterns of input and output (supervised training), the initial parameters of training, the adjustment of parameters is made during the evolution of the training in accordance with optimization methods such as backpropagation and least squares.

For the training of the model were used 50 patterns of input and output to each subfactor according to the following conditions: score for each subfactor respect a likert scale which ranges from -2 to 2, which is related to linguistic values, where -2 means very unfavorable and 2 means very favorable.

The inputs to the model training are the notes of each subfactor identified for the drivers that represent the chain, these values represent the judgment of key supply chain members (those who have more representation), and also based on expert opinion.

The weight of the subfactors represents the importance of each one in relation to its driver, and ranges from 0 to 1, where 0 means the least important and 1 means the most important, the sum of the subfactors of each driver must totalize 1, these weights are used in training by means of calculating the desired outputs for each one of the 50 training profiles, by calculating the weighted average values of each subfactor that comprises inputs, by their respective weights.

During the training, the outputs generated by the n ANFIS modules of drivers that is the competitiveness index of each driver serve as entry of training to the final ANFIS that is responsible for calculating the Final Index of Competitiveness of the chain. The desired output of this module is also provided by applying the weighted average of the indexes of each driver by their weights (importance in relation to the final competitiveness index).

The Neuro-Fuzzy system of greater efficiency based on tests performed included the following parameters:

- Sugeno inference machine;
- Triangular membership functions (using three subsets);
- Hybrid optimization training method (backpropagation + least squares);
- Step Size=0.1;
- Measure of deviation: Root Mean Squared Error (RMSE), Desired RMSE is 0 (zero).

The Figures 2(a), 2(b) and 2(c) show the results of the training module of for the final competitiveness evaluation. The information displayed by the figures demonstrates success in training, as specified below.

The system Neuro-Fuzzy training RMSE error obtained practically zero, as can be seen in Fig. 2.a, throughout all epochs of training the error remained minimal. The validation demonstrated the generalization capability of the model, because the validation error was also low, throughout the epochs.

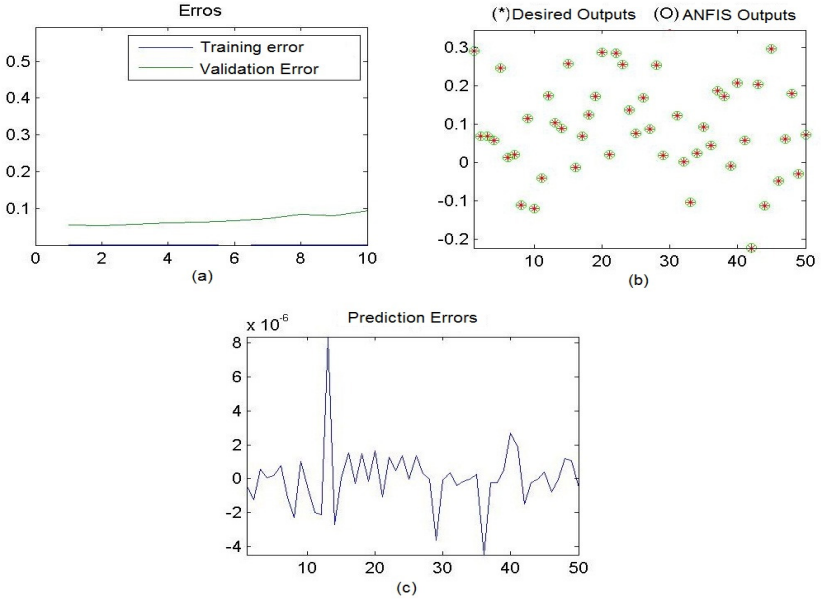


Fig. 2. (a) Training x Validation. (b) Comparison between desired output and ANFIS output. (c) Error evolutions during the training.

The Fig. 2(b) shows the result of training during the presentation of 50 training patterns , in this figure can be seen that the output generated by ANFIS implemented almost overlaps the desired output.

The Fig. 2(c) enhances the efficiency of training showing the evolution of errors throughout the presentation of the 50 input patterns, in this phase occurs the adjustment of the model parameters for maximum optimization; the prediction errors obtained during the training were in the order of 10^{-6}, ie, very close to the optimal error established a priori (RMSE = 0). This explains why the desired outputs and the out-puts generated by Neuro-Fuzzy solution during training scores were nearly coincident, as shown in Fig. 2(b).

4.3 Validation

In some cases the training data may be corrupted or not correctly represent the mod-eled scenario, so it is useful to validate the model. The model validation is the process by which a set of inputs and desired output that were not used

in training the model is presented, to see if the model adequately predicts the output values , as a way to test the generalization capability of the resulting system after training.

The validation is also necessary, because after a certain point of training the model begins to decorate the set of training data, losing the generalization capability. At the beginning of training the error decreases, but when the overfitting problem happens the error only increases. The lack of generalization is identified by comparing the errors obtained with the training set and the validation set, so you can properly choose the point where the model best fits the problem under study, and the best settings for the model to achieve the desired goal. The supply chain chosen to validate the model was the productive chain of broilers, according what was quoted in Sect. 2.

The model validation included 19 profiles related to the drivers representative of the productive chain of broilers of Mato Grosso Brazil, the data were obtained from a researcher who has studied this chain, work that can be seen in [8].

The Fig. 3 presents the result to one of the profiles used in training. This result is composed by the scores achieved by seven drivers (ANFIS Output)) , by the Final Competitiveness Index, the figure also shows the comparison of the evaluation generated by for one of the training profiles with the desired output, which is calculated based on the weights defined by specialists each subfactor and drivers.

The Final Competitiveness Index (FCI) provides an overview of the competitiveness performance. Details of the favorable and unfavorable factors of the chain can be analyzed by means of scores of each driver set used in the solution, as can be seen in Fig. 3, which shows the result obtained for the seven drivers established.

In Fig. 3 it is possible to observe that the values obtained for the competitiveness indexes of each of the seven drivers were predicted quite accurately, because the values generated by Neuro-Fuzzy system adaptive learning were nearly coincident with the desired output values generated by the application of mathematical formulations directly on the input data.

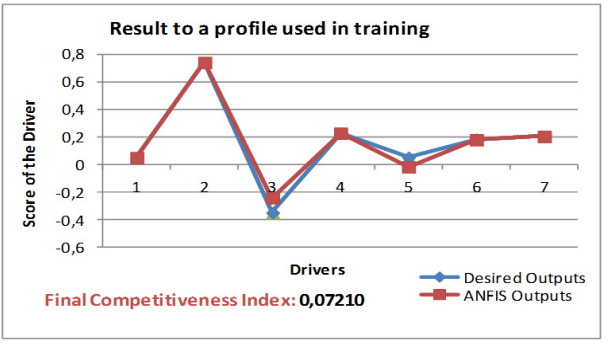


Fig. 3. Example of evaluation generated by the Neuro-Fuzzy model

5 Results Discussion

In this study were tested various configurations for the Neuro-Fuzzy model; from these tests it was observed that the membership functions that generated the best performance were triangular with three subsets for each subfactor.

The optimization algorithm backprobagation offered good results in training, but the hybrid (least squares + backpropagation) yielded better results with lower error values, in both training and validation.

After training, the solution was able to simulate the scenario under study, receiving the input data, treating them appropriately and generating the evaluation through the competitiveness indices of each driver, through the final index of competitiveness of the chain and also through the graphics that are representative of diagnosis.

The example in Fig.3, shows that most drivers of the profile in question have competitiveness considered neutral, and the driver 3 (Market Structures) has been rated as critical, requiring special attention. The scores of the seven drivers together with the Final Competitiveness Index, also neutral, show that this example would be of a chain with low competitiveness.

6 Final Considerations

The results generated by the solution were satisfactory considering the low error obtained in the validation tests. It was found that the Neuro-Fuzzy approach can be a valuable tool for use in solving problems in uncertain environments. The proposed solution can be adapted to other productive chains, considering that the parameters of evaluation may be the same as were used in this case study.

The resources offered by the tool are intended to assist the evaluator of the production chains in the actions aiming at improving the competitiveness indexes of the chain through the monitoring of drivers and subfactors that characterize the performance of its environment. Thus, knowledge of these factors, and how they impact the competitive performance of the supply chain are of great importance for establishing business strategies and public policies to improve its competitiveness.

The functionality of the solution that deals with the generation of suggestions for improvement based on evaluation subfactors and drivers generated by the Neuro-Fuzzy model has not yet been implemented.

6.1 Future Works

The simulation of the solution designed in this work were implemented experimentally using tools of the MATLAB software, but final implementation of the solution Will be made using Java programming language due to its 'characteristics such as: simplicity; robustness; portability (application is platform independent); secure; concurrent (supports concurrent applications).

It is aimed the creation of a robust database to be used integrated with the Java application, allowing that the collection of initial information, like the opinion of experts and key elements of the supply chain, can be done directly through the graphical interface of the solution, and also that the generation of reports results, such as the suggestions for improvements to supply chain, can be done in a manner more automated and well organized due to various graphical features offered by Java.

It is intended the use of the computational solution to trace the competitive profile of the chain of broilers in the state of Par and the proposition of improvements.

References

1. Porter, M.: Estratégia Competitiva: Técnicas para Análise de Indústrias e da concorrência, 2nd edn. Campus, São Paulo (2005)
2. Choy, K.L., Lee, W.B., Victor, L.: An intelligent supplier management tool for benchmarking suppliers in outsource manufacturing. International Journal of Expert Systems with Applications 22, 213–224 (2002)
3. Dall'Agnol, W.J.: Avaliação de simulação computacional na avaliação de desempenho da cadeia produtiva através da colaboração e melhorias de processos logísticos. Dissertação de Mestrado, Departamento de Engenharia de Produção, Pontifícia Universidade Católica do Paraná. Brasil (2005)
4. Silva, C.A., Sousa, J.M.C., Runklerb, T.A.: Rescheduling and optimization of logistic processes using GA and ACO. International Journal of Engineering Applications of Artificial Intelligence 21, 343–352 (2008)
5. Kupfer, D., Hasenclever, L. (org.).: Economia industrial: fundamentos teóricos e práticas no Brasil. Rio de Janeiro: Ed. Campus (2002)
6. Rosseto, C.R.: Uma proposta para combinação do modelo de Porter e do modelo de referência do Instituto Alemão de Desenvolvimento (IAD) no estudo da competitividade sistêmica setorial. In: VII SIMPEP, Bauru, SP, Brasil (2000)
7. Silveira, A.M., Silva, M.L.C., da, O.R.C.L.: de.: Evaluation of the competitiveness of organizations: An essay using Artificial Neural Networks (ANN). In: Torres, G.L., Abe, J.M., Mucheroni, M.L., Cruvinel, P.E. (eds.) Advances in Intelligent Systems and Robotics, 1st edn., vol. 101, pp. 145–158. IOS Press, Amsterdam (2008)
8. Melz, L.J.: Competitividade da cadeia produtiva de carne de frango em Mato Grosso: Avaliação dos seguimentos de Avicultura e Processamento. Dissertação de Mestrado, Departamento de Engenharia de Produção, UFSCar, São Paulo, SP, Brasil (2010)
9. Van Duren, E., Martin, L., Westgren, R.: Assenssing the competitiveness of Canada's agrifood industry. Canadian Journal of Agricultural Economics (39), 727–738 (1991)
10. Silva, C.A.B., Batalha, M.O. (Coord.). Estudo sobre a eficiência econômica e competitividade da cadeia agroindustrial da pecuária de corte no Brasil. Brasília: IEL: CNA: SEBRAE (2000)
11. Siluk, J.C.M., Nora, L.D.D.: Proposta de diagnóstico da inovação e competitividade no setor de serviços. RACE (Revista de Administrao, Contabilidade e Economia) 10(1), 7–30 (2011)

12. Siqueira, M.C.M., Fusco, J.P.A.: Avaliação de competitividade sob o foco de redes simultãneas: alianças estratégicas e parcerias na indústria gráfica. RAU (Revista de Administração da UNIMEP) 3(1), 94–116 (2005)
13. Manoj, K., Prem, V., Shankarc, R.: A fuzzy goal programming approach for vendor selection problem in a supply chain. International Journal of Computers & Industrial Engineering 46, 69–85 (2004)
14. Haykin, S.: Redes Neurais: Princípios e Práticas, 2nd edn. Bookman, Porto Alegre (2001)
15. Santos, M.A., Seixas, J.M., Pereira, B.B.P., Medronho, R.A.: Usando redes neurais artificiais e regressão logística na predição da Hepatite A. Revista Brasileira de Epidemiologia 8(3), 117–126 (2005)
16. Wang, L.: A Course in Fuzzy Systems and Control. PrenticeHall, Englewood Cliffs (1997)
17. Shaw, I.S., et al.: Controle e modelagem Fuzzy. Edgard Blacher Ltda, São Paulo (1999)
18. Tanscheit, R.: Sistemas Fuzzy.: Monografia, Graduação, Departamento de Engenharia Elétrica - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro (2002)
19. de Souza, R.A.: Sistemas Inteligentes de Apoio a Decisâo Utilizando Lógica Fuzzy. Dissertação (Mestrado em Engenharia Elétrica), DEEC, UFPA, Pará (2003)
20. Jang, J.R., Sun, C.: Neuro-Fuzzy Modeling and Control. Proceedings of the IEEE 83(3), 378–406 (1995)
21. Pressman, R.S.: Engenharia de Software, 6th edn. McGraw-Hill, São Paulo (2006)
22. Sommerville, S.: Software engineering, 8th edn. Addison-Wesley (2006)

Appendix: Useful Materials

The files with the training data, a model of script for creation, training and validation of Neuro-Fuzzy model created in this study, and also the table that presents data and results of profile example in Fig. 3 can be obtained at the following web address that are maintained by the authors:
https://www.dropbox.com/sh/423x73jsqiyvzx2/FP0W_a2iql/ANFIS_data_englishVersion.rar.

Experimental Comparison of DWT and DFT for Trajectory Representation

Ronald Annoni Jr. and Carlos Henrique Quartucci Forster

Instituto Tecnológico de Aeronáutica,
12228-900, São José dos Campos, SP, Brazil
{annoni,forster}@ita.br

Abstract. In this work Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT) were experimentally evaluated for their performances as tools for dimensionality reduction in a real data set of air traffic trajectories. Results showed that both DFT and DWT were able to provide very expressive reduction for trajectory representation with minimal loss of information. Overall, DWT performed slightly better requiring fewer coefficients than DFT to achieve the same signal energy or to provide the same quality of reconstruction of the trajectories.

Keywords: trajectories, DFT, DWT.

1 Introduction

Given the wealth of trajectory data made available from mobile wireless technologies, radio frequency, satellite, radar and other devices or sensors, there has been a growing interest in trajectory data mining. Many applications such as video surveillance [1], climate [2], animal behavior [3], location-based services [4] and traffic control [5], use statistical methods and artificial intelligence combined with database management to extract information from a set of trajectories.

To extract useful information from a large amount of trajectories, a representation model is necessary to compare them efficiently. Usually, such model should consist of a small number of trajectory features that are, at the same time, representative and distinctive.

Literature on this subject provides some options. Knorr *et al.* [6] represent the trajectories by using the coordinates of the starting and ending points; the average, minimum, and maximum values of the directional vector; and the average, minimum, and maximum velocities. Though compact, this approach disregards a lot of trajectory information. Lee *et al.* [7] build a model with line segments between consecutive characteristic points determined by a minimum description length (MDL) principle for the trajectories. This scheme has the interesting property of keeping only the relevant points where the trajectory changes its behavior. Eckstein [9] and Gariel *et al.* [8] use principal component analysis (PCA) over resampled trajectories.

Most of the representation models found in the literature try to cope with the complex spatio-temporal characteristics of the trajectories by dimensionality

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 670–677, 2012.

reduction and length equalization. However, surprisingly enough, to the best of our knowledge there is little to none reference to the use of *Discrete Fourier Transform* (DFT) or *Discrete Wavelet Transform* (DWT) for trajectory feature extraction. One rare example is Naftel *et al.* [10], that considers applications for event detection and surveillance and proposes to model trajectories using the lowest order Fourier coefficients obtained by Discrete Fourier Transform.

DFT and DWT have a long history of utilization in time-series and signal processing, especially for compression and reconstruction of signals. As such, their application in trajectory representation is almost immediate. In this paper we compare the results of the utilization of DFT and DWT for trajectory feature extraction. For this study we used real trajectories of aircraft operations in a terminal area including approach/landing and takeoff/departure with the objective of evaluate the performance of each one in terms of dimensionality reduction with minimal loss of information.

2 Transforms Background

A transform is a mathematical operation utilized to obtain a set of coefficients c_n that describes a function $f(x)$ in terms of a set of orthogonal functions $\phi_n(x)$:

$$f(x) = \sum_{n=1}^{\infty} c_n \phi_n(x). \tag{1}$$

Multiplying both sides of (1) by $\phi_m(x)$ and integrating gives [11]

$$\int f(x)\phi_m(x)dx = \sum_{n=1}^{\infty} c_n \int \phi_m(x)\phi_n(x)dx. \tag{2}$$

As we postulate $\{\phi_n(x)\}$ $(n = 1, 2, \ldots)$ to be an orthogonal system, then

$$\int \phi_m(x)\phi_n(x)dx = 0, \quad \text{for } m \neq n, \tag{3}$$

which reduces (2) to

$$\int f(x)\phi_m(x)dx = c_m \int [\phi_m(x)]^2 dx. \tag{4}$$

The integral on the rigth of (4) is a certain constant B_m that is assumed not to be zero. Then

$$c_m = \frac{1}{B_m} \int f(x)\phi_m(x)dx. \tag{5}$$

If the basis functions $\{\phi_m(x)\}$ $(m = 1, 2, \ldots)$ are a trigonometric system $(1, \cos x, \sin x, \ldots, \cos nx, \sin nx, \ldots)$, then the set of coefficients c_m will be called the *Fourier coefficients* of $f(x)$. In case the basis functions are a wavelet system, then the set of coefficients c_m will be called the *wavelet coefficients* of $f(x)$.

2.1 Discrete Fourier Transform

Trajectories on a plane can be represented as a sequence $\mathbf{s}[t]$ of points $(x[t], y[t])$. If we consider the component sequence $x[t]$ (or $y[t]$) periodic with period N so that $x[t] = x[t + rN]$ then it can be represented by a Fourier series that corresponds to a sum of harmonically related complex exponential sequences, with frequencies that are integer multiples of the fundamental frequency $(2\pi/N)$ and has the form [12]

$$x[t] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)kt} \tag{6}$$

where N can be interpreted as the length of the trajectory and $X[k]$ are the Fourier coefficients.

By making use of the orthogonality of the complex exponentials, we can develop (6) in a way similar to the development of (1) to (5) at the beginning of this section so that the coefficients $X[k]$ are given by the relation

$$X[k] = \sum_{t=0}^{N-1} x[t] e^{-j(2\pi/N)kt} \tag{7}$$

which is called the *Discrete Fourier Transform* (DFT) of the sequence $x[t]$, while (6) is its inverse.

The great appeal of using the DFT for reducing the dimensionality can be seen in the example shown in Fig. 1. Most of the energy is concentrated in a few DFT coefficients at the ends of the sequences representing the low frequencies.

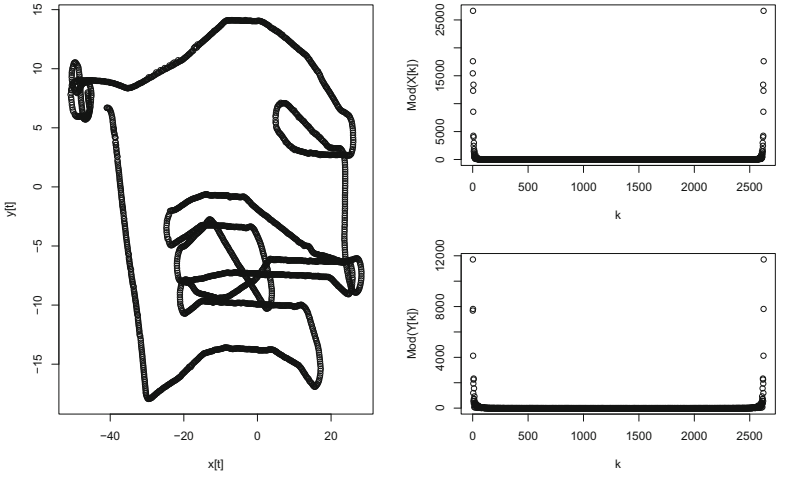


Fig. 1. A real aircraft trajectory and the absolute value of the DFT coefficients

2.2 Discrete Wavelet Transform

The word wavelet means a "small wave" that exists only in a finite domain and is zero elsewhere. The wavelet transform designates the convolution of a signal with the scaled and shifted instances of a wavelet that results in a collection of time-frequency representation of the signal in various resolutions. The general discrete representation of a wavelet is given by [13]

$$\psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}}\psi\left(\frac{t - k\tau_0 s_0^j}{s_0^j}\right), \tag{8}$$

for which it is usually chosen $s_0 = 2$ and $\tau_0 = 1$ so that sampling of the frequency and time correspond to dyadic sampling.

Wavelet multiresolution analysis (MRA) [14] requires two basic functions: a scaling function

$$\varphi(t) = \sum_n h(n)\sqrt{2}\varphi(2t - n), \quad n \in \mathbb{Z} \tag{9}$$

where $h(n)$ are the scaling function coefficients, and a wavelet function

$$\psi(t) = \sum_n h_1(n)\sqrt{2}\varphi(2t - n), \quad n \in \mathbb{Z} \tag{10}$$

for some set of coefficients $h_1(n)$. A signal $g(t)$ could then be written as

$$g(t) = \sum_k c_{j_0}(k)\varphi_{j_0,k}(t) + \sum_k \sum_{j=j_0}^{\infty} d_j(k)\psi_{j,k}(t). \tag{11}$$

The coefficients $c_{j_0}(k)$ and $d_j(k)$ are called the *Discrete Wavelet Transform* (DWT) of the signal $g(t)$. As was the case with the DFT coefficients, Fig. 2 shows that DWT coefficients also drop off rapidly and so is a good tool for dimensionality reduction.

3 Method and Experimental Setup

The data set utilized in our experimental comparison is composed of real aircraft trajectories from a terminal area. It contains a sample of 330 trajectories totaling 42,696 points of radar detections projected on a plane which covers a square area of 40 miles side. It includes takeoff/departure and approach/landing procedures from/to an airport located at the center (coordinates 0,0) of the area, with a runway oriented on 17/35 headings. The quantity of points per trajectory is summarized in Table 1.

Our evaluation consisted of two main parts. In the first part we made use of Parseval's theorem [12,13] that relates the energy of a signal $s[t]$ with the energy of the transform $S[k]$ by

$$\sum_t |s[t]|^2 = \sum_k |S[k]|^2. \tag{12}$$

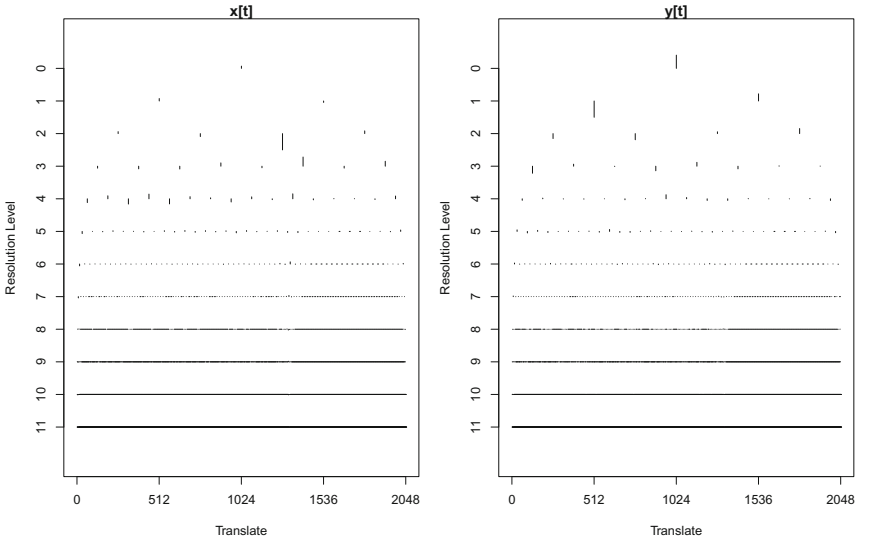


Fig. 2. DWT decomposition from the trajectory shown in Fig. 1

Table 1. Summary of quantity of points per trajectory

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
71	84	110.5	129.4	160	378

We calculated the energy of each trajectory as the sum of the squares of their components. Then we computed DFT and DWT for each trajectory and arranged the coefficients in descending order of their absolute values. Finally we determined how many coefficients from each transform and for each trajectory were necessary to achieve some stipulated percentual values of total energy.

In the second part of our experimental evaluation we tested the performance of each transform for trajectory reconstruction. Given an original trajectory

$$T = \{(x_0, y_0), (x_1, y_1), ...(x_{K-1}, y_{K-1})\} \tag{13}$$

and the reconstructed trajectory using the first d greatest coefficients

$$T'(d) = \{(x'_0, y'_0), (x'_1, y'_1), ...(x'_{K-1}, y'_{K-1})\} \tag{14}$$

the root mean square deviation between T and $T'(d)$ is given by

$$rmsd(T, d) = \sqrt{\frac{1}{K} \sum_{i=0}^{K-1} [(x_i - x'_i)^2 + (y_i - y'_i)^2]} \tag{15}$$

with K being the number of points in the trajectory. The value of $rmsd$ is maximum when we use only one coefficient for reconstruction ($d = 1$) and it

decreases as we increment d until it reaches zero at $d = K$. As we seek the least $rmsd$ value using the smallest possible number of coefficients, a penalty function $P(d)$ is defined which increases linearly with the number of coefficients d used for trajectory reconstruction such that $P(1) = 0$ and $P(K) = rmsd(T, 1)$:

$$P(d) = \frac{rmsd(T,1)}{(K-1)}(d-1). \tag{16}$$

Then for our given trajectory data set the problem is to find the value D_i for each trajectory T_i such that

$$D_i = \underset{d \in \{1..K_i\}}{\operatorname{argmin}} [rmsd(T_i, d) + P(d)]. \tag{17}$$

We repeated the above operation for both DFT and DWT, for every trajectory covering all number of coefficients. The DFT was computed with package "stats" and DWT was computed with package "wavthresh" [16] both in the *R statistical environment* [15].

4 Results

The results from the first part of our experiments can be seen in Fig. 3. The graphs in this figure show the percentage of the total of the trajectories that required a determined number of coefficients to achieve the specified percentage of total energy for the DWT and for the DFT.

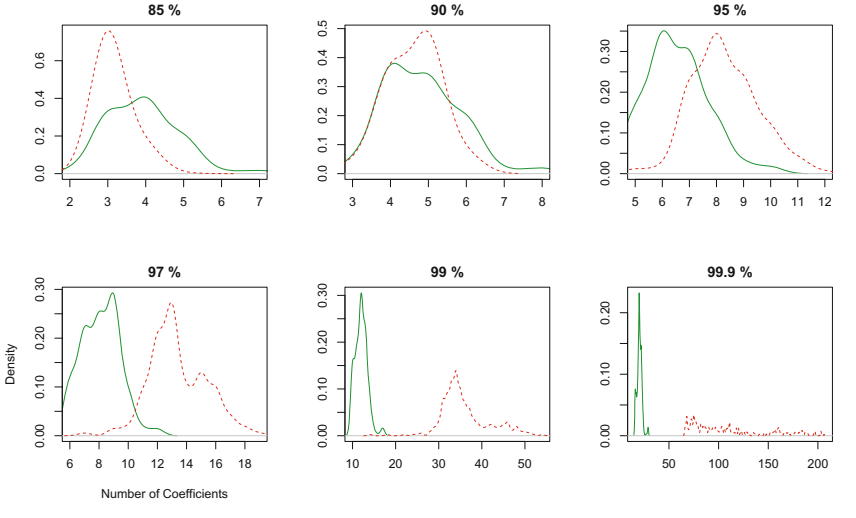


Fig. 3. Comparative density plots of the number of coefficients required for DWT(solid lines) and DFT(dashed lines) to achieve the stipulated percentual of total energy

The first graph in Fig. 3 shows that the DFT needs initially a smaller number of coefficients to reach 85% of the total energy.

In other words, considering that the area below the curves represents the percentage of the number of trajectories, in the first graph in Fig. 3 one can observe that at 85% of total energy DFT holds most of the trajectories in the data set with up to four coefficients while DWT includes only approximately half of the trajectories with the same conditions.

However, as the energy requirement increases, the necessary number of DFT coefficients increases faster than for the DWT, which remains consistently below 25 while the number of DFT coefficients for some trajectories may exceed 200 at 99.9% energy.

The second part of our experiments analysed the performance of the transforms for trajectory reconstruction. The two graphs in Fig. 4 summarizes the results obtained with the 330 trajectories in our data set. They show that in spite of our data set contains trajectories with an average of about 130 points and up to 378 points, both the DWT and DFT were able to perform the reconstruction using less than 35 coefficients. DWT had a small advantage because it used less coefficients for almost all of the trajectories reconstructions and also kept all values virtually below 25.

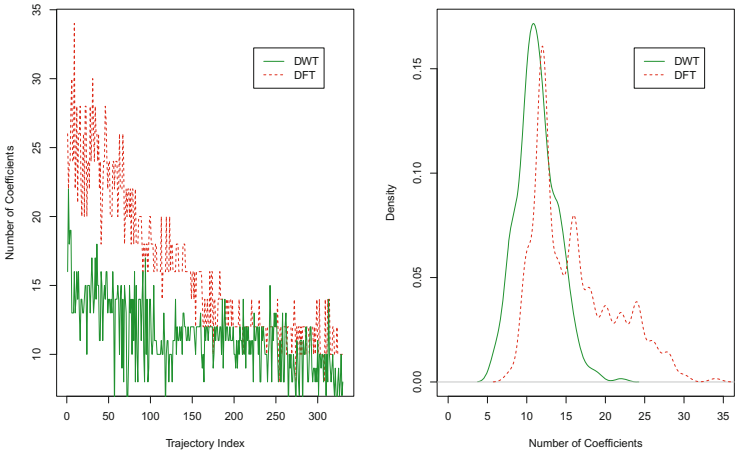


Fig. 4. Number of coefficients used for trajectory reconstruction

5 Conclusions

Trajectory data mining must deal with the difficulties imposed by the high dimensionality of the data. In the literature there are several options to address this problem but there is little use of transforms that are already getting good results in other areas for a long time. This paper presented an experimental comparison of the performance of DFT and DWT for dimensionality reduction on a

real data set of trajectories of aircraft. The results showed that both provide a good dimensionality reduction rate with a slight advantage of DWT over DFT.

References

1. Piciarelli, C., Foresti, G., Snidaro, L.: Trajectory clustering and its applications for video surveillance. In: IEEE Conf. Advanced Video and Signal Based Surveillance, pp. 40–45. IEEE Computer Society, Los Alamitos (2005)
2. Kim, H.-S., Kim, J.-H., Ho, C.-H., Chu, P.-S.: Pattern classification of typhoon tracks using the fuzzy c-means clustering method. Journal of Climate 24(2), 488–508 (2011)
3. Spampinato, C., Giordano, D., Di Salvo, R., Chen-Burger, Y.-H.J., Fisher, R.B., Nadarajan, G.: Automatic fish classification for underwater species behavior understanding. In: Proc. First ACM Int. Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams, ARTEMIS 2010, pp. 45–50. ACM, New York (2010)
4. Lee, J.W., Paek, O.H., Ryu, K.H.: Temporal moving pattern mining for location-based service. Journal of Systems and Software 73(3), 481–490 (2004)
5. Melo, J., Naftel, A., Bernardino, A., Santos-Victor, J.: Detection and classification of highway lanes using vehicle motion trajectories. IEEE Trans. Intelligent Transportation Systems 7(2), 188–200 (2006)
6. Knorr, E.M., Ng, R.T., Tukanov, V.: Distance-based outliers: algorithms and applications. VLDB Journal 8(3), 237–253 (2000)
7. Lee, J.G., Han, J., Whang, K.Y.: Trajectory clustering: a partition-and-group framework. In: Proc. 2007 ACM SIGMOD Int. Conf. Management of Data, pp. 593–604 (2007)
8. Gariel, M., Srivastava, A.N., Feron, E.: Trajectory Clustering and an Application to Airspace Monitoring. IEEE Trans. Intelligent Transportation Systems 12(4), 1511–1524 (2011)
9. Eckstein, A.: Automated flight track taxonomy for measuring benefits from performance based navigation. In: Integrated Communications, Navigation and Surveillance Conference (2009)
10. Naftel, A., Khalid, S.: Motion Trajectory Learning in the DFT-Coefficient Feature Space. In: Proc. IEEE Int. Conf. Computer Vision Systems (2006)
11. Kaplan, W.: Advanced Calculus. Addison-Wesley (1971)
12. Oppenheim, A.V., Schafer, R.W.: Discrete-Time Signal Processing. Pretice-Hall (1989)
13. Burrus, C.S., Gopinath, R.A., Guo, H.: Introduction to Wavelets and Wavelet Transform: A Primer. Prentice-Hall (1998)
14. Mallat, S.G.: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Trans. Pattern Analysis and Machine Learning 11(7), 674–693 (1989)
15. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2011) ISBN 3-900051-07-0, http://www.R-project.org/
16. Nason, G.: wavethresh: Wavelets statistics and transforms. R package version 4.5. (2010), http://CRAN.R-project.org/package=wavethresh

Classification Rule Discovery with Ant Colony Optimization Algorithm

Samaira Hodnefjell[1] and Ilaim Costa Junior[2]

[1] Federal University of Juiz de Fora, Computer Science Department, Brazil
[2] Fluminense Federal University, Institute of Computing, Computer Science Department, Brazil
samira.cruz@gmail.com, ilaim@ic.uff.br

Abstract. Ant Colony Algorithms has been successfully applied to solve combinatorial optimization problems. Subsequently applications on Data Mining (DM) appeared, more specifically aiming to solve classification problems. The Ant-Miner [3] algorithm is a good example of a solution to this problem. This algorithm is better than C4.5 [7] and CN2 [8]. This paper presents a new algorithm which applies an innovative modeling of the foraging behavior of ants [4] to the Ant-Miner. As a result of this adaptation, four different versions of the Ant-Miner algorithm were generated, tested and compared to the original version using seven public domain data sets. One of the versions produced comparatively superior results in terms of predictive accuracy in different system configurations.

Keywords: Ant colony optimization, classification task, data mining, knowledge data base discovery.

1 Introduction

Since the data started being collected and stored on magnetic media, along with the advance of technology that allows storing an increasing volume of data, until today, we have seen a quantitative explosion of databases. In this scenario, a problem is eminent: analyzing this amount of data.

A solution that emerged from that necessity is Data Mining. Its concept is associated with Knowledge Discovery in Data Bases.

Data Mining is a interdisciplinary field that relates techniques of machine learning, statistics and data bases. Its goal is to extract knowledge or, structural patterns, from data as a tool to help explain and make predictions through them.

Such knowledge must be precise and comprehensible to the user, so that it can be used to making decisions. Thus, the user should be able to interpret and evaluate the knowledge acquired.

There are several data mining tasks such as classification, association, regression and clustering. Each one of them represents a specific type of problem that demands a certain type of algorithm for solving it. In these terms, to design a data mining algorithm, it is important first to define what task the algorithm is intended to. This paper will focus on the classification task.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 678–687, 2012.

The term classification refers to any context in which a decision or prediction is made based on information available at any given time. A classification algorithm should be able to make such judgments in new situations.

An example of classification task, extracted from [5], can be seen on table 1 and figure 1 below.

Table 1. Data for a classification system

Sex	Country	Age	Buy (goal)
Male	*France*	*25*	*yes*
Male	*England*	*21*	*Yes*
Female	*France*	*23*	*Yes*
Female	*England*	*34*	*Yes*
Female	*France*	*30*	*No*
Male	*Germany*	*21*	*No*
Male	*Germany*	*20*	*No*
Female	*Germany*	*18*	*No*
Female	*France*	*34*	*No*
Male	*France*	*55*	*No*

IF (Country = "Germany") **THEN** (Buy = "No")
IF (Country = "England") **THEN** (Buy = "Yes")
IF (Country = "France" **AND** Age ≤ 25) **THEN** (Buy = "Yes")
IF (Country = "France" **AND** Age > 25) **THEN** (Buy = "No")

Fig. 1. Classification rules discovered from the above input data

For the classification task the database is divided into two subsets: a training set (where the class is known) and a test set (the class is unknown by the data mining algorithm). The discovered rules are evaluated in the test set [6]. The goal is to make predictions on the unseen database during training.

The reminder of this paper is organized as follows. Section 2 discusses ant colony optimization. Section 3 presents a review of the original Ant-Miner algorithm. Section 4 describes the proposed modifications in Ant-Miner algorithm. Section 5 reports computational results evaluating the proposed algorithm, concludes the paper and suggests future work.

2 Ant Colony Optimization (ACO) Algorithm

An ACO algorithm is an agent-based system that mimics the natural behavior of an ant colony, including mechanisms of cooperation and adaptation. The use of such system as a metaheuristic was proposed in [1] and [2] to solve combinatorial

optimization problems. This metaheuristics has proved be versatile and robust, having been applied successfully to a range of different problems.

Ants are extremely simple beings that have a limited capacity to process and exchange information. To reduce their efforts and exposure to hazards, it is important that they find the shortest path between the nest and a food source. Furthermore, it is important to note that path must be found without using any hierarchical order.

In fact, in most cases, the ants can find this path and also adapt to changes in the environment without using visual information. This capacity has been studied extensively by ethologists. Their studies reveled that, in order to exchange information about which path should be followed, many species of ants, like other social insects communicate with each other through trails of a chemical substance called pheromone. While traveling between the nest and the food source, the ants deposit a certain amount of pheromone on the ground, marking the path with a trail of this substance. The other ants have the ability to perceive the presence of the pheromone and tend to follow the path where the concentration of pheromone is higher. Thus, ants can carry food to the nest very effectively.

When a path becomes obsolete by the depletion of food supply, the amount of pheromone that marks this path is gradually reduced by evaporation, making it less attractive. This is a process of negative feedback that helps the ants to detect new paths.

Thus the ants are able to converge to a shortest path without direct communication, without visual information and no hierarchical order. They modify the environment by depositing pheromone, thus establishing an indirect communication that will influence the behavior leading to a well defined choice of the best path to follow. This characteristic is a desirable property in many artificial systems.

This paper proposed modifications based on [4] to the Ant-Miner algorithm proposed on [3]. Such changes show an enhancement in the quality of the discovered classification rules.

3 The Ant-Miner Algorithm

The Ant-Miner algorithm proposed in [3] and [5] aims at discovering classification rules. The rules have the form: IF <term1 AND term2 AND … term N> THEN <class>.

A high level description of Ant-Miner is show below:

```
TrainingSet={all training cases};
DiscoveredRuleList = [ ];   /* rule list is initialized
with an empty list */

  WHILE (TrainingSet > Max_uncovered_cases) DO
    t = 1;   /* ant index */
    j = 1;    /*convergence test index */
    Initialize all trails with the same amount of pheromone
    REPEAT
```

```
        Ant t starts with an empty rule and incrementally
        constructs a classification rule R_t by adding one
        term at a time to the current rule;
        Prune rule R_t;
        Update the pheromone of all trails by increasing
        pheromone in the trail followed by ant t (propor-
        tional to the quality of R_t) and decreasing phero-
        mone in the other trails (simulating pheromone
        evaporation)
        IF R_t is equal to R_t -1 /*update convergence test*/
            THEN j = j + 1;
            ELSE j = 1;
        END-IF
        t = t + 1;
    UNTIL (i ≥ No_of_ants) OR (j ≥ No_rules_converg)
    Chose the Best rule R_best among all rules R_t constructed
    by all the ants;
    Add  rule R_best to DiscoveredRuleList;
    TrainingSet = TrainingSet – {set of cases correctly
        covered by R_best };
END-WHILE
```

[High-Level description of the Ant-Miner algorithm [3]]

The algorithm follows a sequence of steps in order to obtain an ordered list of classification rules that covers almost all training cases. In the beginning, the list is empty and the training set corresponds to all the training cases. In the **WHILE <…> DO** structure, all tracks are initialized with the same amount of pheromone. The executions of the **REPEAT … UNTIL** structure find a classification rule and during the iterations of the **WHILE <…> DO** structure each rule discovered is added to the list. Thereafter covered cases are removed from the training set. This process is performed iteratively until it reaches the limit *Max_uncovered_cases* (cases not covered in the training).

3.1 Pheromone Initialization

Every trail initially receives the same amount of pheromone in accordance with the following formula:

$$\tau_{ij}(t=0) = \frac{1}{\sum_{i=1}^{a} b_i} \tag{1}$$

Where a is the total number of attributes and b_i is the number of attribute values in the domain i.

3.2 Rule Construction

Consider the *term ij* of the form $A_i = V_{ij}$, where A_i is the *ith*-attribute and V_{ij} is the *jth* value of the domain of A_i. The probability of the *term ij* being chosen to be added to the partial rule is given by equation 2.

$$P_{ij}(t) = \frac{\tau_{ij}(t).\eta_{ij}}{\sum_{i}^{a}\sum_{j}^{b_i}\tau_{ij}(t).\eta_{ij}, \forall i \in I} \tag{2}$$

where:

- η_{ij} is an heuristic function value dependent on the problem for the term *ij*;
- $\tau_{ij}(t)$ is the amount of pheromone associated with the term *ij* at a time *t*, corresponding to the amount of pheromone currently available at the position *ij* in the track being followed by the ant;
- a is the total number of attributes;
- b_i is the total number of values in the *i-th* field attributes;
- I is the set of attributes that are not used by the ants.

3.3 Heuristic Function

In Ant Colony Algorithms, a heuristic function is usually used together with the amount of pheromone information to decide which term should be added to the current rule. For the Ant-Miner, the heuristic function is based on Information Theory [9]. The function is given by the following equation:

$$\eta_{ij} = \frac{\log_2 k - H(W \mid A_i = V_{ij})}{\sum_{m=1}^{a} x_m X\left(\sum_{n=1}^{b_m}\log_2 k - H(W \mid A_m = V_{mn})\right)} \tag{3}$$

where:

- a is the total number of attributes;
- X_m receives value *1* if the attribute A_i has not been used by this ant, the value *0* otherwise;
- b_m is the number of values in the do *m-th* attribute.

3.4 Pruning Rule

The main purpose of pruning a rule is to remove irrelevant terms that may have been included undesirably. A great advantage of pruning is to increase the predictive power

of the rules. Another advantage is the simplicity it brings by making the rule easier to be understood.

The quality of the rule is calculated using the following equation:

$$Q = \left(\frac{TruePos}{TruePos + FalseNeg} \right) X \left(\frac{TrueNeg}{FalsePos + TrueNeg} \right) \tag{4}$$

where:

- TruePos is the number of cases covered by the rule having the same class as the predicted by the rule;
- FalsePos is the number of cases covered by the rule having a different class from that predicted by the rule;
- FalseNeg is the number of cases that are not covered by rule having the class predicted by the rule;
- TrueNeg is the number of cases that are not covered by the rule having a different class than the class predicted by the rule.

3.5 Pheromone Update

After each ant completes the construction of its rule, pheromone updating is carried out as follows:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) xQ, \forall ij \in R \tag{5}$$

where:

- R is the set of terms occurring in the rule constructed by the ant at time t;
- $0 \leq Q \leq 1$. The higher the Q value, the better the quality of the rule.

4 The Proposed Algorithm

In this section we will present a new proposal for the Ant-Miner algorithm, which aims to improve its performance.

The proposal is based on [4]. In this case, the way to choose the trail in which ants must pass is more consistent and therefore generates better results.

Initially, the following verbal description should to be considered: “if a mass forager arrives at a fork in a chemical recruitment trail, the probability that it takes the left branch is all the greater as there is more trail pheromone on it than on the right one”. So the probability of choosing the path is [4]:

$$P(D) = \frac{(1 + \tanh(qD))}{2} \tag{6}$$

where:

- D represents the difference of concentration of pheromone in the left branch and the concentration of pheromone in the right branch, denoted by $D = L - R$ and q is the parameter that determines the slope of curve;
- In [4] the suggested value for $q = 0.016$.

This proposal presents two possible options for choosing the path to be followed by the ant:

— In the first one, the probability that the ant choose a term ij depends only on the difference of pheromone concentration between the term ij and the other values of the domain of the attribute i. We will refer to this option as "local strategy". The probability is given by:

$$P_{ij} = \frac{1 + \tanh\left(q\left(2\tau_{ij}(t) - \sum_{j=1}^{b_i} \tau_{ij}(t)\right)\right)}{b_i} \tag{7}$$

— In our second option, to avoid obtaining only optimal local solutions, the probability of choice of the term ij will be based on the difference of pheromone concentration between the term ij and all other values in the domain of all attributes that have not been used bye the ant. This will be referred to as "global strategy". Its probability is given by:

$$P_{ij} = \frac{1 + \tanh\left(q\left(2\tau_{ij}(t) - \sum_{i=1}^{a} x_i * \sum_{j=1}^{b_i} \tau_{ij}(t)\right)\right)}{\sum_{i=1}^{a} x_i * b_i} \tag{8}$$

Note that, rather than one, we have two formulas for rules construction.

4.1 Heuristic Function

In this new approach, the probability formulas will also consider the heuristic information.

Hence, the difference D will be given by the difference between the product of the pheromone concentration and the heuristic function (η_{ij}), therefore, we present two new probability formulas:

$$P_{ij} = \frac{1 + \tanh\left(q\left(2\tau_{ij}(t) * \eta_{ij} - \sum_{j=1}^{b_i} \left(\tau_{ij}(t) * \eta_{ij}\right)\right)\right)}{b_i} \tag{9}$$

$$P_{ij} = \frac{1 + \tanh\left(q\left(2\tau_{ij}(t) * \eta_{ij} - \sum_{i=1}^{a} x_i * \sum_{j=1}^{b_i} \left(\tau_{ij}(t) * \eta_{ij}\right)\right)\right)}{\sum_{i=1}^{a} x_i * b_i} \tag{10}$$

Note that having four different versions of the probability formula there will also be four versions of the algorithm. See below a general version of the modified Ant-Miner:

```
TrainingSet = {all training cases};
DiscoveredRuleList = [ ]; /*rule list is initialized with
an empty list */
WHILE (TrainingSet > Max_uncovered_cases) DO
  t = 1;   /* ant index */
  j = 1;    /*convergence test index */
  Initialize all trails with the same amount of pheromone;
  REPEAT
    Ant t starts with an empty rule and incrementally
    constructs  a classification rule Rt by adding one term at
    a time to the current rule;
    Prune rule Rt;
    Update the pheromone of all trails by increasing pheromone
    in the trail followed by ant t (proportional to the quality
    of Rt) and decreasing pheromone in the other trails
    (simulating pheromone evaporation)
    IF Rt is equal to Rt -1   /*update convergence test*/
      THEN j = j + 1;
      ELSE j = 1;
    END-IF
    t = t + 1;
UNTIL (i ≥ No_of_ants) OR (j ≥ No_rules_converg)
Chose the Best rule Rbest among all rules Rt constructed
by all the ants;
FOR EACH attribute i in Rbest DO
  IF value j doesn't show in previous rules THEN
    attraction_factor_i = attraction_factor_i + 1 /
    numAttributeValues_i;
  END-IF
END-FOR
Add rule Rbest to DiscoveredRuleList;
TrainingSet = TrainingSet – {set of cases correctly
covered by Rbest};
END-WHILE
```

[Ant-Miner algorithm with modifications. A new algorithm]

The loop *for … do* included before the *end … while* stores information from previous iterations that influences the probability formulas. This is not done in the original Ant-Miner algorithm.

5 Computational Results and Discussion

In these experiments we used seven public domain datasets repositories from the University of California [10]. They are Ljubjana Breast Cancer, Wisconsin Breast Cancer, Tic-tac-toe, Dermatology, Hepatitis, Cleveland Heart Disease and Diabetes.

As stated previously, there are four versions of the algorithm, which according to the probability formulas will be referred respectively as V1, V2, V3 and V4.

The results were collected after running each version of the program 10 times, using 10 folds cross-validation, for each dataset. Hence, the average of the accuracy achieved was calculated considering the results obtained in each of the 100 executions for each dataset. The averages are shown in the table below:

Table 2. Comparison between Ant-Miner and the versions V1, V2, V3 and V4

Data Set	Ant-Miner	V1	V2	V3	V4
Ljubljana	73,60% +/-2,57%	75,04% +/-1,86%	75,10% +/-2,80%	76,63% +/-3,21%	74,04% +/- 2,56%
Wisconsin	91,85% +/-1,00%	92,92% +/-0,95%	92,46% +/-0,93%	91,90% +/-0,92%	92,40% +/- 0,95%
Tic-tac-toe	72,11% +/-1,86%	72,03% +/-1,10%	70,21% +/-1,70%	73,41% +/-1,31%	70,41% +/- 1,38%
Dermatology	95,28% +/-1,18%	95,79% +/-0,95%	95,53% +/-1,03%	95,84% +/-1,34%	95,45% +/- 1,09%
Hepatitis	83,67% +/-2,91%	80,01% +/-3,25%	77,78% +/-3,10%	78,41% +/-4,02%	78,65% +/- 3,75%
Cleveland	77,99%+/-2,38%	77,41% +/-1,58%	78,25% +/-2,56%	78,09% +/-2,32%	78,16% +/- 2,23%
Diabetes	67,16%+/-1,72%	66,48% +/-1,58%	67,23% +/-1,82%	68,69% +/-1,65%	67,74% +/- 1,82%

In all experiments, the parameters were set to *Min_cases_per_rule* = 5 and *Max_uncovere_cases* = 10.

The enhanced performance of the new algorithm could be observed especially in the version 3, in which better results were obtained for 6 of the datasets tested. This version associates the local strategy with the heuristic function. These combined elements have proved very effective in different system configurations.

Future studies may be directed to determine the optimal configuration of the initial parameters for each data set in particular, which maximize the predictive accuracy of the classification rules discovered.

References

1. Dorigo, M., Di Caro, G.: The ant colony optimization meta-heuristic. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 11–32. McGraw-Hill, London (1999)
2. Dorigo, M., Di Caro, G., Gambardella, L.M.: Ant algorithms for discrete optimization. Artificial Life 5(2), 137–172 (1999)
3. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: Data Mining with an Ant Colony Optimization Algorithm. IEEE Transactions on Evolutionary Computing 6(4) (2002)
4. Rozin, V., Margaliot, M.: The Fuzzy Ant. IEEE Computational Intelligence Magazine 2(4) (2007)

5. Parpinelli, R.S.: Um Algoritmo Baseado em Colônias de Formigas para Classificação em Data Mining. Dissertação de Mestrado, UTFPR, Curitiba (2001) (in Portuguese)
6. Chen, M.S., Han, J., Yu, P.S.: Data mining: an overview from database perspective. In: Proceedings of the IEEE Transactions on Knowledge and Data Engineering, vol. 8, pp. 866–883 (1996)
7. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
8. Clark, P., Neblett, T.: The CN2 induction algorithm. Machine Learning 3, 261–283 (1989)
9. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, New York (1991)
10. Frank, A., Asuncion, A.: UCI Machine Learning Repository University of California, School of Information and Computer Science, Irvine, CA (2010), http://archive.ics.uci.edu/ml

Improving Differential Evolution Accuracy for Flexible Ligand Docking Using a Multi-solution Strategy

Camila S. de Magalhães*, Carlos Henrique dos S. Barbosa, Diogo M. Almeida, and Laurent E. Dardenne

Universidade Federal Rural do Rio de Janeiro - UFRRJ, Departamento de Matemática BR 465 km 47, Seropédica, Rio de Janeiro, CEP 23890-000, Brazil
Laboratório Nacional de Computacão Científica - LNCC, Departamento de Mecânica Av. Getúlio Vargas 333, Petrópolis, Rio de Janeiro, CEP 25651-075, Brazil

Abstract. Docking of small ligand molecules in protein active sites is a very important and challenging problem in the structure-based drug design field. In this work we propose a Differential Evolution algorithm in conjunction with a multi-solution strategy for the flexible ligand docking problem. The proposed algorithm is evaluated on five highly flexible HIV-1 protease ligands, with known three-dimensional structures, having up to 19 conformational degrees of freedom. The docking results and comparison with classic Differential Evolution algorithm indicate that the incorporation of a multi-solution strategy in Differential Evolution algorithms is very promising and can significantly improve molecular docking accuracy.

Keywords: Flexible docking, HIV-1 Protease ligands, Differential Evolution, Multi-solutions.

1 Introduction

Computational docking methods have gained increasing importance and have been widely used in drug discovery and design projects [1]. Despite the great research efforts in the docking area over the last two decades, significant challenges still remain. A critical issue in molecular docking is to handle ligand and protein flexibility, taking into account hundreds of thousands of conformational degrees of freedom. A number of methodologies have been proposed to include partial or full protein flexibility in docking algorithms [2, 3]. However, this subject remains as an unsolved question, and most docking methods include only the ligand flexibility, keeping the protein fixed in a predetermined position. In fact, even when the fixed receptor approximation is used, docking large and highly flexible ligands is still a considerable challenge to current docking methods [4–7]. This is due to the high dimensionality of the docking problem which considers ligand translational, rotational and conformational degrees of freedom. So the

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 688–698, 2012.

dimensionality of the ligand flexible docking problem is 7+N (7 translational and rotational degrees of freedom plus N conformational degrees of freedom associated to the number of ligand rotatable bonds). A large number of methods have been presented and evaluated for the flexible ligand docking problem [4–12]. Despite their reasonable success rate in ligand pose prediction, most methods cannot successfully dock highly flexible ligands [13–16]. Some recent works have also addressed the highly flexible ligand question using different search strategies and algorithms, including evolutionary algorithms [17–22], swarm optimization [8] and Monte Carlo methods [14]. Nevertheless, although the hybrid swarm optimization-local search algorithm (SODOCK) was successful in redocking experiments, a much lower performance was found with the other four commonly used docking methods evaluated (DOCK 4.0, FlexX 1.8 and the Lamarckian Genetic Algorithm -LGA of Autodock 3.05). The Monte Carlo method used by the RosettaLigand [14] and the LGA of Autodock 4 [23] were also unable to successfully dock ligands with more than 10 rotatable bonds in most cases. In a previous work, a real-coded steady-state genetic algorithm (SSGA) was used in conjunction with a multi-solution technique, named MRTS (Modified Restrict Tournament Selection), for docking of highly flexible ligands [24, 25]. Very promissing results were found with the SSGA-MRTS algorithm showing that the use of multi-solution methods can be a good strategy for highly flexible ligand docking. Recently, this strategy was implemented in a molecular docking program named Dockthor[25], which uses a classical molecular force field specially parametrized for ligand molecules of interest in medicinal chemistry and pharmacology.

On the other hand, the power of Differential Evolution (DE) algorithms in solving complex optimization problems has been extensively shown in the literature, including its application to the molecular docking problem [26].

In this work we evaluate the performance of the Differential Evolution algorithm in conjunction with the MRTS multi-solution technique for the flexible ligand docking problem. The DE-MRTS algorithm was tested on a set of five HIV-1 protease ligands with up to 19 rotatable bonds. In addition, the DE-MRTS algorithm was also compared with the SSGA-MRTS and the classic DE algorithms. We found that the proposed DE-MRTS outperforms the SSGA-MRTS and the classic DE algorithm in terms of accuracy and success rate.

2 Differential Evolution

The Differential Evolution algorithm was proposed by Storn and Price in 1995. Since then, an increasing number of studies regarding successful DE applications in several optimization problems has been found in the literature [27]. The classic DE algorithm refers to the DE/rand/1/bin variation, as described in [27]. The classic DE algorithm starts with a random initial population. For a determined number of generations GEN, each individual i of the population at generation g (target vector x_g^i) creates another individual (trial vector u_g^i) using crossover and mutation operations. The mutation operation (eq. 1) consists in adding a randomly selected individual to the difference between two other randomly chosen

individuals multiplied by a factor F. The crossover operation (eq. 2) determines which coordinates of the target vector will suffer the mutation operation in order to create the trial vector.

$$v_g^{j,i} = x_g^{j,r_3} + F(x_g^{j,r_1} - x_g^{j,r_2}). \tag{1}$$

$$u_g^{j,i} = \begin{cases} v_g^{j,i} \text{ if } rand[0,1] <= CR \text{ or } j = j_{rand}, \\ x_g^{j,i} \text{ otherwise.} \end{cases} \tag{2}$$

The F parameter in eq. 1 is the mutation factor specified by the user, ranging from (0,1). The individuals r_1, r_2 and r_3 must be different among themselves, besides different from individual i. The CR parameter in eq. 2 is chosen from [0,1) and controls the DE crossover. rand[0,1] is a randomly chosen number from a uniform distribution in the range [0,1]. A randomly selected vector coordinate j_{rand} is also used to prevent the trial vector u_g^i to be equal to the target vector $x_g^{j,i}$. At the end of each generation g the trial vector u_g^i is kept to the next generation $g+1$ if its fitness is equal or better than the fitness of the target vector x_g^i. In this case, the trial replaces the target vector. Otherwise, the target vector is retained in the next generation. In this work a steady-state version of the classic DE algorithm is adopted [28]. In a steady-state DE the trial vector generated at generation g is immediately tested for replacing the target vector in the DE population at the current generation.

3 Differential Evolution for Molecular Docking

In this work two versions of the DE steady-state algorithm were implemented, the classic DE and the DE-MRTS algorithm (DE in conjunction with the MRTS method). The two versions were implemented in C++ programming language and incorporated in the source code of the docking program Dockthor[25].

3.1 Solution Representation

A solution for the molecular docking problem is a three-dimensional structure of the ligand molecule, defined by the translational, rotational and conformational position of the ligand with respect to the protein active site. Each possible solution (individual in the population) is represented in the docking algorithms by a chromosome with three kinds of genes: translational, rotational, and conformational. The translational genes are the X, Y, Z coordinates of the reference atom which determines the ligand spatial position. The reference atom is usually chosen as the closest atom to the center of mass of the ligand. The rotational genes determine the ligand rigid body rotation, fixing the reference atom. They are represented by a quaternion, which is constituted by a rotation angle and a unit vector, defining the rotation axis. The conformational genes are the ligand dihedral angles associated to the rotatable chemical bonds (one gene by torsion).

For each chromosome - a real vector in R^n - a unique ligand spatial structure (coordinates for all atoms) can be computed. The distance between two individuals (ligand structures) is calculated by the RMSD, which is defined as the root-mean-square-deviation between all atomic coordinates (excluding hydrogen atoms) of those structures.

3.2 Fitness Function

The fitness function is the ligand-protein interaction energy plus the ligand intramolecular energy, evaluated using the non-bonded terms and the terms associated to the ligand dihedral torsions of the MMFF94 classical molecular force field [29]. To reduce the computational cost associated to the evaluation of protein-ligand interactions, we implemented a grid-based methodology where the protein active site is embedded in a 3D rectangular grid centered in the protein active site. At each grid point the electrostatic interaction energy and the Lennard-Jones (LJ) potential terms, for each ligand atom type, are computed and stored taking into account all protein atoms. The protein contribution at a given point is obtained during the algorithm run by tri-linear interpolation in each 3D grid cell. The grid dimension used is 23 Å in each direction with a spacing of 0.25 Å.

3.3 The Proposed DE-MRTS Algorithm

The DE-MRTS algorithm consists of the classical steady-state DE algorithm in which the trial vector generated is tested for insertion into the DE population according to the MRTS scheme [24].

The Multi-solution MRTS Method. A great number of multi-solution techniques has been incorporated in DE algorithms for multi-modal function optimization problems[30, 31]. However, in this work, we are particularly interested in the performance of the DE algorithm with the MRTS multi-solution scheme [24]. As the MRTS method was specifically designed to deal with the high modality of the fitness landscape of flexible ligand docking problems, we believe that this strategy can be a good start option. The MRTS(Modified Restrict Tounament Selection) method is based on the RTS (Restricted Tournament Selection) method proposed by Rarik [32]. The MRTS uses the concept of insertion of new offspring in the population replacing similar ones to promote useful diversity and increase the algorithm search capability. In the MRTS scheme two tournaments are made. Tn the first (resp. second) tournament w1 (resp. w2) individuals that are better than the trial vector are drafted. The winner of the first tournament, *CBetter*, is the closest individual to the trial vector, among w1 individuals drafted in the first tournament. The winner of the second tournament, *CWorse* is the closest individual to the trial vector, among the w2 individuals drafted in the second tournament.The trial vector is then inserted in the population in the following way:

– If the trial vector is closer to *CWorse* than *CBetter*, then *CWorse* is replaced by the newly generated individual (trial vector)
– Else, if the RMSD between the trial and *CBetter* is greater than 2.0 Å, then *CWorse* is replaced by the trial vector. Otherwise, the trial vector is discarded.

The distance criterion RMSD $<$ 2. Å is used to avoid an individual insertion when a very similar and better individual already exists in a particular region of the search space. We set $w1 = w2 = 0.6$, it means that the tournament size w1 is initially equal to 60% of the individuals that are better than the trial, and the tournament size w2 is equal to 60% of the individuals that are worse than it. The tournament sizes w1 and w2 linearly decreases with the counter of fitness function evaluations, until the minimum value 0.1.

4 Test Set Preparation

The experimental structures of the five HIV-1 protease-ligand complexes used in this work were obtained from the Protein Data Bank (PDB). The number of dihedral angles/torsions, total number of degrees of freedom (dimension) and the PDB file code for each ligand molecule of the test set are shown in Table 1. The structural formula of the ligands is shown in Fig. 1. All water molecules were removed, and the PDB files were separated in two different files, one containing the protein and the other one with the ligand structure. The protein and ligand atom charges and the other MMFF94 force field parameters were generated by the mmffligand and pdbthorbox tools from the Dockthor program [25].

Table 1. HIV-1 protease ligands complexes tested

Ligand	Number of Dihedral Angles	Total Number of Degrees of Freedom	PDB id
Nelfinavir	12	19	1OHR
DMP323	14	21	1BVE
Indinavir	14	21	1HSG
Saquinavir	15	22	1HXB
Ritonavir	19	26	1HXW

5 Results

The DE-MRTS algorithm was evaluated on a test set of five HIV-1 protease complexes. The HIV-1 protease enzyme is a very well understood and important molecular target in the development of drugs against the AIDS virus. Additionally, the enclosed active site of the of HIV-1 protease enzyme, together with the large conformational flexibility of the ligands considered, makes this test suite a good challenge to docking algorithms [24, 33]. To evaluate the performance of the

DE algorithm in conjunction with the MRTS method, we compared it with the classic DE/rand/1/bin steady-state algorithm and also with the SSGA-MRTS algorithm implemented in the Dockthor program. The SSGA-MRTS uses five genetic operators and an adaptation method to set operator probabilities. A detailed description of the SSGA-MRTS algorithm can be found in [24, 25, 33]. All the results were averaged in 30 independent docking runs. For each run, for each one of the three algorithms, the initial populations were generated using a uniform random number distribution with the same random seeds. For all algorithms the population size (NP) has been fixed to 100 individuals, whereas the maximum number of fitness function evaluation (max$NFEs$) has been set to $5.0\text{x}10^5$. The DE control parameters used in the two algorithms were $F = 0.5$ and $CR = 0.5$.

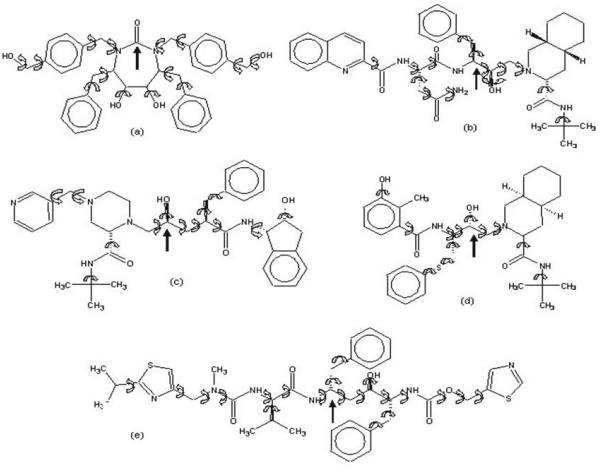


Fig. 1. Structural formula of the HIV-1 protease ligands: (a) DMP323, (b) Saquinavir, (c) Indinavir, (d) Nelfinavir and (e) Ritonavir. The dihedral angles and the reference atoms are shown by curved arrows and right arrows respectively.

Docking success was measured in terms of the RMSD between a ligand pose found by the algorithm and the experimental structure (obtained from the PDB file). In this work, a ligand conformation with a RMSD value less than 2.5 Å from the experimental structure was considered a success. The success rate (SR) was obtained considering the percentage of the lowest energy structures with RMSD value less than 2.5 Å in 30 independent docking runs for each case. Besides the success rate, in order to compare the performance of the two implemented algorithms, we also adopt the following measures: accuracy and robustness. Accuracy is related to the RMSD values of the lowest energy solutions. Smaller the obtained RMSD value, more accurate is the algorithm. The second measurement is related to the averaged energy values. The algorithm is considered robust if the averaged best energy values are close to the lowest energy value found. It means

Table 2. Comparison of docking results for DE classic, DE-MRTS and SSGA-MRTS algorithms

Ligand	Algorithm	Best Energy (BE)[a]	RMSD BE[b]	Average Energy[a]	Average RMSD[b]	Success Ratio (%)[c]
Nelfinavir	DE classic	125.67	1.81	190.47 (38.99)	2.36 (0.61)	83.
	DE-MRTS	60.10	1.76	61.24 (5.36)	1.81 (0.07)	100.
	SSGA-MRTS	60.28	1.80	85.59 (21.72)	3.43 (2.36)	57.
DMP323	DE classic	71.99	1.59	72.02 (0.03)	1.59 (0.02)	100.
	DE-MRTS	72.01	1.60	72.05 (0.05)	1.59 (0.0)	100.
	SSGA-MRTS	72.14	1.60	74.08 (7.43)	3.017 (2.9)	80.
Indinavir	DE classic	163.72	1.45	331.33(81.20)	3.46 (1.86)	30.
	DE-MRTS	70.28	0.67	90.92 (15.41)	1.88 (1.45)	73.
	SSGA-MRTS	71.40	1.07	97.25 (14.32)	4.56 (3.76)	43.
Saquinavir	DE classic	221.18	2.00	326.82 (47.41)	2.48 (0.87)	63.
	DE-MRTS	78.78	0.59	81.40 (8.39)	0.71 (0.32)	100.
	SSGA-MRTS	79.28	0.93	106.22 (15.75)	4.32 (3.55)	47.
Ritonavir	DE classic	323.55	6.10	588.24 (127.31)	6.29 (2.14)	0.0
	DE-MRTS	74.74	5.08	207.55 (117.47)	5.58 (1.56)	3.
	SSGA-MRTS	28.96	1.20	56.16 (12.86)	6.79 (3.43)	10.0

[a]Energies in Kcal/mol. [b] RMSD in Å.[c]Success rate for lowest energy solutions with RMSD < 2.5 Å in 30 docking docking runs. Standard deviations are given between parentheses.

that the algorithm can find solutions close to the best one in each independent docking run.

Docking results for the classic DE, DE-MRTS and SSGA-MRTS algorithms are shown in Table 2. The classic DE algorithm exhibits reasonable performance for the ligands of this test set. The best success rates (SR) are obtained for the ligands DMP323 and Nelfinavir. This two ligands are the easiest docking cases of this set. Nelfinavir has the lowest number of dihedral angles and DMP323 exhibits symmetry and a lower dependence among its conformational degrees of freedom (see Fig. 1). For the other three more challenging ligands, the classic DE algorithm exhibits a poorer performance, with SR of 0.% for Ritonavir ligand and of 30.% and 63.% for Indinavir and Saquinavir, respectively. A similar performance is observed in relation to the average energies in comparison to the lowest energies. The average energy values are closer to the lowest energy values only for the ligands DMP323 and Nelfinavir. Interestingly, with the exception of Ritonavir and indinavir, for all ligands tested the RMSD values (of the lowest energy and the average) obtained by the DE classic algorithm are below 2.5 Å. The DE-MRTS algorithm exhibits very good performance for the ligands of the test set, excepting the Ritonavir ligand. A SR of 100.% was obtained for the ligands DMP323, Nelfinavir and Saquinavir, and a SR of 73.% for the Indinavir. For Ritonavir, the largest and more flexible ligand, the DE-MRTS algorithm obtained

3.% of SR. For all ligands the average energy values are closer to the lowest energy values, excepting Ritonavir and Indinavir ligands. The RMSD of the lowest energy and average RMSD obtained by the DE-MRTS algorithm are below 2.0 Å for the ligands DMP323, Nelfinavir, Saquinavir and Indinavir. If we compare the two DE based algorithms, DE-MRTS obtained a SR higher than classic DE for three of the five ligands tested (Nelfinavir,Indinavir and Saquinavir). The energy and RMSD values obtained by DE-MRTS for these ligands are also much lower than the values obtained by the classic DE algorithm. No difference in the performance of classic DE and DE-MRTS was observed for the ligand DMP323, the easiest docking case tested. However, for the Ritonavir Ligand, the most complex one, despite the poor SR obtained by classic DE and DE-MRTS, the energy and RMSD values obtained with DE-MRTS are much lower than the ones obtained with the classic DE algorithm.

Comparing the DE classic and DE-MRTS algorithms with SSGA-MRTS we observe that the SSGA-MRTS performance is worse than both algorithms for Nelfinavir and DMP ligands. However, for Indinavir and Saquinavir, the SSGA-MRTS algorithm outperforms the classic DE algorithm in terms of energy values and also SR in the case of the Indinavir ligand. The best overall performance was found with the DE-MRTS algorithm. Interestingly, for Ritonavir ligand the best energy and RMSD values were obtained with the SSGA-MRTS method, with a SR of 10.%.

5.1 Effect of Varying Parameters

The effect of varying the population size and the maximum number of fitness function evaluations in the DE-MRTS and DE classic algorithms was verified for the Indinavir ligand. All the other parameters were fixed as in table 2 and one of the specific parameters, NP or max*NFEs*, was varied. The results are shown in Table 3.

When the population size and the maximum number of fitness function evaluations are varied we also observe a much better performance of the DE-MRTS in comparison with the DE classic algorithm. Increasing the population size to 250 and 500 we observe a decreasing performance of the DE-MRTS algorithm. The success rate decreases to 70.% and 50. %, respectively for NP =250 and NP =500. The energy and RMSD values (lowest and average) are also worse than the ones obtained using NP=100. For DE classic algorithm a slightly better SR is observed with NP=250. However, energies and RMSD values are worse than when NP=100 is used. When NP=500 the results are also worse than when using NP=100 and NP=250.

The same behaviour as described above (none improvement increasing NP) was also observed for all ligands of this test set (results not shown). However, when we analyse the final DE population of the two algorithms, we note a high number of distinct solutions that increases with increasing population sizes. So, we believe that this behaviour is due to a poorer convergence of the algorithms when the population size is increased, once that the max*NFEs* parameter is fixed in $5.0\text{x}10^5$.

Table 3. Effect of varying parameters for Indinavir

Varying parameter	Value	Best Energy (BE)[a]	RMSD of the BE[b]	Average Energy[a]	Average RMSD[b]	Success Ratio (%)[c]
DE MRTS						
NP[d]	100	71.28	0.67	90.92 (15.41)	1.88 (1.45)	73.
	250	126.68	0.97	188.55 (37.43)	2.553 (1.29)	70.
	500	146.99	1.21	332.35 (75.14)	3.09 (1.26)	50.
max*NFEs*[e]	2.5x10^5	80.62	0.93	137.27 (49.11)	2.68 (1.59)	30.
	5.0x10^5	70.28	0.67	90.92 (15.41)	1.88 (1.45)	73.
	1.0x10^6	67.68	0.70	78.46 (12.59)	1.41 (1.35)	83.
DE Classic						
NP[d]	100	163.72	1.45	331.33 (81.20)	3.46 (1.86)	33.
	250	292.95	1.80	421.65 (67.57)	3.70 (2.09)	40.
	500	317.21	4.91	507.38 (95.99)	3.49 (1.84)	30.
max*NFEs*[e]	2.5x10^5	259.40	2.35	453.81 (94.50)	3.69 (1.68)	30.
	5.0x10^5	163.72	1.45	331.33 (81.20)	3.46 (1.86)	33.
	1.0x10^6	173.10	4.02	244.70 (36.57)	2.79 (1.10)	60.

[a]Energies in Kcal/mol. [b] RMSD in Å.[c]Success rate for lowest energy solutions with RMSD $<$ 2.5 Å in 30 docking docking runs. Standard deviations are given between parentheses. [d]max*NFEs* set to 5.0x10^5. [e]NP set to 100.

If the max*NFEs* parameter is set to 2.5x10^5 (keeping NP=100), we observe a decrease in the sucess rate, and a higher energy and RMSD values, in comparison with the reference value max*NFEs*=5.0x10^5, for both DE classic and DE-MRTS algorithms. However, when the max*NFEs* parameter is increased to 1.0x10^6, higher success rates are obtained with DE classic and DE-MRTS. Lower averaged energies and RMSD values are also obtained with both algorithms.

6 Conclusion

The performance of two DE based algorithms (DE-MRTS and classic DE) were evaluated for the flexible ligand docking problem. We concluded that the use of the MRTS method can improve significantly the DE performance in flexible docking problems. In addition, comparison of the DE-MRTS with the SSGA-MRTS algorithm, with the parameters used in this work, shows that DE-MRTS outperforms SSGA-MRTS for all ligands tested, with exception of Ritonavir. Moreover, the obtained results indicate that the DE-MRTS method is very valuable in order to develop a faster, more robust and accurate docking methodology to be definitely incorporated in the next versions of the Dockthor program. In this sense, more extensive studies using a more diverse set of ligand-protein complexes and different algorithm parameters (e.g., population size, *CR*, F) are needed.

Acknowledgments. The authors would like to thank to CNPq and FAPERJ for financial support. Project FAPERJ number:E-26/102.443/2009.

References

1. Kitchen, D.B., Decornez, H., Furr, J.R., Bajorath, J.: Docking and Scoring in Virtual Screening for Drug Discovery: Methods and Applications. Nat. Rev. Drug Discov. 3(suppl. 11), 935–949 (2004)
2. Meagher, K.L., Carlson, H.A.: Incorporating Protein Flexibility in Structure-based Drug Discovery: using HIV-1 Protease as a Test Case. J. Am. Chem. Soc. 126, 13276–13281 (2004)
3. Zhao, Y., Sanner, M.F.: Protein-ligand Docking with Multiple Flexible Side Chains. J. Comput. Aided Mol. Des. 22, 673–679 (2008)
4. Cozzini, P., Kellogg, G.E., Spyrakis, F., Abraham, D.J., Costantino, G., Emerson, A., Fanelli, F., Gohlke, H., Kuhn, L.A., Morris, G.M., Orozco, M., Pertinhez, T.A., Rizzi, M., Sotriffer, C.A.: Target Flexibility: An Emerging Consideration in Drug Discovery and Design. J. Med. Chem. 51(suppl. 20), 6237–6255 (2008)
5. Bottegoni, G., Kufareva, I., Totrov, M., Abagyan, R.: Four-dimensional Docking: A Fast and Accurate Account of Discrete Receptor Flexibility in Ligand Docking. J. Med. Chem. 52, 397–406 (2009)
6. B-Rao, C., Subramanian, J., Sharma, D.: Managing Protein Flexibility in Docking and its Applications. Drug Discov. Today 14(suppl. 7/8), 394–400 (2009)
7. Mukherjee, S., Balius, T.E., Rizzo, R.C.: Docking Validation Resources: Protein Family and Ligand Flexibility Experiments. J. Chem. Inf. Model. 50(11), 1986–2000 (2010)
8. Chen, H., Liu, B., Huang, H., Hwang, S., Ho, S.: SODOCK: Swarm Optimization for Highly Flexible Protein-ligand Docking. J. Comput. Chem. 28, 612–623 (2007)
9. Friesner, R.A., Banks, J.L., Murphy, R.B., Halgren, T.A., Klicic, J.J., Mainz, D.T., Repasky, M.P., Knoll, E.H., Shelley, M., Perry, J.K., Shaw, D.E., Francis, P., Shenkin, P.S.: Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. J. Med. Chem. 47, 1739–1749 (2004)
10. Kang, L., Li, H., Jiang, H., Wang, X.: An Improved Adaptive Genetic Algorithm for Protein-ligand Docking. J. Comput. Aided Mol. Des. 23, 1–12 (2009)
11. Li, H., Zhang, H., Zheng, M., Luo, J., Kang, L., Liu, X., Wang, X., Jiang, H.: An Effective Docking Strategy for Virtual Screening Based on Multi-objective Optimization Algorithm. BMC Bioinformatics 10, 58 (2009)
12. Grosdidier, A., Zoete, V., Michielin, O.: Blind Docking of 260 Protein-ligand Complexes with EADock 2.0. J. Comput. Chem. 30, 2021–2030 (2009)
13. Abayan, R., Totrov, M., Kuznetzov, D.: ICM A New Method for Protein Modeling and Design: Applications to Docking and Structure Prediction from the Distorted Native Conformation. J. Comput. Chem. 15, 488–506 (1994)
14. Davis, I.W., Baker, D.: ROSETTA Ligand Docking with Full Ligand and Receptor Flexibility. J. Mol. Biol. 385, 381–392 (2009)
15. Warren, G.L., Andrews, W., Capeli, A., Clarke, B., LaLonde, J., Lambert, M.H., Lindvall, M., Nevins, N., Semus, S.F., Senger, S., Tedesco, G., Wall, I.D., Woolven, J.M., Peishoff, C.E., Head, M.S.: A Critical Assessment of Docking Programs and Scoring Functions. J. Med. Chem. 49, 5912–5931 (2006)
16. Corbeil, C.R., Moitessier, N.: Docking Ligands into Flexible and Solvated Macromolecules. 3. Impact of Input Ligand Conformation, Protein Flexibility, and Water Molecules on the Accuracy of Docking Programs. J. Chem. Inf. Model. 49, 997–1009 (2009)

17. Trosset, J.Y., Scheraga, H.A.: PRODOCK: Software Package for Protein Modeling and Docking. J. Comp. Chem. 20(suppl. 4), 412–427 (1999)
18. Kontoyianni, M., McClellan, L.M., Sokol, G.S.: Evaluation of Docking Performance: Comparative Data on Docking Algorithms. J. Med. Chem. 47, 558–565 (2004)
19. Sousa, S.F., Fernandes, P.A., Ramos, M.J.: Protein-ligand Docking: Current Status and Future Challenges. Proteins 65, 15–26 (2006)
20. Erickson, J.A., Jalaie, M., Robertson, D.H., Lewis, R.A., Vieth, M.: Lessons in Molecular Recognition: The Effects of Ligand and Protein Flexibility on Molecular Docking Accuracy. J. Med. Chem. 47, 45–55 (2004)
21. Li, X., Li, Y., Cheng, T., Liu, Z., Wang, R.: Evaluation of the Performance of Four Molecular Docking Programs on a Diverse Set of Protein-Ligand Complexes. J. Comp. Chem. 31, 2109–2125 (2010)
22. Jones, G., Willet, P., Glen, R.C., Leach, A.R., Taylor, R.: Development and Validation of a Genetic Algorithm for Flexible Docking. J. Mol. Biol. 267, 727–748 (1997)
23. Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Hart, W.E., Belew, R.K., Olson, A.J.: Automated Docking using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function. J. Comput. Chem. 19, 1639–1662 (1998)
24. de Magalhães, C.S., Barbosa, H.J.C., Dardenne, L.E.: Selection-Insertion Schemes in Genetic Algorithms for the Flexible Ligand Docking Problem. In: Deb, K., Tari, Z. (eds.) GECCO 2004, Part I. LNCS, vol. 3102, pp. 368–379. Springer, Heidelberg (2004)
25. Almeida, D.M.: Dockthor: Implementao, Aprimoramento e Validao de um Programa de Docking Receptor-Ligante. MSc. Dissertation (2011)
26. Thomsen, R.: Flexible Ligand Docking using Evolutionary Algorithms: Investigating the Effects of Variation Operators and Local Search Hybrids. Biosystems 72(1-2), 57–73 (2003)
27. Storn, R., Price, K.: Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012, ICSI (March 1995), ftp.icsi.berkeley.edu
28. Zaharie, D.: Extensions of Differential Evolution Algorithms for Multimodal Optimization. In: Proceedings of SYNASC 2004, 6th International Symposium of Symbolic and Numeric Algorithms for Scientific Computing, pp. 523–534 (2004)
29. Halgren, T.: Merck Molecular Force Field.1. Basis, Form, Scope, Parametrization, and Performance of MMFF94. J. Comput. Chem. 17, 490–519 (1996)
30. Thomsen, R.: Multimodal Optimization Using Crowding-based Differential Evolution. In: IEEE Congress on Evolutionary Computation, vol. 2, pp. 1382–1389 (2004)
31. Epitropakis, M.G., Tasoulis, D.K., Pavlidis, N.G., Plagianakos, V.P., Vrahatis, M.N.: Enhancing Differential Evolution Utilizing Proximity-Based Mutation Operators. IEEE Transactions on Evolutionary Computation 15(1), 99–119 (2011)
32. Harik, G.R.: Finding Multimodal Solutions Using Restricted Tournament Selection. In: Eshelman, L. (ed.) Proc. of the Sixth Intl. Conf. on Genetic Algorithms, pp. 24–31. Morgan Kaufmann, San Francisco (1995)
33. De Magalhães, C.S., Barbosa, H.J.C., Dardenne, L.E.: A Genetic Algorithm for the Ligand-protein Docking Problem. Genet. Mol. Biol. 27(4) (2004)

Fault Identification in Doubly Fed Induction Generator Using FFT and Neural Networks

Marcelo Patrício de Santana, José Roberto Boffino de Almeida Monteiro, Geyverson Teixeira de Paula, Thales Eugenio Portes de Almeida, Gustavo Bueno Romero, and Júlio César Faracco

University of São Paulo, School of Engineering of Sao Carlos, São Carlos, Brazil
{marceloengenheiroe,jrm,geyverson.paula,thales.eugenio.almeida}@usp.br, {jcfaracco,gusbromero}@gmail.com
http://www.sel.eesc.usp.br

Abstract. This paper presents a fault identification system for doubly fed induction generator (DFIG). It considers cases of single phase short-circuits and load switching. The system uses the fast fourier transform (FFT) to preprocessor data, which consist by the stator line currents. The principal component analysis (PCA) is employed to reduce the dimensionality of the output data of FFT and the fault identification is made by means of artificial neural network (ANN). Also, a post-processing (PP) is employed in order to increase the network reliability, which reduces the error of ANN. The system is simulated in the MATLAB simulation software using a database with different voltage, speed and load. The results show that the system with PCA, FFT and ANN has a good performance and accuracy with the PP to fault identification in the DFIG.

Keywords: Neural network, Fault identification, Fast fourier transform.

1 Introduction

Generation at constant speed has been used with synchronous generator, where the generator output frequency is controlled by the control of primary mover. However, the mechanical time constant is very high if compared to frequency of the grid . This could be a problem during transient and disturbances like grid short circuit, so the frequency can be easily desynchronized.

Modern high power wind turbines have mechanical variable speed, or so called variable speed generation. The advantage of this topology is a flicker reduction if compared to constant speed generation, and a four-quadrant active and reactive power capabilities [1]. The doubly-fed induction generator (DFIG) is widely used in variable speed generation. The main advantage of this kind of machine is that the power of the converter connected to the rotor winding is rated in 25-30% of the total power of the machine [2]. This is a major factor in order to reduce the generation system overall cost if compared to the systems where converters have the same power capabilities of their machines, or total power converters.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 699–706, 2012.

The commonly topology of DFIG voltage and frequency control is by vectorial machine field decomposition, or vector control, for short, as it offers independent control of field and high torque dynamics [3].

During normal lifetime, DFIG is submitted to critical situations. External short-circuits occurs frequently and causes excessive heating and increases machine currents and causes torque ripple. This can damage electrical equipment as well as the generator itself, thus they need to be effectively protected from such faults. Therefore, monitoring the electrical system from faults is far important as promptly protection actions increase machine reliability, increasing its operational life and reduces preventive maintenance frequency.

There are some works that aim fault identification for DFIGs. For example, in [4] FFT is used to identify different kinds of faults. In [5], different kinds of DFIG internal faults are identified, such as short-circuit in the stator coils. In [6] faults are determined using nonlinear models to estimate the currents values of the machine.

The main goal of this work is to determine a system for fault identification with fast response, 5/4 cycle of the frequency of grid. As fault identification is too critical, system identification reliability is very important, so the identification is subdivided in pre-identification process and a confirmation process.

The paper is organized as follow: in the first section, it's shown the orthogonal transformations applied to the DFIG and the vector control in mechanical variable speed operation; in the second section, it's shown a system for identification using a pre-processing with FFT and neural networks; and finally, some results of simulation of fault identification are shown and discussed.

2 Transformation Vector

The DFIG can be represented by a machine with three stator coils out of phase by 120° (electrical), and by three rotor coils out of phase by 120° (electrical). The equations that represent the machine are dependent on the mechanical position of the rotor, which shows field interactions between rotor and stator.

The commonly used orthogonal transformation in machine equation, i.e., $\alpha\beta$ and dq transformations, are of great value once they simplify machine equations in three phase mode, reducing its order, and also gives a reasonable feeling about the interactions between machine fields and currents. This is a fact that facilitates control design of DFIG, as well as electrical machines in a general way. Eventually, the three phase equations are replaced by two sets of equations (for $\alpha\beta$ axes or for dq axes), but now, out of phase by 90° (electrical).

The final axes that are used are the dq axes, so $\alpha\beta$ axes are intermediate, and can be omitted in transformations. The dq axes are not fixed, differently from $\alpha\beta$ which are fixed in stator, they can be placed in any angle reference, in the mechanical position of the rotor, in the rotor field, in the air-gap resulting field, etc. The used reference for dq angle depends on which control topology will be used.

The equations of the machine in axis with the a generic speed ω_λ, in matrix form are shown in the equations 1 and 2 [10].

$$\boldsymbol{v_S} = R_S \boldsymbol{i_S} + \frac{d}{dt}\boldsymbol{\Psi_S} + j\omega_\lambda \boldsymbol{\Psi_S} \tag{1}$$

$$\boldsymbol{v_R} = R_R \boldsymbol{i_R} + \frac{d}{dt}\boldsymbol{\Psi_R} + j\left(\omega_\lambda - P\omega_{mec}\right)\boldsymbol{\Psi_R} \tag{2}$$

Where:

v_S e v_R: stator voltage vector and rotor voltage vector, respectively;
R_S e R_R: vector of resistances in the stator an rotor, respectively;
i_S e i_R: vector of currents in the stator an rotor, respectively;
Ψ_S e Ψ_R: vector of stator and rotor flux, respectively;
P: number of poles of the machine;
ω_{mec}: mechanical rotor speed; and
ω_λ: generic reference speed.

3 Vector Control of DFIG

The vector control can decouple voltage and frequency of the DFIG with mechanical variable speed and has the advantage of no necessity of magnetic flux measurement or estimation, so it prevents errors caused by noise measurements on the stator voltage [11].

Basically, there are two types of vector control. The first is called indirect control, where the voltage is controlled by the stator flux. The second is called direct control, where the output voltage is controlled by your own value in the stator [3]. This paper uses the direct control.

The direct vector control is divided in two parts, frequency control and voltage control.

3.1 Frequency

The frequency of stator voltage depends on the frequency of the resulting stator flux. Thus, if the speed of the axes dq with the values of flux has the synchronous speed, the same happens with the frequency. This control makes possible to keep control of frequency independently of mechanical speed and load.

3.2 Voltage

The direct vector control is performed by means of the output voltage modulus. When stator flux is in the dq axis with the synchronous speed, the equations 1 and 2 are written according to the equations 3 and 4.

$$v_{ds} = R_s i_{ds} + L_s \frac{di_{ds}}{dt} + L_m \frac{di_{dr}}{dt} \tag{3}$$

$$v_{qs} = R_s i_{qs} + \omega_s L_s i_{ds} + \omega_s L_m i_{dr} \tag{4}$$

L_S e L_m: stator inductance and mutual inductance, respectively;

Observing the equations 3 e 4, the stator voltage depends on the stator currents i_{qs}, i_{qr} and i_{dr}. The stator currents are considered disturbances which depend on the load and are independent of the control. Thus, the component i_{dr} is considered a control variable.

The another variable of control, i_{qr}, is responsible by the machine electromagnetic torque which is contrary to the external mechanical torque.

3.3 The System of Identification

The system of single phase short-circuit identification is shown in Fig. 1.

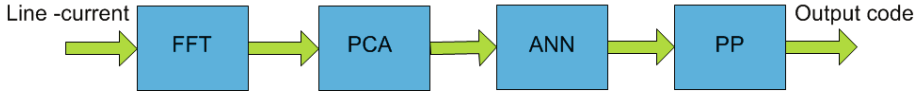


Fig. 1. System Indentificationf Diagram

The inputs are the stator line-currents of each phase. Under normal machine operation, i.e., with no faults, these values are sine functions. Those signals are preprocessed by means of fast fourier fransformer (FFT), which converts the signal in the time domain to the frequency domain.

The amount of harmonic contents is reduced by analysis of frequency by means of principal components analysis (PCA), reducing the dimension of input data and eliminating the harmonics that are not important to the identification.

The identification of fault is made with artificial neural network (ANN). The ANN topology chosen is a perceptron multilayer. The training algorithm is the Levenberg-Marquard, using the Matlab. Each of the phases has two fault identifications. One is in 1/4 of cycle of the frequency and the other is in 5/4 of cycle of the grid frequency.

The post-processing (PP) is a way of increasing network reliability. In load switchings the stator current has abrupt variation and the network can indicates faults in some ocasions. Noises and different sequences of events in database for training can confuse the network and may have conflicting outputs with duplicate indication of fault. In the PP, a fault is identified in 1/4 of cycle. After more one cycle, there is a confirmation if the fault really have occurred. This decreases the error of the network in adverse conditions.

3.4 Simulation and Results

In order to build a database with numerous fault conditions, 275 operational conditions have been simulated. Each of one has different speed, load and voltage values. The training process, validation and test of ANN has about 20.265 data.

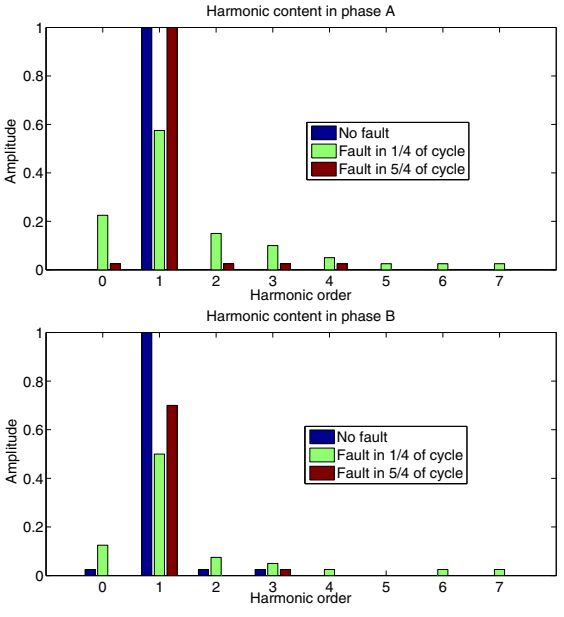


Fig. 2. Harmonic contents

The ANN with the best performance has 9 neurons in the input layer, 10 in the hidden layer and 7 in output layer.

The figure 2 shows the harmonic content for phase A and B in a short-circuit in phase A to neutral. The x-axis has the harmonic order. The PCA reduced the order of the input until the seventh harmonic. The FFT preprocessor normalized the amplitude values by the highest value within the three currents.

As a case study, in order to analyse the proposed identification system, a generator is connected to two loads, A and B. Considering $t = 1$s, a fault occurs in a phase A in the load B. The short-circuit remains for one second and in $t = 2$s the load B is isolated. Thus, in $t = 2$s the generator is connected only to the load A.

Fig. 3 shows graphics using fault identification in 1/4 of cycle in one step and using the method proposed here, in 5/4 of cycle, using two steps. The output is relative to the phase A. As no fault occurs in phase A, the output must be always zero, once a value different of zero indicates a fault.

Fig. 4 shows graphics with a identification in 5/4 of cycle using one step and another using two steps, as proposed in this work. The output is relative to the phase B. As the fault occurs in phase B, the value has to be one, indicating a fault.

Fig. 5 shows the line-current in phase A (above) and the output fault identification for the same phase (bellow). Approximately two cycles after fault in phase A are shown in the graphics, thus it is possible to observe the time response of the proposed identification system.

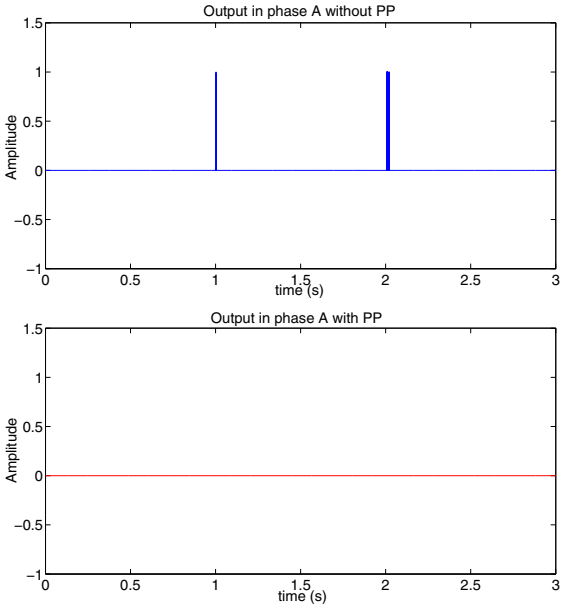


Fig. 3. Comparison of different topologies. Above: fault identification in 1/4 of cycle, using one step; bellow: fault identification in 5/4 of cycle, using two steps.

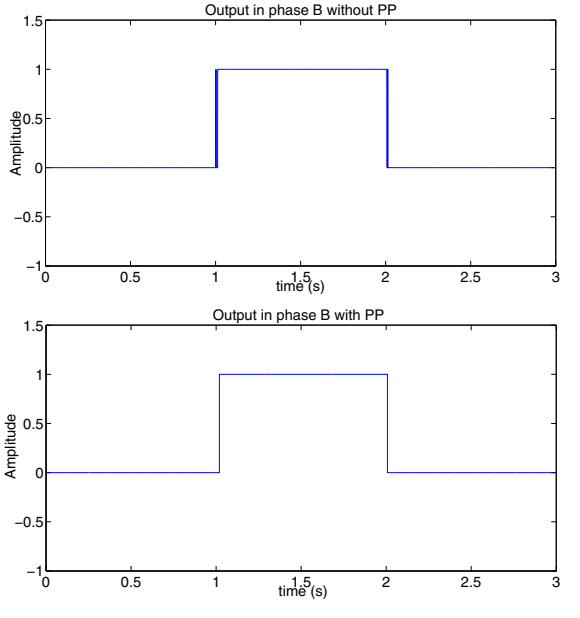


Fig. 4. Comparison of different topologies. Above: fault identification in 5/4 of cycle, using one step; Bellow: fault identification in 5/4 of cycle, using two steps.

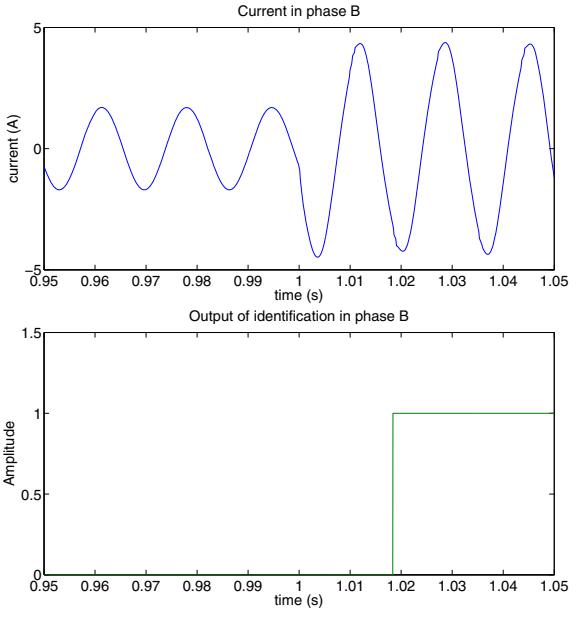


Fig. 5. Analysis of fault in phase A. Above: instantaneous current of phase A; bellow: output fault identification of the proposed method.

4 Conclusion

This paper proposes a method for fault identification in loads fed by doubly-fed induction generators. Basically, the proposed method has two steps: a pre-processing, which is composed by algorithms of principal components analysis, neural network and fast Fourier transform; and a post-processing which consists in the application of new phase current measurements in 5/4 of cycle, so this step is responsible to decrease the error of the ANN, increasing its reliability in fault detection. Also this confirmation step is obtained from a network output which is independent from the network of first step. The method has good performance and it does not give false fault identification when using the database of the simulation.

It is observed that identification with two steps eliminates some false fault detection errors that occurs mainly after load switching where the stator currents has an abrupt changes.

The fault identification can also be used to indicate some extra action to the control of the generator, in addition to protect equipment and the generator. As DFIG has control to the magnitude a angle of the magnetic flux, it's possible do some action to decrease the effect of the fault in the generator after its identification.

References

1. Alaya, J.B., Khedher, A., Mimouni, M.F.: Nonlinear vector control strategy applied to a variable speed DFIG generation system. In: Eighth International Multi-Conference on Systems, Signals & Devices, pp. 1–8 (2011)
2. Boldea, I.: Variable speed generators. Taylor&Francis, Timisoaria (2006)
3. El-helw, H.M., Tennakoon, B.S.: Vector Control of a Doubly Fed Induction Generator for Standalone Wind Energy Application. In: 2008 Wind Power to the Grid, EPE Wind Energy Chapter, pp. 1–6 (2008)
4. Lesic, V., Vasak, M., Peric, N., Wolbank, T., Joksimovic, G.: Fault-tolerant control of a blade-pitch wind turbine with inverter-fed generator. In: 1st IEEE International Symposium on Industrial Electronics, pp. 12097–12102 (2011)
5. Henao, H., Capolino, G.A., Casadei, D., Filippetti, F., Rossi, C.: Simulation of a doubly-fed induction machine for wind turbine generator fault analysis. In: 5th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, pp. 1–6 (2005)
6. Heraud, N., Guelle, I.S., Bennouna, O.: Fault diagnosis of variable speed wind turbine. In: 18th Mediterranean Conference on Control and Automation, Marrakech, Morocco, pp. 471–476 (2010)
7. Ramroop, S.A.D.: Fault detection of fault ride through for doubly-fed induction generator based wind energy systems. Master's thesis, Ryerson University, Toronto (2008)
8. Rothenhagen, K., Fuchs, F.W.: Grid Information Services for Distributed Resource Sharing. In: Current Sensor Fault Detection, Identification, and Reconfiguration for Doubly Fed Induction Generators, Taipei, Taiwan, pp. 1115–1120 (2007)
9. Barbi, I.: Teoria fundamental do motor de indução. Editora da UFSC, Florianopolis (1985)
10. Krause, P.C., Wasynczuk, O.S., Scott, D.: Analysis of electric machinery and drive systems. IEEE Press and Wiley-Interscience, NY (2002)
11. Leng, P., Yongdong, L., Jianyun, C., Guofeng, Y.: Vector control of a doubly fed induction generator for stand-alone ship shaft generator systems. In: Electrical Machines and Systems-ICEMS, Seoul, Korea, pp. 1033–1036 (2007)

Classification of Pharmaceutical Solid Excipients Using Self-Organizing Maps

Giovani Ângelo S. da Nóbrega, Marco V. M. Navarro, Túlio A. de Lima e Moura, Daiane Soares, José Alfredo F. Costa, Edilene P. Lavor, and Regivan H.N. Santiago

Universidade Federal do Rio Grande do Norte,
Avenida Senador Salgado Filho, 3000, Lagoa Nova, Natal, RN, 59078-970, Brazil
giovani.angelo@gmail.com, {marconavarro,moura,alfredo}@ufrnet.br
http://www.ufrn.br

Abstract. In pharmaceutical technology the study of rheological behavior of powders is an important step in pre formulation of solid dosage forms. These studies, particularly the flow behavior of powders, is an exhausting and very time consuming task, which requires tools that render this process faster while maintaining accuracy. The self-organizing map (SOM) is tool in the exploratory phase of data mining. It projects data from input space to low-dimensional regular grid which may be effectively utilized to visualize and explore properties of the data. This paper applies self-organizing map and K-means in order to analyze rheological characteristics of pharmaceutical solid excipients and their binary mixtures e.q. attapulgite, a natural clay candidate to solid excipient. Self-organizing map was able to classify effectively the excipients in ordered and coherent groups and classified attapulgite as a characteristic grouping having properties far distinct from the other groups of excipients. SOM enabled a reduction of experiments via exploratory data analysis about the rheological behavior of these powders.

Keywords: SOM, k-means, unsupervised networks, pharmaceutical solid excipients, attapulgite.

1 Introduction

Solid dosage forms such as tablets and capsules are the most frequently used and one of the most widely researched dosage forms due the patient compliance, cheap technology and high physicochemical stability [1]. Development of these pharmaceutical formulations depends greatly on the properties of their raw materials, particularly excipients e.g. diluents, fillers, lubricants, disintegrants; glidants; etc. Since manufacturing process of solid dosage forms requires the flow of a specific quantity of powder to a specific volume, good flow properties are therefore prerequisite for successful manufacturing. These properties are strongly dependent on factors such as size and shape of particles; adsorption of fluids (particularly air) and other contaminants on the surface; cohesive forces between particles; and bulk density of these powders. Indices of flowability are given by Hausner ratio (FH), Carr index (IC) and repose angle (α), whose acceptable values are ranging from 1.2 to 1.5 for FH; 5 up to 20 for IC; and below 40 for α ([2]). Both IC and FH are based on decrease of powder volume after tapping and so they give good indirect prediction of flowability of powder pharmaceutical raw materials.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 707–716, 2012.

Due to varied mineral composition, clays have different functions in the pharmaceutical field which includes their use as excipient. attapulgite clay is a magnesium silicate complex consisting of elongated and fibrous crystals whose functions are reported as adsorbent, disintegrant, carrier and release agent in solid formulations. However because it is a mineral from natural sources, its composition varies greatly and hence its excipient properties, particularly their rheological properties, must be elucidated before being proposed as excipients in solid formulations [3–8].

Data mining is a part of the process of knowledge discovery on multivariate data where unsupervised methods have found wide application. methods such as clustering techniques aim to find a valid and suitable organization for a set of multivariate data based on similarity between patterns [9]. Self-organizing maps (SOM) or Kohonen maps are the most important neural networks for data visualization and unsupervised classification. These properties can be exploited in various stages of development of a pharmaceutical formulation and also for sorting mixtures of excipients or solid carriers in accordance with specific properties. Accordingly, we evaluate the ability SOM maps in clustering of solid pharmaceutical excipients and particularly in the classification of attapulgite according to rheological properties e.g. IC and FH.

2 Clustering and Visualization Using SOM

The Self-Organizing Map (SOM) is a clustering and data visualization technique based on a neural network. As with other types of centroid-based clustering, the goal of SOM is to find a set of prototypes and to assign each object in the sample of dataset to the prototype that provides the best approximation. SOM is a clustering technique that enforces neighborhood relationships on the resulting prototypes. Because of this, clusters that are neighbors are more related to one another than clusters that are not. Such relationships facilitate the interpretation and visualization of the clustering results.

2.1 Self-Organizing Map Algorithm

The Self-Organizing Map algorithm is an unsupervised neural network that projects data according to a two-dimensional grid trying to preserve at most the input data topology in the output space. Neurons compete for each input pattern. The winning neuron updates itself and neighbor neurons. The effect of topology preservation is that similar data and clusters would be projected close to each other in the output space. The number of nodes in SOM array depends on the requirement to be able to best represent the input data vector set. Given the input data $\mathbf{x}$ and the vector m associated with each neuron, the best match unit c (BMU) represents the weight vector with the smallest distance to $\mathbf{x}$ [10]

$$\|\mathbf{x} - m_c\| = \min_i \{\|\mathbf{x} - m_i\|\} \tag{1}$$

where $\|\cdot\|$ is a distance measure (usually Euclidean). Next step of process, the weight vectors of the BMU as well as its neighboring nodes are moved closer to the input vector. The value of the attraction is driven by the learning rate. The SOM update rule for each weight vector i is

$$m_i(t+1) = m_i(t) + h_{ci}(t)[\mathbf{x}(t) - m_i(t)] \tag{2}$$

Where t denotes time, $\mathbf{x}(t)$ is the input vector randomly drawn from the input data set at time t and $h_{ci}(t)$ is the neighborhood kernel around the winner unit c at time t, a non-increasing function of time and the distance between unit i and BMU, further combining the learning rate $\alpha(t)$:

$$h_{ci}(t) = \alpha(t) \cdot h(\|r_c - r_i\|, t) \tag{3}$$

Where r_c and r_i are positions of neurons c and i on the grid generated.In sequence, The learning rate and neighborhood radius gradually decrease to zero according to a specified decay function. A number of SOM variants and improvements have been proposed since [10].

2.2 Visualization

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 [10, 11]. 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 [12], 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 [13].

The unified distance matrix (U-matrix) shows distances between prototype vectors of neighboring map units. Because they typically have similar prototype vectors, U-matrix is actually closely related to the single linkage measure. A U-matrix is constructed on top of a two-dimensional SOM grid. Let n be a neuron on the map, W_i be the set of immediate neighbors on the map, w_i the weight vector associated with neuron i, then:

$$\text{U-height(i)} = \sum_{j \in W_i} d(w_i, w_j) \tag{4}$$

Where $d(x,y)$ is the distance for input data space. The U-matrix is a display of the U-heights on top of the grid positions of neurons on the map. A U-matrix is usually displayed as a color level picture [14].

Vesanto and Alhoniemi [15] described both hierarchical agglomerative clustering and partitioning clustering using k-means for clustering of SOM. The experiments indicated that clustering the SOM instead of directly clustering the data is computationally effective approach. Other possible approaches for automated segmentation of SOM include usage of mathematical morphology operations, such as filtering and watershed transform on the U-matrix [16, 17]. Costa et al. [16–18] presented methods to detect automaticaly regions of neurons related to clusters in high dimensional space.

Table 1. Pure excipients

LABEL	CODEWORD	EXCIPIENT OR MIXTURE OF EXCIPIENTS
5	ATP	Attapulgite
6	CMC	Microcrystalline Cellulose
7	LCT	Lactose
8	EMG	Magnesium Stearate
9	TLC	Talc
10	MC101	Microcrystalline Cellulose

3 Experimental Process and Dataset

3.1 Experimental Process

Samples of attapulgite were collected from natural deposits in the region of Guadalupe in the state of Piauí, Brazil. Talc (Magnesita SA), magnesium stearate (Stearinerie Dubois), lactose monohydrate (Blanver), microcrystalline cellulose (Microcel 101®, Blanver), colloidal silicon dioxide (Aerosil 200) were used in pharmaceutical grade ([2]). Excipient mixtures were prepared manually by geometric dilution in the proportions showed in table 2. Bulk and compacted densities (D_{ap} and D_C respectively) were obtained by the ratio between bulk (V_0) or tapped volumes (10, 500 and 1250 taps) occupied by 30g of powder, by using a powder compactor (Nova tica 303-D[1]. V10 and V500) were chosen as bulk and compacted volume, respectively. Hausner ratio (FH), Carr index (IC) and repose angle (α) were calculated by the following functions:

$$FH = \frac{D_c}{D_{ap}} \tag{5}$$

$$IC = \frac{(Dc - Dap) \cdot 100}{Dc} \tag{6}$$

$$\tan \alpha = \frac{h}{r} \tag{7}$$

The database is used consisting of 900 samples for 9 attributes described and subdivided into 30 types of substances.

The database can be divided in two wide groups, according to the kind of substance. One of the groups is composed by pure substances 1, while the other one is constituted by the combination of those pure substances in proportions recommended in [2]. See Table 2.

The attributes of database match mass and the volume of the sample (V_0), the volumes of the sample after undergoing the compacting (V_{10},V_{10}, V_{500}, V_{1250}), bulk density (BD), the compacted density (DC), Carr index (CI) and Hausner ratio (FH), more information about the data we can see in Table 3.

[1] European Pharmacopoea 2000.

Table 2. Excipients and their mixtures

LABEL	CODEWORD	EXCIPIENT OR MIXTURE OF EXCIPIENTS
11	CMC/ATP	99/1 Microcrystalline Cellulose / Atapulgite 99% / 1%
12	CMC/ATP	98,5/1,5 Microcrystalline Cellulose / Atapulgite 98,5% / 1,5%
13	CMC/ATP	98/2 Microcrystalline Cellulose / Atapulgite 98% / 2%
14	CMC/TLC	99/1 Microcrystalline Cellulose / Talc 99% / 1%
15	CMC/TLC	98,5/1,5 Microcrystalline Cellulose / Talc 98,5% / 1,5%
16	CMC/TLC	98/2 Microcrystalline Cellulose / Talc 98% / 2%
17	CMC/EMG	99/1 Microcrystalline Cellulose / Magnesium Stearate 99% / 1%
18	CMC/EMG	98,5/1,5 Microcrystalline Cellulose / Magnesium Stearate 98,5% / 1,5%
19	CMC/EMG	98/2 Microcrystalline Cellulose / Magnesium Stearate 98% / 2%
20	LCT/TLC	99/1 Lactose / Talc 99% / 1%
21	LCT/TLC	98,5/1,5 Lactose / Talc 98,5% / 1,5%
22	LCT/TLC	98/2 Lactose/Talc 98% / 2%
23	LCT/EMG	99/1 Lactose/Magnesium Stearate 99% / 1%
24	LCT/EMG	98,5/1,5 Lactose/Magnesium Stearate 98,5% / 1,5%
25	LCT/EMG	98/2 Lactose/Magnesium Stearate 98% / 2%
26	LCT/AER	99/1 Lactose / Aerosil 99% / 1%
27	LCT/AER	98,5/1,5 Lactose / Aerosil 98,5% / 1,5%
28	LCT/AER	98/2 Lactose / Aerosil 98% / 2%
29	CMC/AER	99/1 Microcrystalline Cellulose / Aerosil 99% / 1%
30	CMC/AER	98,5/1,5 Microcrystalline Cellulose / Aerosil 98,5% / 1,5%
31	CMC/AER	98/2 Microcrystalline Cellulose / Aerosil 98% / 2%
32	LCT/ATP	99/1 Lactose / Atapulgite 99% / 1%
33	LCT/ATP	98,5/1,5 Lactose / Atapulgite 98,5% / 1,5%
34	LCT/ATP	98/2 Lactose / Atapulgite 98% / 2%

Table 3. Statistical information about the database

ATTRIBUTE	MINIMUM	MAXIMUM	STANDARD DEVIATION	MEAN
mass	29.9997	30.0004	0.0002	30.0001
volume (V_0)	39	1303	132.4637	83.6722
volume (V_{10})	37	123	23.6594	64.4444
volume (V_{500})	32	100	16.2303	51.6222
volume (V_{1250})	31	86	14.8677	50.5472
bulk density	0.2439	0.810819	0.1933	0.5339
compacted density	0.4054	0.937509	0.1811	0.6408
Car index	4.0000	41.2844	9.7990	18.3387
Hausner ratio	1.0417	1.703125	0.1616	1.2435

3.2 Self Organizing Map Settings

We use self-organizing map with the following settings: a 15x6 grid size, the initialization of the codebook method choosen was the linear method and the neighborhood function choosen was Gaussian. The tool was to implement MATLAB® (version 2011th for Linux) and SOM Toolbox Version 2.0 [14].

4 Result and Discussion

The U-matrix graph (figure 1) shows two main clusters, where the blue is concerned to lactose and its mixtures and the green is concerned to cellulose and mixtures thereof (label 6 and 7 in table 1, respectively); in their neighborhoods can be recognized two additional clusters plotted in orange and green. As visualized in the distance matrix, the two major clusters are clearly separated by a region of low-density of patterns, suggesting a classification of excipient species into two large main groups, the lactose group and the cellulose group.

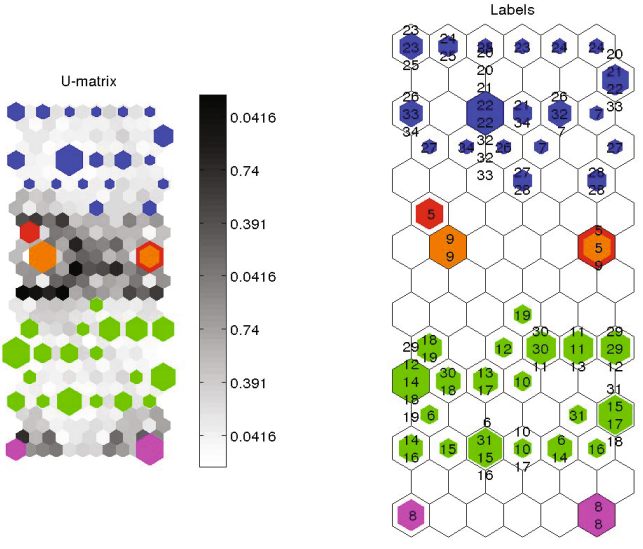
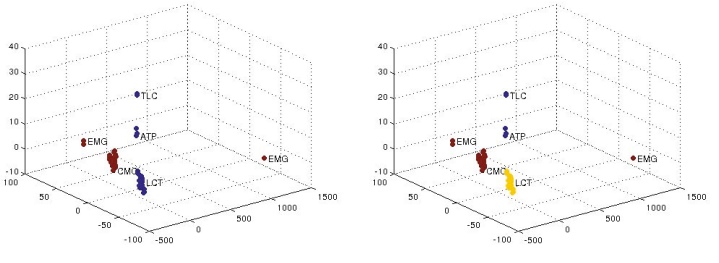


Fig. 1. U-matrix containing grouping class of excipients

Cellulose and lactose are both diluents in tablet and capsule formulation, but despite this the flow properties of cellulose are different and quite poorer than those of lactose. As cellulose and lactose were widely majority in their respective mixtures, the flow behavior of these mixtures assumes similar behavior to the major excipient and it follows



(a) A: K-means with two centers **(b)** B: K-means with three centers

Fig. 2. K-means result (k=3) after dimmension reduction via PCA [10]

the emergence of two larger clusters seen in the SOM graph. Classification performed by K-means (figure 2) with three clusters presents lactose and cellulose are separated into distinct groups.

The labeled EMG cluster corresponds to the magnesium stearate, which is used primarily as a lubricant in the manufacture of capsules and tablets and a powder, is cohesive and have very poor flow properties, poorer than that of cellulose. attapulgite and talc (lines 5 and 9 in Table 1) labeled in orange and red respectively in SOM map, were classified as having similar flow properties. This similarity was confirmed by K-means (figure 2) where the clusters even separated are aligned in the same region, which in a first approach leads to classify attapulgite in a category of excipients close to that of talc. This is particularly interesting because, unlike talc, whose properties and functions as pharmaceutical excipients are known for a long time, the attapulgite has not been used for this purpose and its use as a pharmaceutical excipient is only potential. which leads in a first approach to classify the ATP in the same category Figure 3 shows component planes associated to SOM in figure 1.

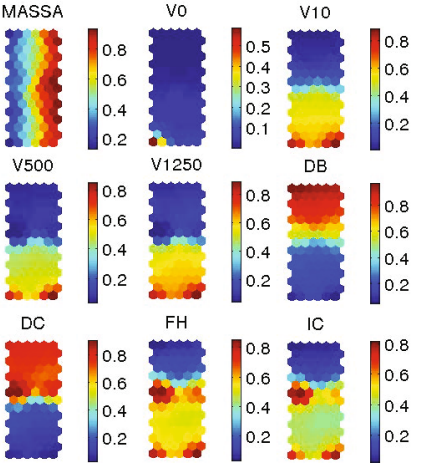


Fig. 3. SOM component planes related to figure 1

An important observation is that the classification according to FH and IC are virtually equal. Although FH and IC are different attributes, they directly influence the flow properties and compressibility of powders because they express both the ability of a powder flow and be compressed, and from this point of view even without a remarkable difference in their properties, there are two major categories of powders according to the compressibility. Figure 4 shows IC SOM plane after quantization (figure on the left) and with labels (figure on the right). Figure 5 refines the classification by attributes shows in figure 3. Free flow (ranging from 5 to 12) and good flow (ranging from 5 to 12) IC values are shown in dark blue and red in figure 4, respectively. Accordingly, all lactose mixtures were free flow powders except mixtures containing 1.5 and 2.0% of talc, whose cohesive effect of the latter was pointed out by SOM at concentrations higher than 1.5%; although these concentrations are not sufficient to cause appreciable impairment on lactose flow. On the other hand, fair flow (green label) is expected for powders

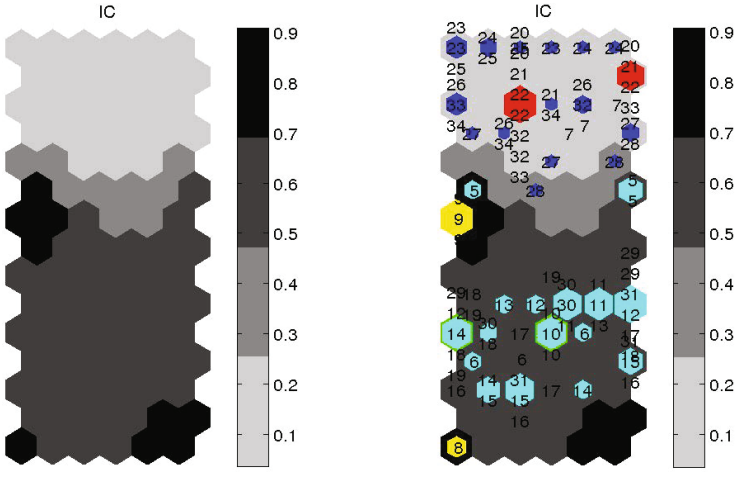


Fig. 4. IC SOM plane after quantization (left) and with labels (right)

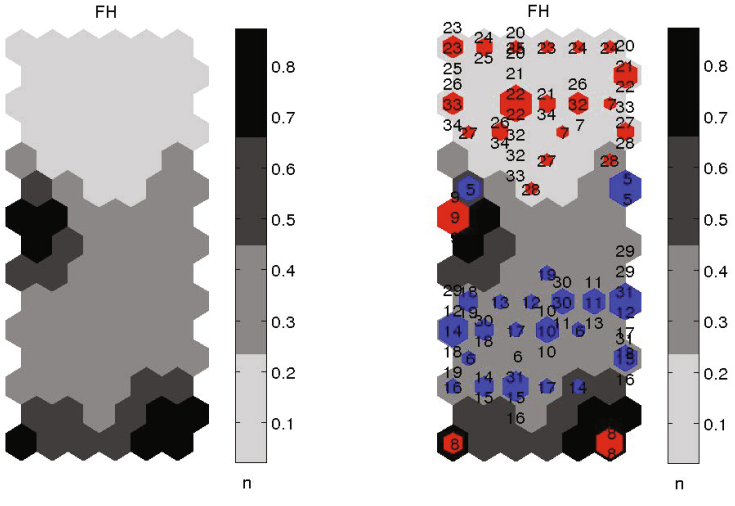


Fig. 5. FH SOM plane after quantization (left) and with labels (right)

with IC values ranging from 18 to 21, and is in agreement with expected cellulose flow behavior. Following this reason poor flow (IC values between 23 and 25, labeled in light blue) was assigned by SOM map for the majority of mixtures containing cellulose at high concentrations. Extremely poor (higher than 40, labeled in yellow) flow IC values were assigned to magnesium stearate and talc. Component quantization, such as done in IC plane (figure 4) presented good visual exploratory discrimination. Although in a first approach talc and attapulgite were classified into categories of excipients with similar flow properties, Figure 4 clearly distinct the attributes for attapulgite classifying it as a powder with poor flow in whereas extremely poor flow was assigned to talc. The

figure 5 shows FH SOM plane after quantization (on the left) and with labels (ont the right) and is similar to figure 4 which is based on IC values. The results are concordant because, in both cases, they express the ability of a powder to flow and corroborate the classification given to attapulgite in a category distinct from talc. Exploratory data analysis and SOM projections enabled a visual classification of rheological properties of excipients.

5 Conclusion

SOM maps was applied to visualization of solid pharmaceutical excipients and its physical mixtures with regard to rheological properties of the major components of a binary mixture of powders and for classification of attapulgite clay, a candidate material as solid pharmaceutical excipient. Furthermore, SOM maps were able to classify powders in range of values for a specific flow parameters which allowed to a qualitative qualification of pharmaceutical powders by simple visual analysis.

References

1. Gibson, M. (ed.): Pharmaceutical Preformulation and Formulation. Informa Healthcare, New York (2009)
2. Lima, E.M.: Propriedades reológicas aplicáveis á tecnologia de formas farmacêuticas sólidas, pp. 20–30 (2000)
3. Aguzzi, C., Cerezo, P., Viseras, C., Caramella, C.: Use of clays as drug delivery systems: Possibilities and limitations. Applied Clay Science 36(1-3), 22–36 (2007)
4. Carretero, M.I., Pozo, M.: Clay and non-clay minerals in the pharmaceutical industry part i. excipients and medical applications. Applied Clay Science 46(1), 73–80 (2009)
5. Carretero, M.I., Pozo, M.: Clay and non-clay minerals in the pharmaceutical and cosmetic industries part ii. active ingredients. Applied Clay Science 47(3-4), 171–181 (2010)
6. Legido, J., Medina, C., Lourdesmourelle, M., Carretero, M., Pozo, M.: Comparative study of the cooling rates of bentonite, sepiolite and common clays for their use in pelotherapy. Applied Clay Science 36(1-3), 148–160 (2007)
7. Viseras, C., Cerezo, P., Sanchez, R., Salcedo, I., Aguzzi, C.: Current challenges in clay minerals for drug delivery. Applied Clay Science 48(3), 291–295 (2010)
8. Viseras, C.: Pharmaceutical applications of some spanish clays sepiolite, palygorskite, bentonite /: some preformulation studies. Applied Clay Science (1999)
9. Costa, J.A.F.: Uma nova abordagem para visualização e detecção de das componentes. Journal of the Brazilian Neural Network Society 9(1), 20–31 (2011)
10. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1995)
11. Vesanto, J., Vesanto, J.: Using SOM in data mining (2000)
12. Ultsch, A.: Self-organizing neural networks for visualization and classification. In: Opitz, O., Lausen, B., Klar, R. (eds.) Information and Classification, pp. 307–313. Springer, London (1993)
13. Costa, J.A.F.: Clustering and Visualizing SOM Results. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) IDEAL 2010. LNCS, vol. 6283, pp. 334–343. Springer, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=1884499.1884540
14. Vesanto, J.: Neural network tool for data mining: SOM toolbox (2000)
15. Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. IEEE-NN 11(3), 586 (2000)

16. Costa, J.A., Netto, M.L.: Estimating the number of clusters in multivariate data by self-organizing maps. International Journal of Neural Systems 9(3), 195–202 (1999)
17. Costa, J.A.F., De Andrade Netto, M.L.: Clustering of complex shaped data sets via kohonen maps and mathematical morphology. In: Dasarathy, B.V. (ed.) Proceedings of SPIE—The International Society for Optical Engineering. Dept. of Comp. Eng./Indust. Automat., Sch. of Elec. and Comp. Engineering, vol. 4384, pp. 16–27. Universidade Estadual de Campinas (2001)
18. Sensing, R., Gonçalves, M.L., Netto, M.L.A., Júnior, J.Z., Costa, J.A.F.: A new method for unsupervised classification of remotely sensed images using kohonen self-organizing maps and agglomerative hierarchical clustering methods. International Journal of Remote Sensing X(X), 3171–3207 (2007)

Regional Models for Nonlinear System Identification Using the Self-Organizing Map

Amauri H. de Souza Junior and Guilherme A. Barreto

Federal University of Ceará, Department of Teleinformatics Engineering
Av. Mister Hull, S/N - Campus of Pici, Center of Technology, Fortaleza, Ceará, Brazil
amauri01@gmail.com, guilherme@deti.ufc.br

Abstract. Global modelling is a common approach to the problem of learning nonlinear dynamical input-output mappings. It consists in training a single multilayer neural network model using the whole dataset. On the other side of the spectrum stands the local modelling approach, in which the input space is divided into very small partitions and simpler (e.g. linear) models are trained, one per partition. In this paper, we propose a novel approach, called Regional Models (RM), that stands in between the global and local modelling ones. By following the approach by Vesanto and Alhoniemi [11], we first partition the input-output space using the Self-Organizing map (SOM), and then perform clustering over the prototypes of the trained SOM in order to find clusters of prototypes. Finally, a regional model is built for each cluster using the data vectors mapped to that cluster. The proposed approach is evaluated on two benchmarking problems and its performance is compared to those achieved by standard global and local models.

1 Introduction

Modern industrial plants have been the source of challeging tasks in dynamical system identification and control [9]. Designing control systems to achieve the level of quality demanded by current industry standards requires building accurate models of the plant being controlled. Building accurate models requires reliable data, usually in the form of input and output time series. Once such data are avaliable, they can be used for obtaining direct or inverse models of nonlinear systems by means e.g. of neural network architectures [7,1].

Although several techniques have been proposed and applied to the modelling and control of nonlinear systems [3,10], they can be roughly categorized in one out of two approaches: global and local modelling. Global models usually implement a single structure, such as a polynomial function or a single neural network model, that approximates the whole mapping between the input and the output of the system being identified. Global models constitute the mainstream in nonlinear system identification and control [8]. Local models have been a source of much interest because they have the ability to fit to the local shape of an arbitrary surface (i.e. mapping), which is particularly difficult when the dynamical system characteristics vary considerably throughout the state space. The input

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 717–724, 2012.

space is usually divided into smaller (localized) regions, each one being associated with a simpler model. To estimate the system output at a given time, a single model is chosen from the pool of available local models according to some criteria defined on the current input data.

Within the neural network literature, local modelling techniques have been implemented mostly through the use of the Self-Organizing Map (SOM) [4,2,10]. The results reported on those studies are rather appealing, indicating that SOM-based local models can be feasible alternatives to global models based on supervised neural network architectures, such as the MLP network, the RBF network and the recently proposed Extreme Learning Machine (ELM) [5].

In this paper, we propose a novel approach, called Regional Models (RM), that stands in between the global and local modelling ones. RM is motivated by the work of Vesanto and Alhoniemi [11] in the sense that, we first partition the input-output space using the SOM, and then we perform clustering over the prototypes of the trained SOM in order to find a set of K clusters of prototypes. Finally, K regional models are built, one for each cluster, using only the data vectors mapped to that cluster. Computer simulations on two benchmarking problems reveal that RM is a competitive alternative to nonlinear system identification.

The remainder of the paper is organized as follows. In Section 2, the modelling problem of interest is described. In Section 3, the RM fundamentals are then presented. Computer simulations and performance analysis of the proposed approach on two benchmarking problems are presented in Section 4. The paper is concluded in Section 5.

2 Inverse Modeling Approach

Let us assume that the dynamical system we are dealing with can be described mathematically by the NARX model[1] [8]:

$$y(t) = f\left[y(t-1), \ldots, y(t-p); u(t), u(t-1), \ldots, u(t-q+1)\right], \tag{1}$$

where $f(\cdot)$ is an unknown nonlinear mapping, $u(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$ denote, respectively, the input and output of the model at time step t, while $q \geq 1$ and $p \geq 1$ ($q \leq p$) are the input-memory and output-memory orders, respectively.

In this paper, we are interested in evaluating the RM approach on inverse system identification. For this purpose, we define

$$\mathbf{x}(t) = [u(t-1), \ldots, u(t-q); y(t-1), \ldots, y(t-p)] \in \Re^{p+q} \tag{2}$$

as the input regression vector and $u(t)$ as the target value at time t. Thus, the neural network models should implement the following inverse mapping:

$$u(t) = f^{-1}[\mathbf{x}(t)], \tag{3}$$

whose goal is to estimate the input of a given system based on previous values of the input and output variables.

[1] NARX stands for Nonlinear Autoregressive model with exogenous inputs.

3 The Proposed Approach: Regional Models

As mentioned in the introduction, regional modelling is an alternative approach to the more traditional local and global modelling techniques. A single global model may not be able to capture the dynamics of regions with high curvatures, smoothing out the details. On the one hand, global models are good to capture such details of the system dynamics, but on the other hand, they may suffer from underutilization, since some local models can be attached to unimportant regions of the system dynamics. The RM approach then comes out as an alternative to finding a tradeoff between the standard global local approaches.

In order to build regional models, one should follow the procedure introduced by Vesanto and Alhoniemi [11]. In this regard, the very first step requires training the SOM algorithm. The SOM learns from examples a mapping (projection) from a high-dimensional continuous input space $\mathcal{X}$ onto a low-dimensional lattice $\mathcal{A}$ of M neurons which are usually arranged as a rectangular 2-dimensional array. The map $i^*(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{A}$ is defined by the weight vectors $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}, \mathbf{w}_i \in \mathbb{R}^{p+q}$, and their corresponding coordinates $\mathbf{r}_i \in \mathbb{R}^2$ in the lattice $\mathcal{A}$. In addition, consider $\mathbf{X} = \{\mathbf{x}_l\}_{l=1}^N, \mathbf{x}_l \in \mathbb{R}^{p+q} \subset \mathcal{X}$, to be the available training data set.

Once the SOM is trained, the second step consists in finding partitions $\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_L$ $(L < M)$ of the training dataset by assigning a region $r \in \{1, ..., L\}$ to each neuron $i \in \mathcal{A}$. Here, this is carried out through the K-means algorithm, in accordance with Vesanto and Alhoniemi's procedure. For this purpose, we use $\mathbf{W}$ instead of $\mathbf{X}$ as input to the K-means algorithm, i.e. clustering is performed over the prototype vectors of the SOM.

Let the set of K prototypes comprising the K-means algorithm be denoted by $\mathbf{P}^K = \{\mathbf{p}_j\}_{j=1}^K$, $\mathbf{p}_j \in \mathbb{R}^{p+q+1}$. The optimal number of regions L is then estimated by the following search procedure:

$$L = \min_{K=2,\ldots,M-1} DB(\mathbf{W}, \mathbf{P}^K), \tag{4}$$

where $DB(.)$ denotes the Davies-Bouldin (DB) index function [6].

Once the optimal L is determined, the r-th cluster of SOM prototypes is comprised of all weight vectors $\mathbf{w}_i$ that are mapped onto the prototype $\mathbf{p}_r$ of the K-means algorithm. Mathematically, we have

$$\mathbf{W}_r = \{\mathbf{w}_i \in \mathbb{R}^{p+q} \mid \|\mathbf{w}_i - \mathbf{p}_r\| < \|\mathbf{w}_i - \mathbf{p}_j\|, \forall j \neq r, j = 1, \ldots, L\}. \tag{5}$$

Finally (for the second step), let $\mathbf{X}_r$ denote the set of all input vectors $\mathbf{x}_l$ whose closest SOM prototype belongs to $\mathbf{W}_r$.

The third and last step consists in training the L regional models using $\mathbf{X}_r$, $r = 1, \ldots, L$. For regional linear models, one can simply estimate the coefficient vector $\mathbf{a}_r$ of the r-th regional model by the least squares method:

$$\mathbf{a}_r = \left(\mathbf{X}_r^T \mathbf{X}_r\right)^{-1} \mathbf{X}_r \mathbf{t}_r, \qquad r = 1, ..., L, \tag{6}$$

where $\mathbf{t}_r \in \mathbb{R}^{N_r}$ is a vector of target values associated with $\mathbf{X}^r$ and N_r is the cardinality of $\mathbf{X}_r$ (i.e. the number of input vectors in $\mathbf{X}_r$).

If we choose to build nonlinear regional models, one can train the r-th regional model using the input vectors in $\mathbf{X}_r$ and the corresponding target values in $\mathbf{t}_r$, for $r = 1, \ldots, L$. In this paper, we use ELM networks in order to building the nonlinear regional models. Thus, we denote by *Regional Linear Model* (RLM) a regional model built using linear models. By the same token, we will use the term *Regional Extreme Learning Machine Models* (RELM) to denote regional models comprised of ELM networks.

Once the regional models are trained, the test procedure for a given input signal $\mathbf{x}(t) \in \mathbb{R}^{p+q}$ consists in computing the predicted output value using the suitable regional model. For RLMs, the output value is computed as

$$\hat{y}_{RLM}(t) = \mathbf{a}_{r^*}^T \mathbf{x}(t), \tag{7}$$

while for RELMs, the output value is computed as

$$\hat{y}_{RELM}(t) = \text{ELM}_{r^*}(\mathbf{x}(t)), \tag{8}$$

where, for both cases, $r^*(t)$ is the index of the regional model associated with the prototype $\mathbf{p}_{r^*}$. More formaly, we have

$$r^*(t) = \arg \min_{r=1,\ldots,L} \|\mathbf{x}(t) - \mathbf{p}_r\|. \tag{9}$$

4 Computer Simulations

In this section, we present the results of a comprehensive computer simulations carried out with two input-output datasets commonly used for benchmarking purposes in system identification. The proposed RLM and RELM models are compared to the MLP-based and ELM-based global models and the KSOM [2] and LLM [12] local models in the task of inverse system identification. All the MLP- and ELM-based models were implemented from scratch using Matlab script language. In particular, we used the SOM toolbox for Matlab to build the SOM-based models.

All the models are evaluated via the statistics of the resulting normalized mean-squared estimation error (NMSE), $NMSE = \frac{\sum_{t=1}^{N} e^2(t)}{N \cdot \hat{\sigma}_u^2} = \frac{\hat{\sigma}_e^2}{\hat{\sigma}_u^2}$, where $\hat{\sigma}_u^2$ is the variance of the original time series $\{u(t)\}_{t=1}^N$ and N is the length of the sequence of residuals.

Two benchmarking datasets were used to evaluate all the models, namely: (i) the hydraulic actuator dataset[2] (input u: valve position, output y: oil pressure), and (ii) the flexible robot arm dataset[3] (input u: reaction torque of the structure, output y: acceleration of the flexible arm).

[2] Download from `http://homes.esat.kuleuven.be/~smc/daisy/daisydata.html`.
[3] Download from `http://homes.esat.kuleuven.be/~smc/daisy/daisydata.html`.

Table 1. Performance results for the hydraulic actuator data

Neural Models	NMSE			
	mean	*min*	*max*	*variance*
RLM	1.14e-004	1.13e-004	1.38e-004	2.09e-011
RELM	1.14e-004	1.13e-004	1.27e-004	9.25e-010
ELM	0.0012	0.0001	0.0026	1.04e-007
KSOM	0.0019	0.0002	0.0247	1.15e-005
LLM	0.0347	0.0181	0.0651	1.58e-004
Linear	0.0380	0.0380	0.0380	0.0083
MLP-LM	0.0722	0.0048	0.3079	0.0041
MLP-1h	0.3485	0.2800	0.4146	4.96e-004
MLP-2h	0.3516	0.0980	2.6986	0.0963

4.1 Results on the Hydraulic Actuator Dataset

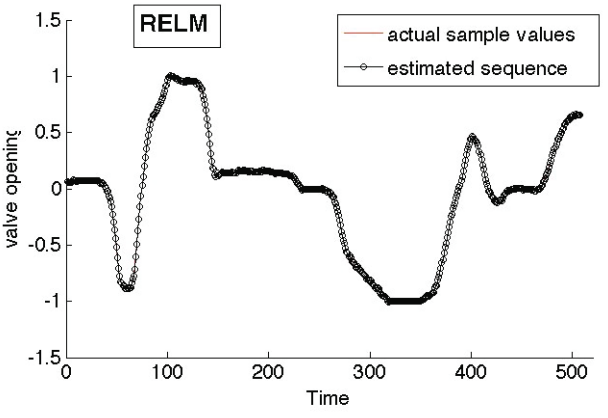
The RLM and RELM models are compared with an one-hidden-layered MLP trained by the backpropagation algorithm (MLP-1h), another one-hidden-layered MLP trained by the Levenberg-Marquardt (MLP-LM) algorithm and a two-hidden-layered MLP (MLP-2h) trained by the backpropagation algorithm. For all MLP-based global models, the hidden neurons use the hyperbolic tangent activation function, while the output neuron uses a linear one. For the sake of completeness, we also evaluated the performance of a global linear inverse model, trained by the LMS algorithm.

Initially, a systematic search for the optimal number of hidden neurons that leads to the smallest generalization error was carried out, with values ranging from 2 to 30. Eventually, the best configurations of the MLP-1h and MLP-LM and ELM models have 20 hidden neurons. For the MLP-2h, the number of hidden neurons in the second layer is heuristically set to half the number of neurons in the first hidden layer, respectively 10 and 20 neurons at each hidden layer. The MLP-based models were trained with constant learning rate equal to 0.1.

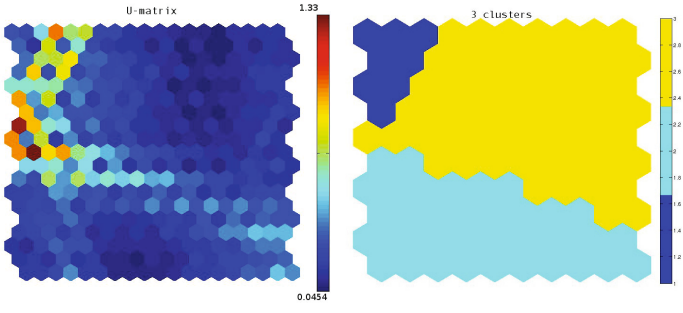
The models are trained using the first 512 samples of the input/output time series and tested with the remaining 512 samples. Input/output time series are rescaled to the $[-1, +1]$ range. Memory orders were set to $p = 5$ and $q = 4$, respectively. For each SOM-based model, the number of neurons was set to $M = 20$. The initial and final learning rates are set to $\alpha_0 = 0.5$ and $\alpha_T = 0.01$. The initial and final values of the gaussian neighborhood function radius were $\sigma_0 = M/2$ and $\sigma_T = 0.001$. The learning rate α' (LLM model) was set to 0.01.

The regional models were comprised of a SOM grid of 10×10 neurons, using the aforementioned parameters. The number of clusters found by the K-means clustering algorithm in combination with the minimum DB index was $L = 3$.

The obtained results are shown in Table 1, where are displayed the mean, minimum, maximum and variance of the NMSE values, collected over 100 training/testing runs, with the weights of the neural models randomly initialized at each run. In this table, the models are sorted in decreasing order according to the mean NMSE values.



(a)



(b) (c)

Fig. 1. (a) Typical estimated sequences of the valve position provided by RELM model. Open circles 'o' denote actual sample values, while the solid line indicates the estimated sequence. (c) U-Matrix associated with the SOM. (b) Clusters of SOM prototypes found by the K-means algorithm using $K = 3$.

One can easily note that the performances of RLM and RELM models on this real-world dataset are far better than those of all other models. The ELM and KSOM model had also acceptable performances on this dataset. A curious fact is that the linear global model performed better than the three global MLP-based models (MLP-LM, MLP-1h, MLP-2h). Among the MLP-based models, the use of second-order information could also explain the better performance of the MLP-LM, which uses information extracted from the Hessian matrix.

Figure 1 shows typical results provided by the RELM model. Figure 1 (a) shows the sequence generated by the RELM model, where the actual and estimated sequences are almost indistinguishable because of the small estimation error. Figures 1 (b) and (c) show, respectively, the U-matrix associated with the trained SOM and the clusters of SOM prototypes found by the K-means algorith using $K = L = 3$ as indicated by the DB index.

Table 2. Performance results for the robotic arm data

Neural Models	NMSE			
	mean	*min*	*max*	*variance*
KSOM	0.0064	0.0045	0.0117	1.83e-006
RELM	0.0070	0.0067	0.0072	1.62e-008
RLM	0.0071	0.0068	0.0074	2.88e-008
ELM	0.0285	0.0171	0.0457	2.73e-005
MLP-LM	0.1488	0.0657	0.4936	0.0107
MLP-1h	0.1622	0.1549	0.1699	1.03e-005
LLM	0.3176	0.2685	0.3558	2.23e-004
Linear	0.3848	0.3848	0.3848	0.0445
MLP-2h	0.6963	0.5978	1.5310	0.0368

4.2 Results on the Robot Arm Dataset

For this dataset the best configurations found for the MLP-1h and MLP-LM models have 30 hidden neurons. For the MLP-2h, the number of neurons in the second hidden layer was again set to half the number of neurons in the first hidden layer, respectively 15 and 30 neurons. The learning rate for the MLPs was set to 0.1. For the ELM model, the number of hidden neurons was set to 30.

The number of neurons for the KSOM and LLM algorithms was set to $N = 30$. For each SOM-based model, the initial and final learning rates were set to $\alpha_0 = 0.5$ and $\alpha_T = 0.01$. The initial and final values of radius of the neighborhood function are $\sigma_0 = N/2$ and $\sigma_T = 0.001$, and the learning rate α' (LLM model) was set to 0.1. The regional models were composed of a SOM grid of 10×10 neurons, using the aforementioned parameters. The optimal number of clusters found was $L = 9$.

For this dataset, the best performances were achieved by the KSOM, RLM and RELM with the two last ones presenting smaller variances. They have shown similar performance. Among the global models, the ELM model achieved the best performance. The LLM model performed only better than the linear and MLP-2h models. The performances of these three models were very poor.

5 Conclusion

In this paper we have introduced a novel approach to building multiple models for system identification, named Regional Models. A comprehensive evaluation of the proposed approach was carried out for the task of inverse modelling of two benchmarking dynamical systems. Their performances based on normalized mean-squared errors were compared to those achieved by standard MLP-based global models and classical algorithms in local linear modelling.

The main general conclusion of the presented experiments is that regional models can be considered a promising approach for nonlinear dynamical systems modelling. Currently, we are evaluating the robustness of the RLM and RELM to outliers and developing variants of them based on robust regression techniques (e.g. M-estimation).

References

1. Barreto, G.A., Araújo, A.F.R.: Identification and control of dynamical systems using the self-organizing map. IEEE Transactions on Neural Networks 15(5), 1244–1259 (2004)
2. Barreto, G.A., Souza, L.G.M.: Adaptive filtering with the self-organizing maps: A performance comparison. Neural Networks 19(6), 785–798 (2006)
3. Chen, J.-Q., Xi, Y.-G.: Nonlinear system modeling by competitive learning and adaptive fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics-Part C 28(2), 231–238 (1998)
4. Cho, J., Principe, J., Erdogmus, D., Motter, M.: Quasi-sliding mode control strategy based on multiple linear models. Neurocomputing 70(4-6), 962–974 (2007)
5. Huang, G.B., Zhu, Q.Y., Ziew, C.K.: Extreme learning machine: Theory and applications. Neurocomputing 70(1–3), 489–501 (2006)
6. Jain, A.K., Dubes, R.C., Chen, C.C.: Bootstrap techniques for error estimation. IEEE Transactions on Pattern Analysis and Machine Ingelligence 9(5), 628–633 (1987)
7. Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks 1(1), 4–27 (1990)
8. Norgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural Networks for Modelling and Control of Dynamic Systems. Springer (2000)
9. Peng, H., Nakano, K., Shioya, H.: A comprehensive review for industrial applicability of artificial neural networks. IEEE Transactions on Control Systems Technology 15(1), 130–143 (2007)
10. Principe, J.C., Wang, L., Motter, M.A.: Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control. Proceedings of the IEEE 86(11), 2240–2258 (1998)
11. Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. IEEE Transactions on Neural Networks 11(3), 586–600 (2000)
12. Walter, J., Ritter, H., Schulten, K.: Non-linear prediction with self-organizing map. In: Proceedings of the IEEE International Joint Conference on Neural Networks, IJCNN 1990, vol. 1, pp. 587–592 (1990)

Fast Opposite Maps: An Iterative SOM-Based Method for Building Reduced-Set SVMs

Ajalmar R. da Rocha Neto[1] and Guilherme A. Barreto[2]

[1] Federal Institute of Ceará, Maracanaú Campus, Teleinformatics Department, Av. Contorno Norte, S/N, Maracanaú, Ceará, Brazil
ajalmar@ifce.edu.br

[2] Federal University of Ceará, Department of Teleinformatics Engineering Center of Technology, Campus of Pici, Fortaleza, Ceará, Brazil
guilherme@deti.ufc.br

Abstract. Opposite Maps (OM) is a method that can be used to induce sparse SVM-based and LS-SVM-based classifiers. The main idea behind the OM method is to train two Self-Organizing Maps (SOM), one for each class, $\mathcal{C}_{-1}$ and $\mathcal{C}_{+1}$, in a binary classification context and then, for the patterns of one class, say $\mathcal{C}_{-1}$, to find the closest prototypes among those belonging to the SOM trained with patterns of the other class, say $\mathcal{C}_{+1}$. The subset of patterns mapped to the selected prototypes in both SOMs form the reduced set to be used for training SVM and LSSVM classifiers. In this paper, an iterative method based on the OM, called Fast Opposite Maps, is introduced with the aim of accelerating OM training time. Comprehensive computer simulations using synthetic and real-world datasets reveal that the proposed approach achieves similar results to the original OM, at a much faster pace.

Keywords: SVM, LS-SVM, Support Vectors, SOM and Reduced Set.

1 Introduction

A theoretical advantage of kernel-based machines concerns the production of sparse solutions [11]. By sparseness we mean that the induced classifier can be written in terms of a relatively small number of input examples, the so-called support vectors (SVs). In practice, however, it is observed that the application of different training approaches to the same kernel-based machine over identical training sets yield disctint sparseness [2], i.e. produce solutions with a greater number of SVs than are strictly necessary.

To handle this issue, several Reduced Set (RS) methods have been proposed to alleviate this problem, either by eliminating less important SVs or by constructing a new (smaller) set of training examples, often with minimal impact on performance [1,4,14,6]. An alternative to standard SVM formulation is the Least Squares Support Vector Machine (LS-SVM) [13], which leads to solving linear KKT systems[1] in a least square sense. The solution follows directly from solving a linear equation system, instead of a quadratic programming optimization

[1] Karush-Kuhn-Tucker systems.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 725–732, 2012.

problem. As we know, it is in general easier and less computationally intensive to solve a linear system than a QP problem. On the other hand, the introduced modifications also result in loss of sparseness of the induced SVM. It is common to have all examples of the training data set belonging to the set of the SVs. To mitigate this drawback, several pruning methods have been proposed in order to improve the sparseness of the LS-SVM solution [7,3,12].

In this paper, we introduce an interative method based on the recently proposed Opposite Maps (OM) [10] for building reduced-set SVM/LSSVM classifiers. Our goal is to induce sparse classifiers at a faster training pace than that achieved by the original OM method.

This paper is organized as follows. In Section 2 we review the fundamentals of the SVM and LS-SVM classifiers. In Section 3 we describe the proposed approach method. The resulting FOM-SVM and FOM-LSSVM approaches are presented in Section 4. Simulations and results are shown in Section 5. The paper is concluded in Section 6.

2 SVM and LSSVM Classifiers

Consider a training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{L}$, so that $\mathbf{x}_i \in \mathbb{R}^p$ is an input vector and $y_i \in \{-1, +1\}$ are the corresponding class labels. For soft margin classification, the SVM dual problem is defined as

$$\begin{aligned} \max J(\boldsymbol{\alpha}) &= \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i d_j \mathbf{x}_i^T \mathbf{x}_j, \\ \text{subject to} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C. \end{aligned} \tag{1}$$

where $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^{L}$ and $\boldsymbol{\beta} = \{\beta_i\}_{i=1}^{L}$ are Lagrange multipliers. The constant $C \in \mathbb{R}$ is a cost parameter that controls the trade-off between allowing training errors and forcing rigid margins. Once we have the values of the Lagrange multipliers, the output can be calculated based on the classification function described as

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{L} \alpha_i y_i \mathbf{x}^T \mathbf{x}_i + b \right). \tag{2}$$

where b is the bias.

As an alternative to the standard SVM formulation, the primal problem for the LS-SVM approach [13] is given by

$$\begin{aligned} \min_{\mathbf{w}, \xi_i} & \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^{L} \xi_i^2 \right\}, \\ \text{subject to} \quad & y_i[(\mathbf{w}^T \mathbf{x}_i) + b] = 1 - \xi_i, i = 1, \ldots, L \end{aligned} \tag{3}$$

where $\{\xi_i\}_{i=1}^{L}$ are the slack variables and γ is a positive cost parameter similar to the parameter C in the SVM formulation (see Eq. (1)).

By differentiating the lagrangian function associated with Eq. (3) with respect to all the parameters of interest ($\mathbf{w}$, b, α_i and ξ_i), the LS-SVM problem can be written as a linear system $\mathbf{Ax} = \mathbf{B}$ as follows:

$$\left[\begin{array}{c|c} 0 & \mathbf{y}^T \\ \hline \mathbf{y} & \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I} \end{array}\right] \left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{1} \end{array}\right] \tag{4}$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{L \times L}$ is a matrix whose entries are $\Omega_{i,j} = y_i y_j \mathbf{x}_i{}^T \mathbf{x}_j$, $i, j = 1, \ldots, L$. In addition, $\mathbf{y} = [y_1 \ \cdots \ y_L]^T$ and the symbol $\mathbf{1}$ denotes a vector of ones with dimension L. The LS-SVM output can also be computed as in Eq. (2).

3 The Proposed Approach: Fast Opposite Maps

The OM method [10] was proposed in order to find a reduced set of training vectors to induce SVM and LS-SVM classifiers. For a classification problem with K classes, the original OM requires K self-organizing maps (SOM) [5] to be trained, one for each available class. It is worth pointing out, however, that any vector quantization algorithm other than the SOM can be used by the OM method.

After a number of experiments, it was observed that the OM algorithm could be improved even further if one applies it iteratively, i.e. the reduced dataset achieved by the current run of the OM (say, at iteration j) serves as input to another run of the OM algorithm, and so on. The details of the iterative version of the OM method, henceforth called fast opposite maps (FOM), are described below for a binary classification problem ($K = 2$).

Initialization - Specify the maximum number of iterations n and set $j = 1$. Also, let $\mathcal{D}_j$ denote the reduced dataset at iteration j. For $j = 1$, we set $\mathcal{D}_j = \mathcal{D}$, where $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ is the original dataset. Finally, let SOM$^{(j)}$ denote a $s_j \times s_j$ SOM network, where s_j is an escalar value which must be set at iteration j, $j = 1, \ldots, n$.

STEP 1 - Split the available data set $\mathcal{D}_j = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ into two subsets:

$$\mathcal{D}_j^{(1)} = \{(\mathbf{x}_i, y_i) | y_i = +1\}, \quad i = 1, \ldots, l_1 \qquad \text{(for class 1)} \tag{5}$$

$$\mathcal{D}_j^{(2)} = \{(\mathbf{x}_i, y_i) | y_i = -1\}, \quad i = 1, \ldots, l_2 \qquad \text{(for class 2)} \tag{6}$$

where l_1 and l_2 are the cardinalities of the subsets $\mathcal{D}_j^{(1)}$ and $\mathcal{D}_j^{(2)}$, respectively.

STEP 2 - Train a SOM network using the subset $\mathcal{D}_j^{(1)}$ and another SOM using the subset $\mathcal{D}_j^{(2)}$. Refer to the trained networks as SOM-1$^{(j)}$ and SOM-2$^{(j)}$.

STEP 3 - For each vector $\mathbf{x}_i \in \mathcal{D}_j^{(1)}$ find its corresponding BMU in SOM-1$^{(j)}$. Then, prune all *dead neurons*[2] in SOM-1$^{(j)}$. Repeat the same procedure for each vector $\mathbf{x}_i \in \mathcal{D}_j^{(2)}$: find the corresponding BMUs in SOM-2$^{(j)}$ and prune all the dead neurons. Refer to the pruned networks as PSOM-1$^{(j)}$ and PSOM-2$^{(j)}$.

STEP 4 - At this step the BMUs for the data subsets are searched within the set of prototypes of the *opposite map*.

[2] Neurons which have never been selected as the BMU for any vector $\mathbf{x}_i \in \mathcal{D}^{(1)}$.

Step 4.1 - For each $\mathbf{x}_i \in \mathcal{D}_j^{(1)}$ find its corresponding BMU in PSOM-$2^{(j)}$:

$$c_i^{(2)} = \arg\min_{\forall j} \|\mathbf{x}_i - \mathbf{w}_j^{(2)}\|, \quad i = 1, \ldots, l_1, \tag{7}$$

where $\mathbf{w}_j^{(2)}$ is the j-th prototype vector in PSOM-$2^{(j)}$. Thus, $c_i^{(2)}$ denotes the index of the BMU in PSOM-$2^{(j)}$ for the i-th example in $\mathcal{D}_j^{(1)}$.

Step 4.2 - For each $\mathbf{x}_i \in \mathcal{D}_j^{(2)}$ find its corresponding BMU in PSOM-$1^{(j)}$:

$$c_i^{(1)} = \arg\min_{\forall j} \|\mathbf{x}_i - \mathbf{w}_j^{(1)}\|, , \quad i = 1, \ldots, l_2, \tag{8}$$

where $\mathbf{w}_j^{(1)}$ is the j-th prototype vector in PSOM-$1^{(j)}$. Thus, $c_i^{(1)}$ denotes the index of the BMU in PSOM-$1^{(j)}$ for the i-th example in $\mathcal{D}_j^{(2)}$.

STEP 5 - Let $\mathcal{C}^{(2)} = \{c_1^{(2)}, c_2^{(2)}, \ldots, c_{l_2}^{(2)}\}$ be the index set of all BMUs found in Step 4.1, and $\mathcal{C}^{(1)} = \{c_1^{(1)}, c_2^{(1)}, \ldots, c_{l_1}^{(1)}\}$ be the index set of all BMUs found in Step 4.2.

STEP 6 - At this step the new reduced set $\mathcal{D}_j$ is formed. For this purpose, three substeps are required.

Step 6.1 - For each PSOM-$1^{(j)}$ unit in $\mathcal{C}^{(1)}$ find its nearest neighbor among the data vectors $\mathbf{x}_i \in \mathcal{D}_j^{(1)}$. Let $\mathcal{X}^{(1)}$ be the subset of nearest neighbors for the PSOM-$1^{(j)}$ units in $\mathcal{C}^{(1)}$.

Step 6.2 - For each PSOM-$2^{(j)}$ unit in $\mathcal{C}^{(2)}$ find its nearest vector $\mathbf{x}_i \in \mathcal{D}_j^{(2)}$. Let $\mathcal{X}^{(2)}$ be the subset of nearest neighbors for the PSOM-$2^{(j)}$ units in $\mathcal{C}^{(2)}$.

Step 6.3 - Set $j = j+1$ and define the new reduced set as $\mathcal{D}_j = \mathcal{X}^{(1)} \cup \mathcal{X}^{(2)}$.

STEP 7 - If $j < n$, go to Step 1. Otherwise, finish the algorithm and return the final reduced dataset $\mathcal{X}_{rs} = \mathcal{D}_j$.

4 The FOM-SVM and FOM-LSSVM Classifiers

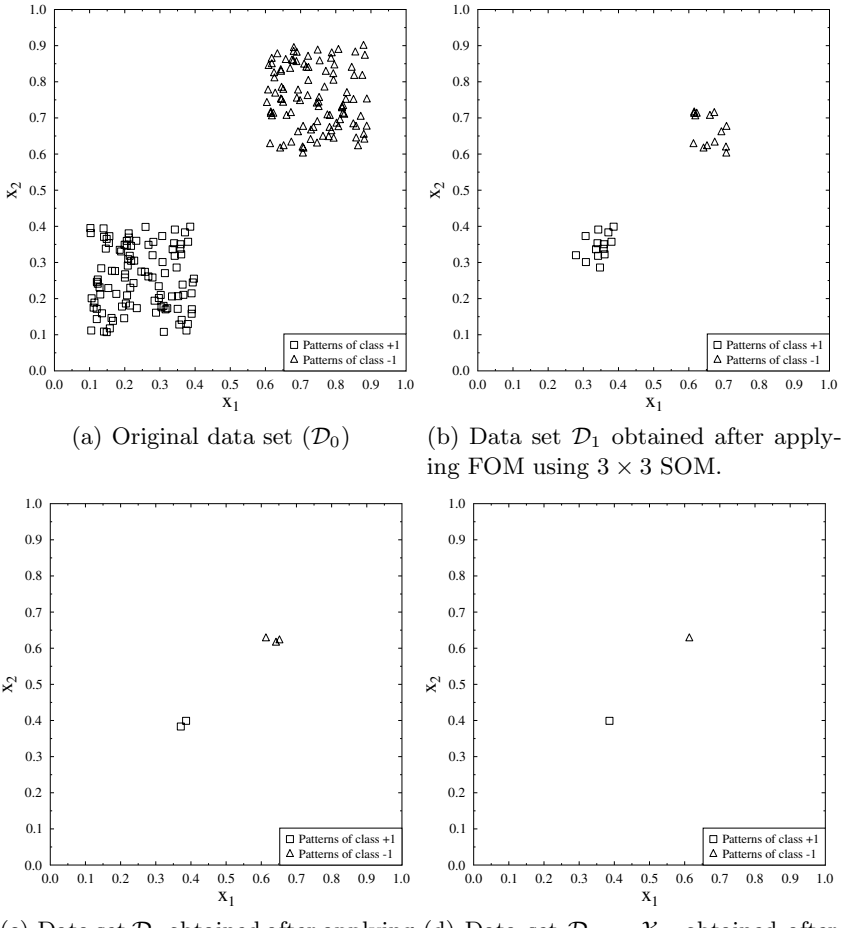
In this paper, we use the SMO algorithm [8], which is usually much faster than standard numerical techniques used to solve the quadratic programming optimization problem required by training SVMs. By using the (F)OM algorithm, it is possible to speed up the SMO algorithm even further. The main idea is to deliver to the SMO algorithm an "almost solved problem", since for all data examples out of the reduced set (i.e. $\mathbf{x}_i \notin \mathcal{X}_{rs}$) their Lagrange multipliers are set to zero, the SMO algorithm is run only over the data examples belonging to the reduced set. This approach is henceforth called FOM-SVM.

For the LS-SVM approach, we cannot set the Lagrange multipliers of the data examples out of the reduced set to zero, since the LS-SVM usually provide non-sparse solutions. However, the Lagrange multipliers associated with data examples located along the class border or within the overlapping region between the classes indeed assume higher values, i.e. $\alpha_i \gg 0$. These instances are the most likely to be the SVs.

Based on this property, we use the (F)OM algorithm to build a modified version of the LS-SVM formulation shown in Eq. (4). Instead of building the original square matrix $\mathbf{A}$ and inverting it to find $\mathbf{x}$, we build a *non-square* reduced matrix $\mathbf{A}_{rs}$ using the data examples belonging to the reduced set $\mathcal{X}_{rs}$. Thus, since $\mathbf{A}_{rs}$ is a non-square matrix, we solve for $\mathbf{x}$ using the pseudoinverse method: $\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{B}$, where $\mathbf{A}^{\dagger} = (\mathbf{A}_{rs}^T\mathbf{A}_{rs})^{-1}\mathbf{A}_{rs}^T$. This approach is henceforth called FOM-LSSVM.

5 Simulations and Discussion

For all experiments to be described, 80% of the data examples were randomly selected for training purposes. The remaining 20% of the examples were used



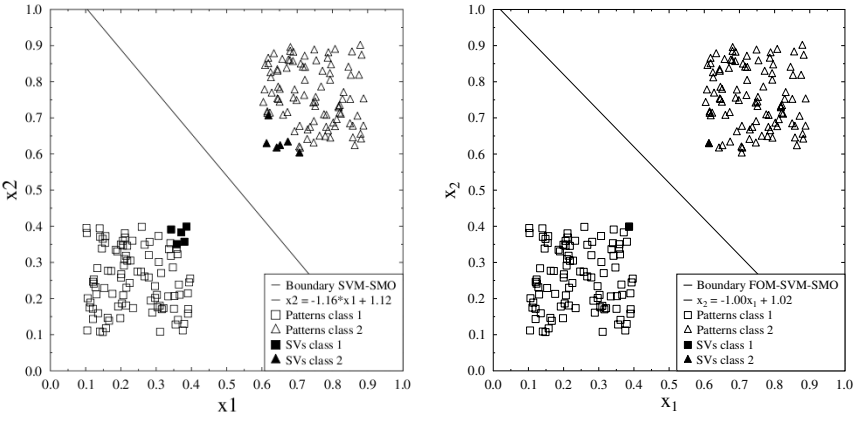
(a) Original data set ($\mathcal{D}_0$)

(b) Data set $\mathcal{D}_1$ obtained after applying FOM using 3×3 SOM.

(c) Data set $\mathcal{D}_2$ obtained after applying FOM using 3×3 SOM.

(d) Data set $\mathcal{D}_3 = \mathcal{X}_{rs}$ obtained after applying FOM using 2×2 SOM.

Fig. 1. Fast Opposite Maps Method applied iterarively to an artificial problem



(a) Decision line and SVs for the standard SVM trained with the SMO

(b) Decision line and SVs for the FOM-SVM classifier

Fig. 2. OM-SVM and FOM-SVM classifiers applied to an artificial problem

Table 1. Results for the SVM, OM-SVM and FOM-LSSVM classifiers

Data Set	Model	C	Tol.	Map Sizes	Accuracy	Train. Size	SVs #	SV Red.	Train. Time
VCP	SVM	2.5	0.001	–	85.2 ± 4.0	248	86.8	–	10.4
VCP	OM-SVM	2.5	0.001	[10]	84.6 ± 4.3	248	69.2	20.3%	43.6
VCP	FOM-SVM	2.5	0.001	[3, 3, 5, 6]	84.9 ± 4.6	248	67.5	22.2%	16.5
								Reduction of training time =	62.2%
BC	SVM	0.04	0.001	–	97.2 ± 1.0	546	61.7	–	1.5
BC	OM-SVM	0.04	0.001	[10]	95.9 ± 1.4	546	46.2	25.1%	98.5
BC	FOM-SVM	0.04	0.001	[3, 5]	96.4 ± 1.3	546	47.5	23.0%	11.9
								Reduction of training time =	87.9%
PIMA	SVM	2.5	0.01	–	77.2 ± 2.6	614	321.0	–	23.8
PIMA	OM-SVM	2.5	0.01	[10]	76.6 ± 2.7	614	293.0	8.7%	145.0
PIMA	FOM-SVM	2.5	0.01	[3, 4, 8]	76.8 ± 2.7	614	297.8	7.2%	111.3
								Reduction of training time =	23.2%

Table 2. Results for the LS-SVM, OM-LSSVM and FOM-LSSVM classifiers

Data Set	Model	γ	Map Sizes	Accuracy	Train. Size	SVs#	SV Red.	Train. Time
VCP	LS-SVM	0.04	–	81.2 ± 4.7	248	248.0	–	0.2
VCP	OM-LSSVM	0.04	[10]	80.3 ± 4.6	248	109.4	55.9%	41.7
VCP	FOM-LSSVM	0.04	[3, 3, 5, 6]	81.4 ± 4.3	248	107.2	56.8%	15.0
							Reduction of training time =	64.0%
BC	LS-SVM	0.04	–	96.9 ± 1.6	546	546.0	–	3.8
BC	OM-LSSVM	0.04	[10]	94.3 ± 4.8	546	54.3	90.1%	101.1
BC	FOM-LSSVM	0.04	[4, 5, 6, 6]	95.3 ± 1.7	546	23.3	95.7%	19.2
							Reduction of training time =	81.0%
DIA	LS-SVM	0.04	–	76.1 ± 3.3	614	614.0	–	5.5
DIA	OM-LSSVM	0.04	[10]	78.3 ± 3.1	614	430.6	29.9%	116.4
DIA	FOM-LSSVM	0.04	[3, 6, 9]	76.4 ± 3.2	614	423.3	31.1%	99.6
							Reduction of training time =	14.4%

for testing the classifiers' generalization performances. All simulations were conducted using a standard 2-D SOM, hexagonal neighborhood, Gaussian neighborhood function, with random weight initialization.

For tests with the original OM method, we trained two SOMs with fixed 10×10 map grid, for 80 epochs with initial and final neighborhood radius (learning rate) of 5 (0.5) and 0.1 (0.01), respectively. For SVM-like classifiers we used linear kernels. The map grid size for each FOM-SVM and FOM-LSSVM classifiers is presented in the result tables.

Initially, as a proof of concept, we have applied the FOM method to an artificial problem, consisting of a linearly separable two-dimensional data set. Data instances within each class are independent and uniformly distributed with the same within- and between-class variances. The FOM method working is presented in Figure 1 and the results in Figure 2 indicate that the FOM-SVM produced a decision line equivalent to the standard SVM using fewer SVs.

Tests with real-world benchmarking datasets were also carried out. We used two UCI datasets, Pima Diabetes (PIMA) and Breast Cancer (BC), as well as the vertebral column pathologies dataset described in [9], named henceforth VCP dataset. For this study, we transformed the original three-class VCP problem into a binary one by aggregating the two classes of pathologies, the disc hernia and spondylolisthesis, into a single one. The normal class remained unchanged.

The results for the SVM, OM-SVM and FOM-SVM classifiers are shown in Table 1. For the LS-SVM, OM-LSSVM and FOM-LSSVM classifiers, the results are shown in Table 2. We report performance metrics (mean value and standard deviation of the recognition rate) on the testing set averaged over 20 independent runs. We also show the map grid sizes (Map Sizes), the average number of SVs (SVs #), the reduction of the number of support vectors (SV Red.), the average training time in seconds, as well as the values of the parameters C (SVM), γ (LS-SVM) and the tolerance.

By analyzing these tables, one can easily conclude that, as expected, the accuracies of all the reduced-set classifiers were equivalent to those achieved by the full-set classifiers. Moreover, one can also conclude that the accuracies of FOM-SVM (FOM-LSSVM) classifier was similar to those achieved by the OM-SVM (OM-LSSVM) classifiers, with the advantage of reducing the training times.

6 Conclusion

In this paper, we have proposed an iterative variant of the recently proposed OM algorithm for building reduced-set SVM/LSSVM classifiers. The proposed approach, called fast opposite maps (FOM), consists in successive applications of the original OM algorithm to the resulting reduced dataset, i.e. the reduced dataset achieved by the current run of the OM (say, at iteration j) serves as input to another run of the OM algorithm, and so on. The obtained results indicated that the FOM method performs as well as the original OM approach, but a much faster rate, providing a reduced number of SVs while mantaining equivalent accuracy. Currently, we are evaluation the OM/FOM algorithm on multiclass problems.

References

1. Carvalho, B.P.R., Braga, A.P.: IP-LSSVM: A two-step sparse classifier. Pattern Recognition Letters 30, 1507–1515 (2009)
2. D'Amato, L., Moreno, J.A., Mujica, R.: Reducing the Complexity of Kernel Machines with Neural Growing Gas in Feature Space. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) IBERAMIA 2004. LNCS (LNAI), vol. 3315, pp. 799–808. Springer, Heidelberg (2004)
3. Hoegaerts, L., Suykens, J.A.K., Vandewalle, J., De Moor, B.: A Comparison of Pruning Algorithms for Sparse Least Squares Support Vector Machines. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 1247–1253. Springer, Heidelberg (2004)
4. Hussain, A., Shahbudin, S., Husain, H., Samad, S.A., Tahir, N.M.: Reduced set support vector machines: Application for 2-dimensional datasets. In: Proc. of the Second International Conf. on Signal Processing and Communication Systems (2008)
5. Kohonen, T.K.: Self-Organizing Maps. Springer (1997)
6. Lee, Y.-J., Mangasarian, O.L.: SSVM: A smooth support vector machine for classification. Computational Optimization and Applications 20(1), 5–22 (2001)
7. Li, Y., Lin, C., Zhang, W.: Letters: Improved sparse least-squares support vector machine classifiers. Neurocomputing 69, 1655–1658 (2006)
8. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. MIT Press, Cambridge (1999)
9. Rocha Neto, A.R., Barreto, G.A.: On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. IEEE Latin America Transactions 7(4), 487–496 (2009)
10. Rocha Neto, A.R., Barreto, G.A.: A Novel Heuristic for Building Reduced-Set SVMs Using the Self-Organizing Map. In: Cabestany, J., Rojas, I., Joya, G. (eds.) IWANN 2011, Part I. LNCS, vol. 6691, pp. 97–104. Springer, Heidelberg (2011)
11. Steinwart, I.: Sparseness of support vector machines. Journal of Machine Learning Research 4, 1071–1105 (2003)
12. Suykens, J.A.K., Lukas, L., Vandewalle, J.: Sparse approximation using least squares support vector machines. In: Proceedings of 2000 IEEE International Symposium on Circuits and Systems, Geneva, Switzerland, pp. 757–760 (2000)
13. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters 9(3), 293–300 (1999)
14. Tang, B., Mazzoni, D.: Multiclass reduced-set support vector machines. In: Proc. of the 23rd International Conf. on Machine Learning, pp. 921–928 (2006)

Root Attribute Behavior within a Random Forest

Thais Mayumi Oshiro and José Augusto Baranauskas

Department of Computer Science and Mathematics,
Faculty of Philosophy, Sciences and Languages at Ribeirao Preto,
University of Sao Paulo, Brazil
{thaismayumi,augusto}@usp.br

Abstract. Random Forest is a computationally efficient technique that can operate quickly over large datasets. It has been used in many recent research projects and real-world applications in diverse domains. However, the associated literature provides few information about what happens in the trees within a Random Forest. The research reported here analyzes the frequency that an attribute appears in the root node in a Random Forest in order to find out if it uses all attributes with equal frequency or if there is some of them most used. Additionally, we have also analyzed the estimated out-of-bag error of the trees aiming to check if the most used attributes present a good performance. Furthermore, we have analyzed if the use of pre-pruning could influence the performance of the Random Forest using out-of-bag errors. Our main conclusions are that the frequency of the attributes in the root node has an exponential behavior. In addition, the use of the estimated out-of-bag error can help to find relevant attributes within the forest. Concerning to the use of pre-pruning, it was observed the execution time can be faster, without significant loss of performance.

Keywords: Machine Learning, Random Forest.

1 Introduction

A great interest in the machine learning research concerns ensemble learning — methods that generate many classifiers and combine their results. It is largely accepted the performance of a set of many weak classifiers is usually better than a single classifier given the same quantity of train information [21]. Ensemble methods widely known are boosting [11,23] and bagging [6], and more recently Random Forests [5,16].

In the bagging method (*b*ootstrap *agg*regation), different training subsets are randomly drawn with replacement from the entire training set. Each training subset is fed as input to base learners. All extracted learners are combined using a majority vote. While bagging can generate classifiers in parallel, boosting generates them sequentially.

Random Forests is another ensemble method, which constructs many decision trees that will be used to classify a new instance by the majority vote. Each split

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 733–742, 2012.

of the decision tree uses a subset of attributes randomly selected from the whole original set of attributes. Additionally, each tree uses a different bootstrap sample of the data in the same manner as bagging.

Normally, bagging is more accurate than a single classifier, but it is sometimes much less accurate than boosting. On the other hand, boosting can create ensembles that are less accurate than a single classifier. In some situations, boosting can overfit noisy datasets, thus decreasing its performance. Random Forests, on the other hand, are more robust than boosting with respect to noise; faster than bagging and boosting; its performance is as good as boosting and sometimes better, and they do not overfit [5].

Nowadays, Random Forest is a method of ensemble learning widely used in the literature and applied fields. But the associate literature provides few information about what happens in the trees within the Random Forest. This study tries to provide insights about what happens at the root level of Random Forests. We have analyzed the frequencies of all attributes in trees at root level. In addition, aiming to find out if these attributes were good ones and if there is a best attribute among the top ten, we have used an estimated out-of-bag error as a supplementary metric. This metric enables to differentiate attributes that had the same frequency, and thus, may identify the best attribute used by a tree in the root node.

Biomedical datasets are characterized by fairly few instances and many attributes. Irrelevant attributes not only lead to low performance, but also add extra difficulties in finding potentially useful knowledge [18,20]. Therefore, excluding irrelevant attributes facilitates the data visualization and may improve the classification performance. In addition, identifying a subset or a single best attribute in a biomedical dataset can improve human knowledge.

Even Random Forests are being widely used, we did not find any similar work to this one in the literature and thus, this work can provide new insights that may help future works and researches. Even considering there is no similar work to this one, we preset in the next section some recent researches using Random Forests.

The remaining of this paper is organized as follows. Section 2 describes some related work. Section 3 describes what Random Tree and Random Forest are and how they work. The datasets used in the experiments are described in Section 4 and Section 5 describes the experimental methodology used and the results of the experiments are shown in Section 6. Section 7 presents the conclusions.

2 Related Work

Since Random Forests are efficient, multi-class, and able to handle large attribute space, they have been widely used in several domains such as real-time face recognition [22], bioinformatics [13] and there are also some recent research in medical domain, for instance [15] as well as medical image segmentation [24].

[22] proposes a tracking algorithm using adaptive random forests for real-time face tracking and the approach was equally applicable to tracking any

moving object. [13] presents one of the first illustrations of successfully analyzing genome-wide association (GWA) data with a machine learning algorithm (Random Forests). In [15] they introduce an efficient keyword based medical image retrieval method using image classification with Random Forests classifier and confidence assigning of each keyword. [24] proposes an enhancement of the Random Forests to segment 3D objects in different 3D medical imaging modalities.

3 Random Trees and Random Forests

Assume a training set T with a attributes and n instances and define T_k a bootstrap training set sampled from T with replacement, containing n instances and using m random attributes ($m \leq a$) at each node.

A Random Tree is a tree drawn at random from a set of possible trees, with m random attributes at each node. The term "at random" means that each tree has an equal chance of being sampled. Random Trees can be efficiently generated, and the combination of large sets of Random Trees generally leads to accurate models [25,9].

A Random Forest is defined formally as follows [5]: it is a classifier consisting of a collection of L Random Tree classifiers $\{h_k(\mathbf{x}, T_k)\}$, $k = 1, 2, \ldots, L$, where T_k are independent identically distributed random samples and each tree casts a unit vote for the most popular class at input $\mathbf{x}$.

As already mentioned, Random Forests employ the same method bagging does to produce random samples of training sets (bootstraps samples) for each Random Tree. Each new training set is built, with replacement, from the original training set. Thus, the tree is built using the new subset and a random attribute selection. The best split on the random attributes selected is used to split the node. The trees grown are not pruned.

In a Random Forest, the out-of-bag method works as follows: given a specific training set T, generate bootstrap training sets T_k, construct classifiers $\{h_k(\mathbf{x}, T_k)\}$ and let them vote to create the bagged classifier. For each $(\mathbf{x}, y)$ in the training set, aggregate the votes only over those classifiers for which T_k does not contain $(\mathbf{x}, y)$. This is the out-of-bag classifier. Then the out-of-bag estimate for the generalization error is the error rate of the out-of-bag classifier on the training set [5].

4 Datasets

The experiments reported here used 14 datasets (with 2 variants), all representing real medical data and none of which had missing values for the class attribute. The biomedical domain is of particular interest since it allows one to evaluate Random Forests under real and difficult situations often faced by human experts. A brief description of each dataset is provided.

Lung Cancer, CNS (Central Nervous System Tumour Outcome), *Lymphoma, Ovarian 61902, Leukemia, Leukemia nom.,* and *WDBC* (Wisconsin Diagnostic

Breast Cancer) are all related to cancer and their attributes consist of clinical, laboratory and gene expression data. *Leukemia* and *Leukemia nom.* represent the same data, but the second one had its attributes discretized [17]. *C. Arrhythmia* (*C.* stands for *Cardiac*) is related to heart diseases and its attributes represent clinical and laboratory data. *Allhyper, Allhypo, Sick* and *Thyroid 0387* are a series of datasets related to thyroid conditions. *Dermatology* is related to human conditions. Datasets were obtained from the UCI Repository [10], except *CNS* and *Lymphoma* were obtained from [2]; *Ovarian 61902* was obtained from [3]; *Leukemia* and *Leukemia nom.* were obtained from [1].

5 Experimental Methodology

In a previous work [19], it was analyzed an optimal range for the number of trees within a Random Forest, i.e., a threshold from which increasing the number of trees would bring no significant performance gain, and would only increase the computational cost. We found, based on the performed experiments, that a range between 64 and 128 trees in a forest is the most indicated for accuracy estimation. Under this situation, we have tried to generate forests containing 128 trees but without stability on the experiments reported in the next section. In this case, we assume that a result (10 top attributes subset) is stable when increasing the number of trees there is almost no change in the subset (the 10 top attributes remain basically the same). Denote α_L the subset of the 10 most important attributes obtained from L Random Trees (see in Section 6 how the subsets were obtained). Then, we define attributes subset stability using L Random Trees as $|\alpha_L \cap \alpha_{2L}| \simeq 10$. First, we have used 64 and 128 trees in datasets with many attributes, and it was observed that the subsets of the 10 top attributes varied widely. We have also tried $\frac{a}{2}$ trees, again without stability, where a is the number of attributes in the dataset. Finally, forests containing a and $2a$ trees presented stable results that are reported in this text. The lesson that can be learnt from the previous and the current experiments is that nice accuracy can be reached rapidly with 64–128 trees; this point-of-view sees the random forest as a black-box. However, looking specific factors inside a random forest, i.e. looking the random forest as a white-box, may require more trees.

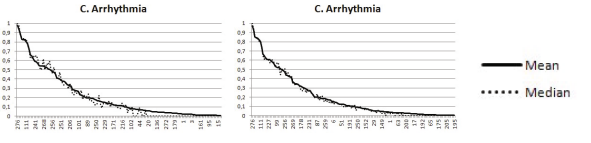
To assess performance, 10-fold cross-validation was performed in the experiments. All experiments refer to the position of the attribute (i.e., the index of the attribute in the dataset according to Weka [14], which starts at zero) as its ID.

In order to analyze if some results were significantly different, we applied the Friedman test [12], considering a significance level of 5%. If the Friedman test rejects the null hypothesis, a *post-hoc* test is necessary to check in which classifier pairs the differences actually are significant [8]. The *post-hoc* test used was the Benjamini-Hochberg [4] and we performed an all versus all comparison, making all possible comparisons among the twelve forests. The tests were performed using the R software for statistical computing (`http://www.r-project.org/`).

6 Experiments, Results and Discussion

Experiment 1. In this experiment we have looked for the average attribute frequency at root level, for instance, if attributes appear uniformly or if there is a subset of them that is most commonly used.

To perform this experiment, two measures were used: the number of times an attribute was among the m random attributes selected by the tree at the root level ($timesSelected$) and the number of times this attribute was, in fact, chosen to be at the root node ($timesRoot$). Then we have used the ratio between them ($Frequency = timesRoot/timesSelected$) to analyze attribute frequency. After sorting frequencies for all attributes in each dataset, it is possible to note they present an exponential behavior. In Figure 1 is shown this exponential behavior using the dataset C. Arrhythmia. The other datasets present similar behaviors, and thus, these figures were omitted. In Figure 2 only the ten highest frequencies are shown in order to facilitate the analysis. There are four lines in each graphic representing the mean and median frequencies for forests using a and $2a$ trees (ordered by the mean frequencies of the forest using a trees). From this figure, it is possible to observe in some cases there is a single attribute that stands out (for instance, Allhyper and Allhypo both containing few attributes), and in other cases there is a subset of attributes most often used (for instance, Lymphoma and Leukemia, both containing a large number of attributes).



(a) Using a trees (b) Using $2a$ trees

Fig. 1. Frequency of all attributes using the dataset C. Arrhythmia

Experiment 2. Now suppose there are three attributes in the best attribute subset, A, B and C. Assume all of them have the same frequency, but attribute A has estimated out-of-bag error equals to 0.90, B equals to 0.65 and C equals to 0.20. In this case, we assume attribute C is the best one in the subset, since its performance is the best. The question that arises is how to estimate the out-of-bag error for a given attribute. The root level attribute represents the most important condition for classes discrimination [7] in the tree, and therefore, one can assume it determines the performance of the tree itself. Under this assumption we have used the out-of-bag error of the tree when attribute α is at its root level as an estimate of performance for attribute α.

With this modifications, we have performed a second experiment, in which frequencies have been altered to $Frequency \times (1 - OOB)$, where OOB corresponds to the mean out-of-bag error of an attribute as explained before. The results of this experiment are shown in Figure 3.

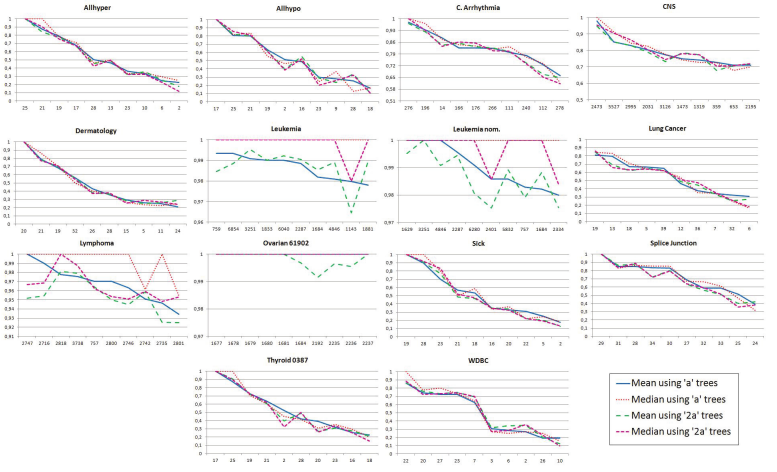


Fig. 2. Frequency of the 10 most used attributes in all datasets. The x-axis corresponds to the number of the attribute and the y-axis corresponds to the frequency. Although all y-axis ranges from 0 to 1, this interval varies in some graphics for better viewing.

Analyzing the results, it can be observed that in all datasets the frequency had an exponential behavior or similar, even in the datasets that showed a linear behavior in the first experiment. Thus, using the estimated out-of-bag error there is mostly only one attribute that stands out in each dataset.

Experiment 3. As mentioned earlier, Random Forests do not overfit, although trees within them are grown without post-pruning. In this experiment we have analyzed the behavior of pre-pruning, since it can speed up Random Forest induction. To perform pre-pruning of the trees of the Random Forest, the parameter $minNum$ was used. It determines the minimum total weight of the instances in a leaf, where the default value in Weka [14] is 1.0, generating very large trees. Based on this, we have used ten different values of $minNum$: 1, 2, 3, 5, 7, 11, 13, 17, 19 and 23. As explained before, for accuracy estimation a range from 64–128 trees are enough [19]. Based on this result, we have built forests with 128 trees in this experiment. To analyze the several values of $minNum$ we have used AUC values and we have applied the Friedman test [12], considering a significance level of 5%. In addition, we have observed the average execution time to induce the forest using each different value of $minNum$. This measure was taken based on the average execution time to induce the forest using $minNum = 1$, i.e, the execution time to induce this forest was taken as 100% and the remaining percentages were calculated based on this one, since for large values of $minNum$ the time is shorter, due to the pre-pruning process that stops the growth of trees.

Table 1 presents the results of the *post-hoc* test after the Friedman's test and the rejection of the null hypothesis, the average rank and the percentage of the average execution time of each value of $minNum$. In this table $\triangle$ ($\blacktriangle$) indicates the Random Forest at the specified row is better (significantly) than the Random Forest at the specified column; $\triangledown$ ($\blacktriangledown$) the Random Forest at the specified row is

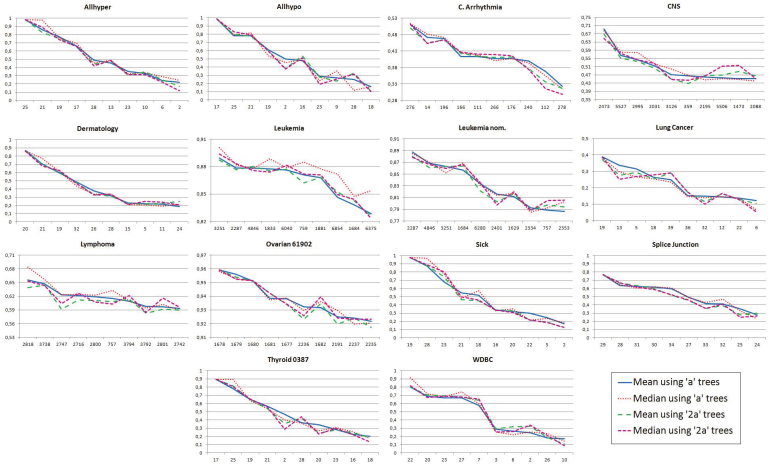


Fig. 3. Frequency of the 10 most used attributes in all datasets using the estimated out-of-bag error

worse (significantly) than the Random Forest at the specified column; ∘ indicates no difference whatsoever. The lower triangle of this table is not shown because it has opposite results to the upper triangle by symmetry.

Table 1. Friedman's test results for AUC values using 128 trees and considering a significance level of 5%; average rank of each value of $minNum$ and the percentage of the average execution time

$minNum$	1	2	3	5	7	11	13	17	19	23
1	∘	▽	▽	▽	△	△	▽	△	△	△
2		∘	▽	▽	△	△	△	△	△	△
3			∘	▽	△	△	△	△	△	△
5				∘	△	△	△	△	△	△
7					∘	△	▽	△	△	△
11						∘	▽	△	△	△
13							∘	△	△	△
17								∘	△	△
19									∘	△
23										∘
Average Rank	5.18	4.64	4.46	4.39	5.46	5.54	4.93	6.25	6.79	7.36
Time(%)	100.00	95.04	92.22	85.49	81.80	76.15	73.64	70.30	69.39	66.91

As can be seen, the execution time decreases as the value of $minNum$ increases, which is expected since a higher value represents a smaller tree, and therefore, a shorter execution time. Although there is no significant differences, it is possible to observe from Table 1 that $minNum = 5$ appears to be an interesting value with the best average rank. Using this value, the second experiment was repeated and the results are shown in Figure 4. As can be seen, there were not significant differences between the frequencies behaviors shown in Figures 3 and 4, but the later is almost 15% faster than the former. However, there were

differences in some subsets of the ten most used attributes. For instance, in four datasets (Leukemia, Lymphoma, Ovarian and WDBC) the ten most used attributes were the same in both experiments, but their sequences were different; in other four datasets (Arrhythmia, CNS, Leukemia nom. and Lung Cancer) there were some attributes that appear in both cases (in the same order and in different order) and there were different attributes between them. On the other hand, there were six datasets (Allhyper, Allhypo, Dermatology, Sick, Splice and Thyroid) where the sequences of the ten most used attributes were the same in both experiments.

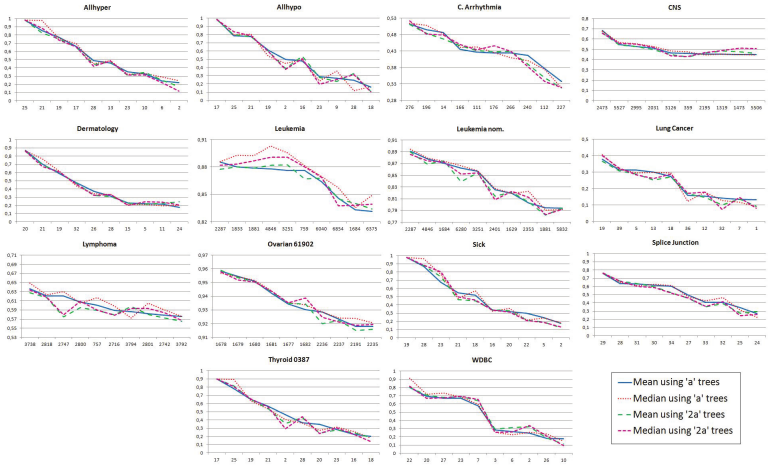


Fig. 4. Frequency of the 10 most used attributes in all datasets using the estimated out-of-bag error and $minNum = 5$

7 Conclusion

This study evaluated 14 medical datasets (with 2 variants) using Random Forests with a and $2a$ trees, where a is the number of attributes in the dataset. Analyzing the results, it can be seen that, in the medical domain, the Random Forest chooses a subset of attributes or a single one in each dataset. In addition, the frequency that the attributes appear in the root node has an exponential behavior. It seems that when we used a and $2a$ trees, the subset of attributes is stable. In addition, we can observe that not always that an attribute is used more than other, its performance is better. Sometimes another attribute had a minor estimated out-of-bag error and when this measure was used this attribute was ahead of the first one. Using the estimated out-of-bag error as a complement, we notice that in all datasets, in general, one attribute stood out. It is noteworthy that in biomedical datasets, finding a subset or a single best attribute can enhance the knowledge discovery and the classification performance. Furthermore, the value of the measure $minNum$ does not seem to significantly affect the results although it decreases the execution time. We used $minNum = 5$ and the results

did not present significant differences. Future work can improve the estimated out-of-bag error, for instance, weighting each attribute by its tree level or even by the number of instances reaching that node.

Acknowledgments. This work was funded by National Research Council of Brazil (CNPq), and the Amazon State Research Foundation (FAPEAM) through the Program National Institutes of Science and Technology, INCT ADAPTA Project (Centre for Studies of Adaptations of Aquatic Biota of the Amazon).

References

1. Cancer program data sets. Broad Institute (2010), http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi
2. Dataset repository in arff (weka). BioInformatics Group Seville (2010), http://www.upo.es/eps/bigs/datasets.html
3. Datasets. Cilab (2010), http://cilab.ujn.edu.cn/datasets.htm
4. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B 57, 289–300 (1995)
5. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
6. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
7. Costa, P.R., Acencio, M.L., Lemke, N.: A machine learning approach for genome-wide prediction of morbid and druggable human genes based on systems-level data. BMC Genomics 11(suppl. 5) (2010)
8. Demšar, J.: Statistical comparison of classifiers over multiple data sets. Journal of Machine Learning Research 7(1), 1–30 (2006)
9. Dubath, P., Rimoldini, L., Süveges, M., Blomme, J., López, M., Sarro, L.M., De Ridder, J., Cuypers, J., Guy, L., Lecoeur, I., Nienartowicz, K., Jan, A., Beck, M., Mowlavi, N., De Cat, P., Lebzelter, T., Eyer, L.: Random forest automated supervised classification of hipparcos periodic variable stars. Monthly Notices of the Royal Astronomical Society 414(3), 2602–2617 (2011), http://dx.doi.org/10.1111/j.1365-2966.2011.18575.x
10. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
11. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 123–140. Morgan Kaufmann, Lake Tahoe (1996)
12. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11(1), 86–92 (1940)
13. Goldstein, B., Hubbard, A., Cutler, A., Barcellos, L.: An application of random forests to a genome-wide association dataset: Methodological considerations and new findings. BMC Genetics 11(1), 49 (2010), http://www.biomedcentral.com/1471-2156/11/49
14. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining Explor. Newsl. 11(1), 10–18 (2009)

15. Lee, J.-H., Kim, D.-Y., Ko, B.C., Nam, J.-Y.: Keyword annotation of medical image with random forest classifier and confidence assigning. In: International Conference on Computer Graphics, Imaging and Visualization, pp. 156–159 (2011)
16. Liaw, A., Wiener, M.: Classification and regression by randomforest. R News 2(3), 1–5 (2002), http://CRAN.R-project.org/doc/Rnews/
17. Netto, O.P., Nozawa, S.R., Mitrowsky, R.A.R., Macedo, A.A., Baranauskas, J.A.:
18. Oh, I.S., Lee, J.S., Moon, B.R.: Hybrid genetic algorithms for feature selection. IEEE Trans. Pattern Anal. Mach. Intell. 26, 1424–1437 (2004)
19. Oshiro, T.M., Perez, P.S., Baranauskas, J.A.: How Many Trees in a Random Forest? In: Perner, P. (ed.) MLDM 2012. LNCS, vol. 7376, pp. 154–168. Springer, Heidelberg (2012)
20. Saeys, Y., Inza, I.n., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23, 2507–2517 (2007)
21. Sirikulviriya, N., Sinthupinyo, S.: Integration of rules from a random forest. In: International Conference on Information and Electronics Engineering, vol. 6, pp. 194–198 (2011)
22. Tang, Y.: Real-Time Automatic Face Tracking Using Adaptive Random Forests. Master's thesis, Department of Electrical and Computer Engineering, McGill University, Montreal, Canada (June 2010)
23. Wang, G., Hao, J., Ma, J., Jiang, H.: A comparative assessment of ensemble learning for credit scoring. Expert Systems with Applications 38, 223–230 (2011)
24. Yaqub, M., Javaid, M.K., Cooper, C., Noble, J.A.: Improving the Classification Accuracy of the Classic RF Method by Intelligent Feature Selection and Weighted Voting of Trees with Application to Medical Image Segmentation. In: Suzuki, K., Wang, F., Shen, D., Yan, P. (eds.) MLMI 2011. LNCS, vol. 7009, pp. 184–192. Springer, Heidelberg (2011)
25. Zhao, Y., Zhang, Y.: Comparison of decision tree methods for finding active objects. Advances in Space Research 41, 1955–1959 (2008)

Fault Detection in Continuous Industrial Chemical Processes: A New Approach Using the Hidden Markov Modeling. Case Study: A Boiler from a Brazilian Cellulose Pulp Mill

Gustavo Matheus de Almeida* and Song Won Park

Dept. of Chemical Engineering and Statistics, CAP, Federal Univ. of Sao Joao del-Rei
Rod. MG 443, Km 07, Faz. do Cadete, 36.420-000, Ouro Branco, MG, Brazil
Dept. of Chemical Engineering, PS, University of Sao Paulo
Av. Prof. Luciano Gualberto, 380/Tv. 3, Butanta, 05.508-9000, Sao Paulo, SP, Brazil
gustavoalmeida@ufsj.edu.br,sonwpark@usp.br

Abstract. The development of automatic and reliable monitoring systems is an open issue in continuous industrial chemical processes. The challenges lay on simultaneously managing multiple normal modes of operation as well as the transitions among them with reasonable false alarm rates, and in reaching early fault detection. This work explores and attests the capacity of the signal processing method called hidden Markov model (HMM) in contributing to overcome these issues. After presenting the motivation for its use in this engineering field, the methodology is introduced and an application is illustrated. Here, the HMM ability of directly learning from process historical data both desired features system dynamics and structure of correlations is shown. Aiming to reach practical insights a real case study based on operations of an industrial boiler is used. A comparison with Principal Components Analysis (PCA) and Self-Organizing Maps (SOM) shows the effectiveness of the proposed HMM-based fault detection system.

Keywords: Process monitoring, Probabilistic learning, Signal processing, Hidden Markov model, Real-world problem, Industrial database.

1 Introduction

The statistical method called hidden Markov model (HMM), an extension of the Markov chain modeling, came up around the seventies, with applications on speech processing. In the ending of the eighties, it became more popular with Rabiner [1] and Rabiner and Juang [2]. Since then, others fields have experienced its potential, namely *e.g.* bioinformatics [3], telecommunications [4] and financial

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 743–752, 2012.

engineering [5]. The present paper discusses the application of HMM in the industrial engineering field, and more specifically in the monitoring of continuous chemical processes. A reliable fault detection system should reach early detection with a minimum generation of false alarms. Continuous processes as found in cellulose mills and petrochemical refineries are characterized by having multiple normal states of operation whose management constitutes a practical challenge in the development of monitoring systems [6,7]. Most studies of application of HMMs in process monitoring employs computer-simulated data [8,9,10,11,12,13]. The present study investigates, in a practical point of view, the use of this method for the development of reliable fault detection systems in continuous chemical industrial processes. For that, a methodology is proposed and a real case study is used for illustration. The initial step verifies the capacity of HMMs to deal with multiple normal operating states, which would minimize the generation of false alarms, and in sequence the early fault detection issue is addressed. To the best of our knownledge, such approach was not contemplated before in the literature.

2 Hidden Markov Models

The hidden Markov Model (HMM) method is used to identify changes of statistical nature in signals over some index *e.g.* time. It is a statistical sequential pattern recognition tool since it works with data sequence as input. HMMs are an extension of Markov models. In addition to the first stochastic process concerning the state-transitions following the Markov property *i.e.* $P(q_t|q_{t-1})$, a second one with respect to the observation-emissions *i.e.* $P(o_t|q_t)$ is present. This means that every state of the Markov chain is potentially able to emit any observation. This explains the term *hidden*, as the underlying Markov chain is no longer directly observable. Given a number of states for the Markov chain (N), a discrete HMM (λ) is specified by (1) the initial state probability distribution (π), used to initiate the Markov chain, and (2) the state-transition probability distribution (A) as in Markov models, besides (3) emission probability distributions (B) (*i.e.* $\lambda = (\pi, A, B)$). In the continuous case (used here), matrix B is replaced by *pdfs* usually given by finite mixture of Gaussians for which it is necessary to specify mixture components (c), mean vectors (μ) and covariance matrix (Σ) [1].

2.1 Fault Detection Using HMMs

Every chemical process operates under random influences. Hence process measurements may be seen as realizations of an underlying stochastic system, and in consequence operating conditions can be described by specific probability distributions [14]. In this context the pattern recognition role played by HMMs enable them to be applied in monitoring applications by detecting changes in the parameters of these distributions over time. Here the signals are composed of measurements of process variables continuously collected. Its input-output relation is as follows: $O \longrightarrow \text{HMM} \longrightarrow (-log[P(O|\lambda)])$. The input, called observation sequence (O), contains T observation vectors (o_t), *i.e.* $O = \{o_1, o_2, ..., o_t, ..., o_T\}$,

which can be discrete or real (used here) valued. The output is a likelihood value ($-log[P(O|\lambda)]$), that measures the capacity of the model (λ) in generating the observed data sequence (O) [1]. The logarithmic form is used to overcome numerical computation problems.

3 Methodology

The input of the proposed methodology is a process historical database collected in mill. Firstly, data pre-processing tasks are used to *e.g.* check missing and anomalous registers. In sequence, a set of key monitoring process variables is chosen. A feature extraction procedure may also be employed. Next step is concerning the identification of a HMM to be used as the monitoring system. The vectors in the observation sequences (O) are composed of measurements of the variable subset previously defined. Since HMM is a data-driven method, this step makes use of training and validation data subsets, for parameter estimation (π, A, B) and for final model selection, respectively. Finally, by using independent data sets, a performance evaluation of the monitoring system is carried out. Each step is detailed along the discussion of the results.

4 Case Study

The case study uses a chemical recovery boiler belonging to a cellulose pulp mill located in Brazil. One of its goals is to produce high pressure steam using a heat transfer section containing a series of heat exchangers, as in power boilers. A critical situation that may happen during the operations concerns the reduction of its thermal efficiency due to the deposition of specific materials over tubes of them. Hence it is continuously monitored. Common practices are still based on operators' actions, namely patrols around the boiler and accompanying of key variables, mainly the fuel gas temperature measures along it [15,16]. To facilitate the results visualization, the case study only uses two of these temperatures as monitoring variables, namely those measured after both heat exchangers the super-heater (T_{SH}, in ^{o}C) and the boiler bank (T_{BB}, in ^{o}C). Under normal conditions they present a high positive correlation and their operating ranges are, respectively, equal to $[453.8-771.1]$ ^{o}C and $[282.0-397.8]$ ^{o}C. The extension to the whole heat transfer section is straightforward. This data set comprehends three months of operation, with a sample interval of five minutes. In addtion to T_{SH} and T_{BB} the liquor (fuel) flow rate (F_{BL}), ranging from 14.0 to 53.0 ton/h, was used for purposes of results interpretation. This independent variable is associated to the multiple normal operating states the boiler can takes on. Other collected variables were used to verify its operating condition at the whole period.

5 Results and Discussion

Firstly, it is justified the use of the hidden Markov modeling for describing chemical processes with continuous operations. With a sequential data series in hand,

such modeling requires a set of conditions, namely (1) a mixture distribution for the marginal distribution of the data, *i.e.* collapsed over time, (2) a serial dependence of the data over some index and (3) an underlying system, given by a series of states it assumes, not entirely clear (*i.e.* hidden). This last condition is what distinguishes HMMs from Markov models. Using the T_{SH} and T_{BB} data, Fig. 1 illustrates the presence of all these characteristics. The classical/widely employed (also used here) (1) Forward algorithm (alg.), for computing the probability of a sequence of observed events [1], (2) Viterbi alg., for finding the most likely hidden state sequence [17], and (3) Baum-Welch alg., for parameter estimation were not described here, due to the limited space [18]. The MATLAB software (v. 7.8.0.347, The MathWorks Inc., 2009) was used in all steps.

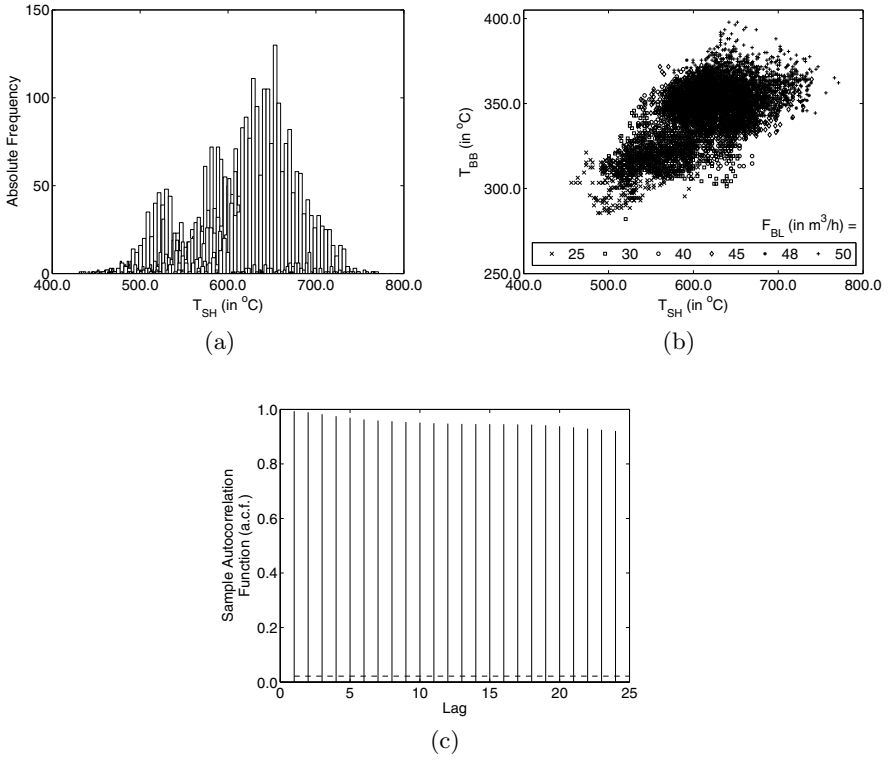


Fig. 1. Conditions for using the hidden Markov modeling: (a) mixture distribution, conditioned to (the independent variable) F_{BL}, (b) serial dependence of T_{BB} due to a high sample frequency, with 95% confidence limits (dotted lines) and (c) system states, determined by F_{BL}, not completely clearly

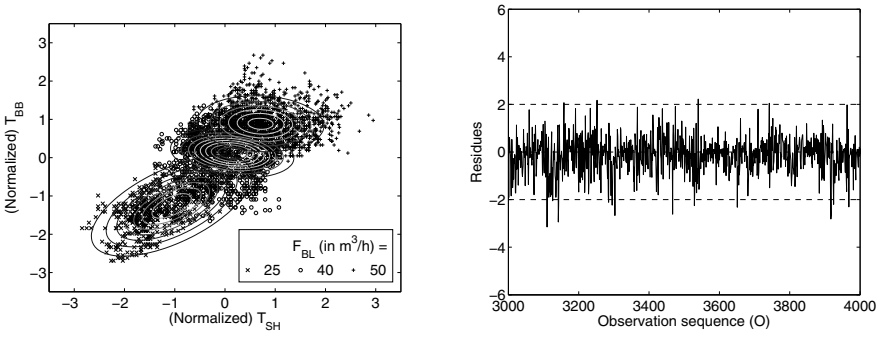
Step 1 - Data Pre-processing: One of the three available months of operation was used in the identification step, for construction of a monitoring system characteristic of normal operations, and the other two in the model evaluation step. To the former a series of filters were applied over the raw mill data. They

regard the range of the variables, the operating conditions of the equipment, the relationship among variables, and the occurrence of missing data.

Step 2 - Variable Selection: From the monitoring variables, T_{SH} and T_{BB}, the structure of the input observation sequences (O) is as follows: $O = \{[T_{SH}\ T_{BB}]'_{t=1}, [T_{SH}\ T_{BB}]'_{t=2}, ..., [T_{SH}\ T_{BB}]'_{t=T}\}$, where T varied from 13 to 447. The definition of its size should take into account the process dynamics being long enough to capture it. Thus, each observation sequence covers at least one hour of operation, which is equivalent to $T = 12$, once the sample time is five minutes. The liquor (fuel) flow rate (F_{BL}) variable is not used as input information to the HMM-based monitoring system, however its role here is to justifies the use of the hidden Markov modeling (as illustrated before) and to support the validation of the results.

Step 3 - HMM Identification: Firstly, the resulting pre-processed data set was divided into training and validation subsets. The former was used to generate a set of candidate models by varying N from 2 up to 6 and c from 1 up to 3, and the latter to select one of them to be used as the monitoring system. The k-means clustering algorithm was used for initialization of the mixture distribution parameters, where $k = N \times c$. The full covariance matrix was used. The model topology was fixed to the ergodic one. The basis for this choice was the characteristic of the physical system in presenting a set of multiple normal operating conditions in conjunction with frequent transitions among them. To the model (λ) selection, a *10*-fold cross-validation procedure was carried out using an amount of 82 observation sequences (O) with varied size, as a function of the F_{BL} values. The selection criterion was based on the standard deviation of the residuals, which are a function of the output likelihood ($-log[P(O|\lambda)]$) values calculated onto the validation data. (Next paragraph explains the reason for using residuals data instead of likelihood values.) The standard deviation measure is directly related to the concept of control chart employed to monitor the operations. The selected model out of 15, to be used as the monitoring system, has $N = 3$ and $c = 1$. The resulting final estimation of the parameters ($\lambda = (\pi, A, B)$), with training and validation data subsets of sizes 82 and 28, respectively, are as follows: $\pi = [0.31\ \ 0.34\ \ 0.35]$, $A = [0.97\ \ 0.00\ \ 0.03; 0.01\ \ 0.99\ \ 0.00; 0.03\ \ 0.00\ \ 0.97]$, $\mu = [-0.04\ \ -1.21\ \ 0.87; 0.20\ \ -1.44\ \ 0.79]$, $\Sigma_1 = [0.25\ \ -0.07; -0.07\ \ 0.19]$, $\Sigma_2 = [0.49\ \ 0.20; 0.20\ \ 0.28]$, $\Sigma_3 = [0.33\ \ -0.04; -0.04\ \ 0.21]$, and $c = [1\ \ 1\ \ 1]$. From Fig. 2(a) the overlapping of the joined training and validation data by the Gaussian distributions of the Markov chain states can be verified. Here, the specialization of the states of the model in describing particular conditions in the boiler, as desired, can also be observed. These results contribute to the model validation in practice.

In order to achieve early fault detection the observation sequences are constructed using a sliding temporal window. As a consequence an autocorrelation arises over the likelihood values calculated along time. To overcome this issue an autoregressive (AR) model of order 2, *i.e.* $y_t = \theta_0 + \theta_1 \times y_{t-1} + \theta_2 \times y_{t-2} + \xi$ (after trying 1^{st}-order), with $\hat{\theta} = [-1.97, -0.61, 1.45]$, where y assumes $-log[P(O|\lambda)]$, was used to obtain independent residuals, *i.e.* $\xi\ (= y_t - \hat{y}_t)$ [20]. Thus the boilers'



(a) (b)

Fig. 2. (a) Overlapping of the full operating range, given T_{SH} and T_{BB}, by the estimated Gaussian distributions (one for each Markov chain state) and association of the model states with boilers' operating conditions: 25, 40 and 50 m^3/h, and (b) Monitoring result for observation sequences associated to particular normal states of operation.

operations are accompanied by a control chart for individual measurements of the residuals data [19], with $\pm 3s$ (lower and upper control limits) $= \pm 2.0$.

Step 4 - Monitoring System Evaluation: Every five minutes (the sample time) an observation sequence (O) of size 5 is fed to the fault detection system (λ) leading to a real time monitoring. Other two months of operation characterizing independent data sets are used. Firstly, it is verified the ability of HMM-based monitoring systems to handle multiple normal modes of operation, which concurs to lower false alarm rates, and in sequence the early fault detection issue.

Step 4.1 - Multiple Normal Operating States: To verify the monitoring system capacity in dealing with the several multiple normal operating states the boiler assumes, 7382 observation sequences are fed to it. They are composed of temperature measurements collected in the first considered month of operation, and associated to particular normal conditions. As an example, for sequences composed of measurements collected during liquor flow rates equal to 25 m^3/h, $O = \{[T_{SH}\ T_{BB}]'_{t=1;F_{BL}=25m^3/h}, ..., [T_{SH}\ T_{BB}]'_{t=T=5;F_{BL}=25m^3/h)}\}$. From the control chart in Fig. 2(b), it can be observed that most residue values lay within the normal region.

Now, 965 observation sequences, composed of measurements collected during transitions among normal modes of operations at the same first month before, were considered. One example is given by $O = \{[T_{SH}\ T_{BB}]'_{t=1;F_{BL}=25m^3/h)}, \cdots, [T_{SH}T_{BB}]'_{t=T=5;F_{BL}=28m^3/h)}\}$, composed of observation vectors associated to $F_{BL} = 25$ and 28 m^3/h. Load variations are very common in chemical industries, and in general they are due to process upsets or equipment malfunctioning. From these sequences, it was possible to verify the capacity of HMMs to manage transitions among normal modes of operation. Most residue values also lay within the normal region, and the resulting control chart (not shown) is similar to that in Fig. 2(b). It occurs because a set of states (of the Markov chain) with or without significantly distinct parameter distribution (*i.e.* μ and/or Σ) values depending

on the case is responsible for describing such observation sequences. One example is the state sequence that describes $O = \{[T_{SH}\ T_{BB}]'_{t=1;F_{BL}=40m^3/h)}, ..., [T_{SH}\ T_{BB}]'_{t=T=5;F_{BL}=45m^3/h)}\}$, given by {1 1 1 2 2}. It is due to the fact that the observation vectors, *i.e.* $[T_{SH}\ T_{BB}]'_t$, are individually modeled by the state associated to the underlying normal condition, given here by F_{BL}. Analogous results were obtained for the other transitions.

Table 1 summarizes the numerical results of this section. The results obtained with the HMM signal processing method were compared with well-established multivariate statistical process control (MSPC) and computational inteligence techniques, namely, respectively, Principal Components Analysis (PCA) [21] and Self-Organizing Maps (SOM) [22]. To the former, a dynamic (D-PCA) approach was employed. The dynamics is taken into account by augmenting each sample observation with previous l (= 5) ones, where l is the number of lags [23,24]. The identified D-PCA monitoring system, derived from the correlation matrix, uses 2 principal components out of 10 (the number of elements in the observation sequences of size 5), explaining 95.0% of the total variance. To the latter field, it was used the global SOM-based novelty detection method[1] [25]. The training data subset was the same used for identification of the HMM model, of size 82. For the comparison of the sensitivities of the statistics in a fair basis, *i.e.* $-log[P(O|\lambda)]$ for HMM, T^2 for D-PCA, and E (Quantization Errors) for SOM, the thresholds of the last two techniques were adjusted to give the same false alarm rate (*i.e.* type I error) of the HMM system, equal to 4.5% (first row of the table). The same validation data subset used before for HMM identification, of size 28, was employed. (Regarding D-PCA, the T^2 statistics gives a better result than the Q statistics, and then the latter is not shown.)

Now, to compare them, the previous data sets of sizes 7382 and 965 were used. To the first (second row of the table), it can be verified that the performance of D-PCA and SOM is slightly better than that of HMM. On the other hand (third row of the table), the HMM modeling, with a false alarm rate of 8.3%, gives a better result than D-PCA and SOM. Differently of the other techniques mainly for D-PCA, it can also be noted for the HMM approach the promixity of the rates when comparing observation sequences collected during both particular (9.1%) and transition (8.3%) periods. As stated before, this may be due to the fact of HMM makes use of a set of states of the Markov chain to individually describe the distinct (determined by different normal modes of operation) observation vectors ($[T_{SH}T_{BB}]'_t$) in an observation sequence (O). This result suggests the ability of HMMs, in a more stable way, to manage multiple normal modes of operations, an inherent and challenging characteristic of continuous industrial chemical processes contributing to lower false alarm rates.

Step 4.2 - Early Fault Detection: After verifying the capacity of HMMs in dealing with multiple normal modes of operation, the early fault detection issue was addressed. The second month of continuous operations presenting a real

[1] Parameter values: zero mean/unit variance, 6×6 rectangular topology, sequential training, 100 training epochs, initial learning rate (η): 0.8, exponential decay learning rate function, Gaussian weighting function, and initial/final radius (σ): 3/0.

Table 1. Comparison of false alarm rates (in %) between HMM- and SOM- and D-PCA- based monitoring systems

Observation sequences (O)	**D-PCA**	**SOM**	**HMM**
In the validation data subset *(reference basis)*	4.5	4.5	4.5
Not associated to transitions ($n = 7382$)	8.8	8.1	9.1
Associated to transitions ($n = 965$)	15.1	11.6	8.3

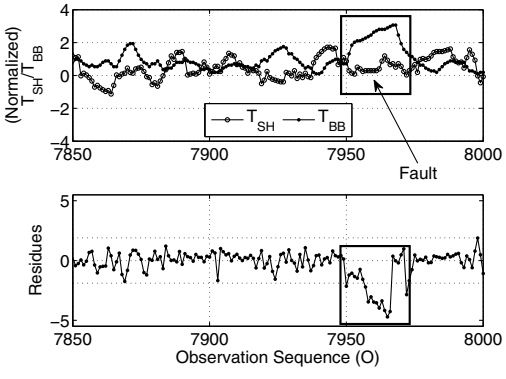


Fig. 3. Early fault detection (below) and change occurred in the relationship between T_{SH} and T_{BB} (above).

fault in the boiler was used. Fig. 3 (below) shows the progressive deviation of the residue values from the normal operating window. The detection is due to the capacity of HMMs in recognizing changes of statistical nature in signals over time. They are caused either by shifts of process variables from their usual operating ranges or by changes in the relationship among them. Both features are present here, as shown in Fig. 3 (above). Whereas T_{BB} assumes greater values, T_{SH} remains nearly constant. Under normal conditions, a high positive correlation between them is expected. The detection occurred in the initial stage of the fault when its perception by control room operators is not an easy task. By dealing with real data, the start and end times of the fault is not exactly known. In addition, the stability of the residue values (between $\pm 3s$) during the stable operations before and after the abnormal event can also be observed. This behaviour was verified for the whole period. Therefore, in addition to the system dynamics (explored in *Step 4.1*) HMM also captured the structure of

correlations in the monitoring variables. These both features are very welcomed for attaining a reliable monitoring of continuous industrial chemical processes.

6 Conclusions

After justifying, this work proposes a methodology and demonstrates the use of the hidden Markov modeling in industrial chemical processes of continuous operations. In order to reach practical insights, a real case study using an industrial boiler was employed. (A prior work using simulated data was conducted by the same authors [8].) It was demonstrated the capacity of the hidden Markov modeling to handle the inherent presence of multiple normal modes of operation and the transitions among them, which in turn minimizes the false alarm rates, as well as to achieve early fault detection. Attending to these issues contributes to the development of automatic and reliable fault detection systems, a current open question in practice. To the best of our knowledge, such approach was not contemplated before in the literature. In addition, its explicit modeling based on the probability theory allows an association of its parameters with the physical system, as it was also illustrated. This possibility is rich in both directions to get insights about the real system learned by the model as also to help it in learning the physical phenomena more efficiently. Moreover, its data-driven modeling overcomes the complex mathematical description of industrial chemical processes. Here, for purposes of clarity, a reduced set of monitoring variables was employed, however the extension for a high dimensional subset is straightforward. Also, we believe that the results we have observed hold generally for different but similar systems. Concluding, HMM-based monitoring systems can play a useful role in helping control room operators and engineers to accomplish monitoring tasks in continuous industrial processes. As cited by Patton [26], "signal processing tools (as is HMM) constitutes an alternative to the development of FDI (Fault Detection and Isolation) systems".

Acknowledgments. The authors thank the financial support of CNPq (under grant 482455/2010-1) and FAPEMIG (under grant APQ-02544-09) Brazilian research foundations and the mill for the cession of the process historical data.

References

1. Rabiner, L.: A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. IEEE 77(2), 257–286 (1989)
2. Rabiner, L., Juang, B.: An Introduction to Hidden Markov Models. IEEE ASSP Mag. 3(1), 4–16 (1986)
3. Kosky, T.: Hidden Markov Models of Bioinformatics. Springer, New York (2002)
4. Cappe, O., Moulines, E.: Inference in Hidden Markov Models. Springer, New York (2010)
5. Mamom, R.S., Elliott, R.J.: Hidden Markov Models in Finance. Springer, New York (2007)

6. Chiang, L.H., Russell, E.L., Braatz, R.D.: Fault Detection and Diagnosis in Industrial Systems. Springer, London (2001)
7. Wang, X.Z.: Data Mining and Knowledge Discovery for Process Monitoring and Control. Springer, London (1999)
8. Almeida, G.M., Reis, M.P.S., Park, S.W.: A Signal Processing Approach for Fault Detection Problem: Application to the DAMADICS Actuator Benchmark Problem. To appear in: 22nd European Symposium on Computer Aided Process Engineering (ESCAPE), IChemE Press, London (2012)
9. Chen, J., Chang, W.: Applying Wavelet-Based Hidden Markov Tree to Enhancing Performance of Process Monitoring. Chem. Eng. Sc. 60(18), 5129–5143 (2005)
10. Sun, W., Palazoglu, A., Romagnoli, J.A.: Detecting Abnormal Process Trends by Wavelet-Domain Hidden Markov Models. AIChE Journal 49(1), 140–150 (2003)
11. Wong, J.C., McDonald, K.A., Palazoglu, A.: Classification of Abnormal Plant Operation Using Multiple Process Variable Trends. J. Proc. Control 11(4), 409–418 (2001)
12. Bakhtazad, A., Palazoglu, A., Romagnoli, J.A.: Detection and Classification of Abnormal Process Situations Using Multidimensional Wavelet Domain Hidden Markov Trees. Comp. and Chem. Eng. 24(2-7), 769–775 (2000)
13. Wong, J.C., McDonald, K.A., Palazoglu, A.: Classification of Process Trends Based on Fuzzified Symbolic Representation and Hidden Markov Models. J. Proc. Control 8(5-6), 395–408 (1998)
14. Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K.: A Review of Process Fault Detection and Diagnosis - Part III: Process History Based Methods. Comp. Chem. Eng. 27(3), 327–346 (2003)
15. Vakkilainen, E.K.: Kraft Recovery Boilers - Principles and Practice. Valopaino Oy., Helsink (2005)
16. Adams, T.N., Frederick, W.J., Grace, T.M., Hupa, M., Iisa, K., Jones, A.K., Tran, H.: Kraft Recovery Boilers. Tappi Press, Atlanta (1997)
17. Viterbi, A.J.: Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Trans. Inform. Theory IT-13, 260–269 (1967)
18. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. Ann. Math. Stat. 41(1), 164–171 (1970)
19. Montgomery, D.C.: Introduction to Statistical Quality Control, 6th edn. Wiley, New York (2008)
20. Aguirre, L.A.: Introducao a Identificacao de Sistemas, 3a edn. UFMG, Belo Horizonte (2007)
21. Kresta, J.V., Macgregor, J.F., Marlin, T.E.: Multivariate Statistical Monitoring of Process Operating Performance. Can. J. Chem. Eng. 69(1), 35–47 (1991)
22. Kohonen, T.: Self-Organizing Maps. Springer, Helsink (1995)
23. Russel, E.L., Chiang, L.H., Braatz, R.D.: Fault Detection in Industrial Processes using Canonical Variate Analysis and Dynamic Principal Component Analysis. Chemometr. Intell. Lab. Syst. 51, 81–93 (2000)
24. Ku, W., Storer, R.H., Georgakis, C.: Disturbance Detection and Isolation by Dynamic Principal Component Analysis. Chemometr. Intell. Lab. Syst. 30(1), 179–196 (1995)
25. Barreto, G.A., Mota, J.C.M., Souza, L.G.M., Frota, R.A., Aguayo, L.: Condition Monitoring of 3G Cellular Networks Through Competitive Neural Models. IEEE Trans. Neural Netw. 16(5), 1064–1075 (2005)
26. Patton, R.J.: Preface. Control Eng. Prac. 14(6), 575–576 (2006)

Modified Reducts and Their Processing for Nearest Neighbor Classification

Naohiro Ishii[1], Ippei Torii[1], Yongguang Bao[2], and Hidekazu Tanaka[3]

[1]Aichi Institute of Technology, Yakusacho, Toyota, Japan 470-0392
[2] Aichi Information System, Kariya, Japan 448-0852
[3] Daido University, Minamiku, Nagoya, Japan 457-8530
{ishii,mac}@aitech.aitech.ac.jp, Baoyg|860@hotmail.com,
hitanaka@daido-it.ac.jp

Abstract. Dimensional reduction of data is still important as in the data processing and on the web to represent and manipulate higher dimensional data. Rough set concept developed is fundamental and useful to process higher dimensional data. Reduct in the rough set is a minimal subset of features, which has almost the same discernible power as the entire features in the higher dimensional scheme. But, we have problems of the application of reducts for the classification. Here, we develop a method which connects reducts and the nearest neighbor method to classify data with higher accuracy. To improve the classification ability of reducts, we propose a new modified reduct based on reducts and its optimization method for the classification with higher accuracy. Then, it is shown that the modified reduct improves the classification accuracy, which is followed by the optimized nearest neighbor classification.

1 Introduction

Rough sets theory firstly introduced by Pawlak[1,2] provides us a new approach to perform data analysis, practically. Up to now, rough set has been applied successfully and widely in machine learning and data mining. The need to manipulate higher dimensional data on the web and to support or process them gives rise to the question of how to represent the data in a lower-dimensional space to allow more space and time efficient computation. Thus, dimensional reduction of data still remains as an important problem. An important task in rough set based data analysis is computation of the attributes or feature reducts for the classification. By Pawlak's rough set theory[1,2], 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]. Here, we discuss reducts for reduction of features dimension and its application to the nearest neighbor classification Then, it is not necessarily efficient to apply reducts directly to nearest neighbor classification method. We propose a

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 753–762, 2012.

modified reduct, which are added redundant attributes to reducts for nearest neighbor classification. Then, to improve the accuracy of the classification by reducts, a weighted nearest neighbor algorithm is introduced. It is shown that the modified reduct improves the classification accuracy, which is followed by the optimized nearest neighbor classification.

2 Classification and Decision Table

The proposed classification consists of some reducts followed by respective Nearest Neighbor(NN) system[6,7].

Table 1. An example of the decision table

	a	b	c	d	class
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	2
x_5	2	1	0	1	2
x_6	2	1	1	0	1
x_7	2	1	2	1	2

A knowledge representing system containing the set of attributes, called condition attributes and the set of decision attributes, is called a decision table. The decision table is useful for classification. Here, a simple example of the decision table is shown in Table 1.

The left side data in the column in Table 1 as shown in, $\{x_1, x_2, x_3, .., x_7\}$ is a set of instances, while the data { a, b, c, d} on the upper row, shows the set of attributes of the instance.

3 Discernibility Matrix

Skowron(1991) proposed to represent decision table in the form of a discernibility matrix[3,4]. This representation has many advantages, in particular it enables simple computation of the core, reducts other concepts[1,2,3]. The discernibility matrix is defined as follows. Let $T = \{U, A, C, D\}$ be a decision table, with $U = \{x_1, x_2, ...x_n\}$. By a discernibility matrix of T, denoted by $M(T)$, which is $n \times n$ matrix defined as

Table 2. Dicernibility matrix of the decision table in Table 1

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	—					
x_3	b,c,d	b,c				
x_4	b	b,d	—			
x_5	a,b,c,d	a,b,c	—	—		
x_6	—	—	a,b,c	a,b,c,d	—	
x_7	a,b	a,b,d	—	—	c,d	c,d

$$m_{ij} = \{a \in C : a(x_i) \neq a(x_j) \wedge (d \in D, d(x_i) \neq d(x_j))\}\ i, j = 1, 2, ...n \tag{1}$$

, where U is the universe of discourse, C is a set of features,

A is a subset of C called condition attributes, and D is a set of decision ones. In Table 2, the dicernibility matrix of the decision table in Table 1 is shown.

The dicernibility function is represented by taking the combination of the disjunction expression of the discerniblity matrix. In Table 2, the item (b,c,d) in the second row and the first column, implies $b \vee c \vee d$ in the Boolean expression, which shows the attribute b, c and d appear together for the discrimination between instances x_1 and x_3 [4]. Finally, these items are shown in conjunction expression. The reducts are derived from the dicernibility function $f_A(D)$ as follows,

$$\begin{aligned} f_A(D) = &(b \vee c \vee d) \wedge b \wedge (a \vee b \vee c \vee d) \wedge (a \vee b) \\ &\wedge (b \vee c) \wedge (b \vee d) \wedge (a \vee b \vee c) \wedge (a \vee b \vee d) \\ &\wedge (a \vee b \vee c) \wedge (a \vee b \vee c \vee d) \wedge (c \vee d) \wedge (c \vee d) \end{aligned} \tag{2}$$

The dicernibility function is reduced by Boolean formula to

$$f_A(D) = (b \wedge c) \vee (b \wedge d) \tag{3}$$

From the above equation $f_A(D)$, two reducts are obtained as

$$\{b, c\} \quad \text{and} \quad \{b,d\} \tag{4}$$

Thus, the core of two reducts, becomes {b}.

4 Modified Reducts – Nearest Neighbor Classification

The reduct in Rough set, is a reduced set of attributes in information system. Then, the classification is expected easily and simply. The nearest neighbor algorithm[5] is a supervised learning algorithm that simply retains the entire training set during learning. During execution, a new input data is composed is compared to each

instance in the training set. The class of the instance that is most similar to the new data(using some distance function) is used as the predicted class. The nearest neighbor quickly(O(n)) classifies a training set of n instances[6,7].

Table 3. Reduct {b,c}

	b	c	class
x_1	0	2	1
x_2	0	2	1
x_3	2	0	2
x_4	2	2	2
x_5	1	0	2
x_6	1	1	1
x_7	1	2	2

Table 4. Reduct {b,d}

	b	d	class
x_1	0	1	1
x_2	0	0	1
x_3	2	0	2
x_4	2	1	2
x_5	1	1	2
x_6	1	0	1
x_7	1	1	2

It is guaranteed to learn a consistent training set(i.e. one in which there are no instances with the same input data(vector) and different outputs) and will not get stuck in local minima. Here, we have problems whether the nearest neighbor algorithm is applicable to the reduct in rough set theory. Experiments to check the nearest neighbor algorithm are performed in the following by using decision table in Table 1. Data by reduct {b,c} is shown in Table 3, while data by reduct {b,d} is shown in Table 4 in the following.

To the data in Table 3, nearest neighbor algorithm is applied. The cross validation method is applied to the data. Since the data (0,2) in x_1 and x_2 in Table 3 is the same, these two data become to one data (0,2),which is represented by x_1 .We assume here that data x_1 is unknown in the class. Then, data x_1 is most close to the data x_7 of the class 2 with the Euclidean distance, 1. Thus x_1 belongs to the class 2. But, this is contradiction, since the data was in the class 1. Next, we assume that data x_3 is unknown in the class. Since the data x_3 is closest to the data x_5 of the class 2 with the distance, 1, x_3 belongs to the class 2. This is correct. Third, we assume that x_5 is unknown in the class. Since x_5 is most close to data x_3 of the class 2 and data x_6 of the class 1 with distance 1, respectively, x_5 will belong to the class 1 and 2 with 50%. Similarly, the accuracy for classification by reduct {b,d} in Table 4 is computed. The accuracy becomes 0.58, that is 58 %.

4.1 Generation of Modified Reducts Based on Paulak's Reducts

Combination of reducts is proposed for the improvement of classification ability. Simultaneous occurrence of attributes is computed from reducts in the dicernibility matrix. Based on the attribute b in reducts {b,c} and {b,d} derived from the discernibility matrix in Table 2, the simultaneous occurrence attributes are counted in the order.

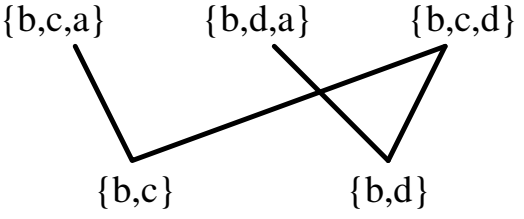


Fig. 1. Generation of modified reducts

In Fig. 1, the modified reducts {b,c,a}, {b,d,a} and {b,c,d} are created on the discernibility matrix, which are occurred with reducts {b,c} and {b,d} and one more attribute is added. In Fig. 1, the added variable occur more than 3 times with reducts {b,c} and {b,d}. From the point of Pawlak`s reducts, these modified reducts include redundant attributes.

As an example, values and the class of the modified reduct {b,d,a} is shown on the three dimensional coordinates as shown in Fig. 2 . Fig. 2 shows the instances in the triangle class(class 1 in Table 1) and those in the circle class(class 2 in Table 1) is classified separately in the nearest neighbor system. Then, the modified reduct{b,d,a} –NN classifies instances correctly.

Since the reduct {b,d}-NN is difficult to classify instances correctly, the classification accuracy bcomes 50%. The modified reduct { b,d,a} classifies correctly with accuracy 100%. This is because the values of the added attribute {a} is different between classes, i.e., the value of the 1 class takes 0 or 1, while that of the 2 class takes 2. Thus, the values of the added attribute are useful to be different between classes for the classification.

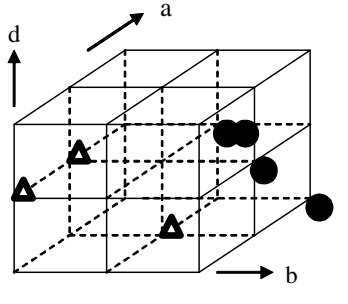


Fig. 2. Representation of modified reduct {b,d,a}

Similarly, the classification accuracy by the modified reduct {b,c,d} becomes (4/7)=0.66, i.e., 66%, while that by the modified reduct {b,c,a} becomes (5/7)=0.71,i.e, 71%. These accuracies 66% and 71% is improved in the classification than the two reducts {b,c} and {b,d} accuracies 50% and 58%, respectively.

4.2 Characterization of Modified Reducts-NN

The modified reducts-NN are characterized from the basic structure of Pawlak's reducts-NN as follows for the classification.

4.2.1 Based on Incorrectly Classified Instance, x_i, in the Pawlak`s Reduct

By adding a new attribute(variable), v, to the Pawlak's reduct, a modified reduct is created. Assume that an incorrectly classified instance, x_i is given. Then, a set of instances, $\{s_i\}$, are created such that the value of the added attribute is same as the value of v of the given instance. x_i . Since the x_i is incorrectly classified in the followed nearest neighbor(NN), there is a nearest neighbor instance, y_i ,which is in different class against x_i . Here, a set in the different class against x_i is assumed to be $\{t_i\}$, each member of which has the same value of the v . Then, we can creat a new set by $\{s_i\}-\{t_i\}$. Next, a nearest neighbor instance , z , of the given instance x_i , is searched, which belongs to the same class of x_i . The instance, z , will be in $\{s_i\}-\{t_i\}$ or not. Finally, we conpute the distances between x_i and member of $\{s_i\}-\{t_i\}$. If the instance, z , is nearest to x_i , then the instance x_i is correctly classified in modified reduct-NN. This case improves the classification accuracy.

When the distance between the given instance x_i and z in the same class, is equal to the distance between the x_i and one element $st \in \{s_i\}-\{t_i\}$, which is different in the class, the x_i belong to the either class with probability 1/2. This implies the x_i escapes from the incorrectly classification in the modified reduct. When such element st exists more than two, the given x_i is also incorrectly classified in the modified reduct.

4.2.2 Based on Correctly Classified Instance, x_i, in the Pawlak`s Reduct

When the value of the added attribute of the given instance x_i is same as the value of the nearest neighbor instance in the Pawlak's reduct, then the instance x_i is also classified correctly in the modified reduct.

4.3 Dependency Relation of Reducts and Modified Ones

To make clear the dependency between the modified reduct and its related reducts, the notion of the dependency of knowledge is applied here[1,2]. In Fig.2, it is expected to show how much the modified reduct {b,c,d} is related to reduct {b,c} and {b,d}. By using Pawlak's notation, reduct {b,c} –NN classification is represented by the relation as

$$U / R_{\{b,c\}} = \{\{x_3, x_4\}, \{x_1, x_2, x_6\}, \{x_5, x_7\}\} \tag{5}$$

$$U / R_{\{b,d\}} = \{\{x_3, x_4\}, \{x_1, x_3, x_6\}, \{x_5, x_7\}\} \tag{6}$$

$$U / Mod.R_{\{b,c,d\}} = \{\{x_2, x_4\}, \{x_6\}, \{x_1, x_3, x_5, x_7\}\} \tag{7}$$

, where U in equation (5) is the set of all instances, $R_{\{b,c\}}$ is the reduct {b,c} followed by nearest neighbor relation. Then $U / R_{\{b,c\}}$ becomes the correctly classified set, $\{x_3, x_4\}$, incorrectly classified set, $\{x_1, x_2, x_6\}$ and the undecided set, $\{x_5, x_7\}$. Similarly, the equation (6) holds in reduct {b,d} and the equation (7) in modified reduct {b,c,d}. The modified reduct relation, $Mod.R_{\{b,c,d\}}$ depends in a degree k $(0 \le k \le 1)$ from the reduct relation $R_{\{b,c\}}$ as follows,

$$k_1 = \frac{card\ \{POS_{R_{\{b,c\}}}(Mod.R_{\{b,c,d\}})\}}{card\ \{U\}} \tag{8}$$

, where $card\{\ \}$ denotes cardinality of the set. Equation (8) is a numerical value.

Since $POS_{R_{\{b,c\}}}(Mod.R_{\{b,c,d\}}) = \{x_5, x_7\}$ holds, $k_1 = 2/7$. Similarly, $POS_{R_{\{b,d\}}}(Mod.R_{\{b,c,d\}}) = \{x_2, x_4, x_5, x_7\}$ holds, $k_2 = 4/7$. This shows the reduct {b,d} effects largely than the reduct {b,c} to the modified reduct {b,c,d}. Similar dependency discussions are carried out on the modified reducts {b,c,a} and {b,d,a}.

5 Weighted Nearest Neighbor Algorithm

To improve the accuracy of the classification by reducts, a weighted nearest neighbor algorithm is introduced. The weight $\{\omega_i\}$ for the attribute is introduced in the distance measure as shown in the following.

For the training instances, the optimized weighted distances are computed as follows: the distance between training instances is defined by

$$Dist(x_r, y_r) = \sqrt{\sum_{i=1}^{n} \omega_i (x_{ri} - y_{ri})^2} \tag{9}$$

, where the suffix r of instances shows the r-th class in the decision table. In the case of the instances x_r and y_r to be in the same r-th class, The weight set $\{\omega_i\}$ is determined so as to minimize the classification errors. Since Euclidean distance values are changed, the classification accuracy is changed. In the case of the modified reduct {b,c,d}, weights ω_b =0.5, ω_c =0.8, and ω_d =0.4 are set. Then, the accuracy of the modified reduct {b,c,d} with weighting becomes 0.86, i.e., 86.4%, as shown in the left side black bars in Fig.3. while the gray bar shows the accuracy of modified

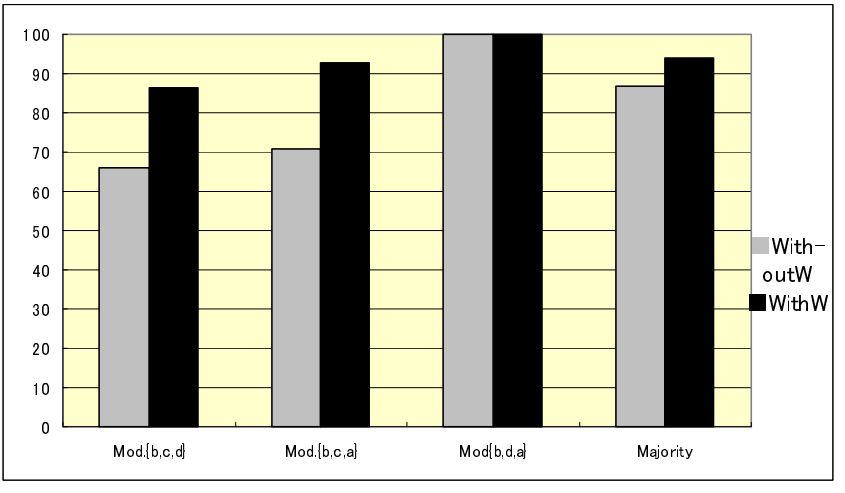


Fig. 3. Improved accuracy by modified reducts

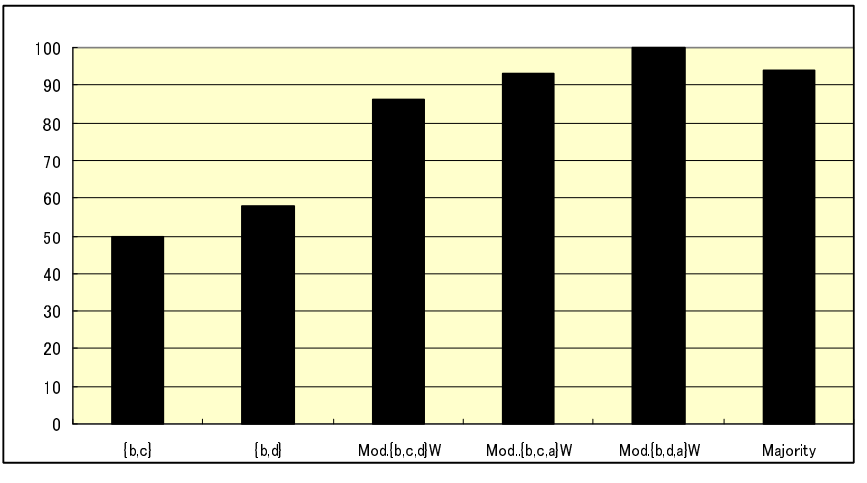


Fig. 4. Accuracy comparison in reducts-NN and modified reducts-NN

reduct {b,c,d}. Similarly, modified reducts {b,c,a} and {b,d,a} with weighting show higher accuracy. Majority voting is carried out as the final classification among modified reducts-NN. Further, the accuracy of reducts-NN and modified reducts-NN with weighting is compared in Fig.4. Fig.4 shows that reducts-weighted nearest neighbor method is effective using the modified result for the classification.

The weighting problem discussed here is generalized as follows. Among instances,

$$Dist(x_r, y_r) \leq Dist(x_r, z_s) \tag{10}$$

holds, where the instance z_s is the different s-class other than the r-th class. Further,

$$\sum_{i=1}^{m} \omega_i = 1 \quad \text{and} \quad \omega_i \geq 0 \tag{11}$$

are assumed here. The cardinality of the instances of the subset in the r-th class to the total number of pairs of instances is defined as

$$card\{(x_r, y_r) \mid Dist(x_r, y_r)\ in\ the\ r-th\ class\} \tag{12}$$

Then, the accuracy of the classification of the r-th class is defined as follows,

$$\xi_r = card\{(x_r, y_r) \mid Dist(x_r, y_r)\ in\ the\ r-th\ class\} / card\{U\} \tag{13}$$

Then, the total quality of the accuracy of the classification is

$$T = \sum_r \xi_r \tag{14}$$

Under the conditions of equations (10) and (11) , the maximization of the equation (14) is called the weight ω_i optimization here. The equation (14) as the fitness function, is maximized for the optimized weights ω_i by genetic algorithm[6,8]. The classification ability of the reduct, the modified reduct and the subset of the attribute variables is ordered in case of the weight ω_i optimization as follows,

$$T_{reduct} \leq T_{m.reduct} \leq T_{subset1} \leq T_{subset2} \text{ ……..} \tag{15}$$

,where T_{reduct} shows the accuracy value of the reduct weight ω_i optimization classification, $T_{m.reduct}$ shows that of the modified reduct, also $T_{subset1}$ is increased more one attribute variable than the modified reduct. Similarly, $T_{subset2}$ is increased more one attribute variable. The reason why the equation (10) holds, is as follows. When $\leq$ in the equation (10) does not hold, the increased weight ω_i of the last variable is set to be 0. Thus, the accuracy ability of the classification of the decision table is represented in the finite number of the weight ω_i optimization, which is determined from the equation (15). Thus, the ability of accuracy is stopped at the weighted modified reduct.

6 Conclusion

This paper discusses effective classification of the reduct of rough set followed by nearest neighbor classification. Up to now, rough set characteristics are well studied. But, the reduct-nearest neighbor classification, which is effective in time and memory processing, is not discussed. In this paper, for the effective reduct-nearest neighbor classification, the modified reduct and weighted distance between instances, are newly proposed. First, it is shown that the reduct-nearest neighbor classification has shortcomings in the accuracy of the classification. By proposing modified reduct concept, an improved accuracy is shown for the classification of the instances. Second, the ability of the accuracy of classification is developed on the weighted distance. The confidence problem of the reduct-NN classification is remained as the future problem[6,8,9].

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 Informatica 30(3-4), 345–358 (1997)
5. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
6. Ishii, N., Morioka, Y., Bao, Y., Tanaka, H.: Control of Variables in Reducts - kNN Classification with Confidence. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011, Part IV. LNCS, vol. 6884, pp. 98–107. Springer, Heidelberg (2011)
7. Bao, Y., Tsuchiya, E., Ishii, N., Du, X.-Y.: Classification by Instance-Based Learning Algorithm. 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 2006, vol. 1, pp. 699–708. IEEE Computer Society (2006)
9. Cheetham, W., Price, J.: Measures of Solution Accuracy in Case-Based Reasoning Systems. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 106–118. Springer, Heidelberg (2004)

Credit Scoring for SME Using a Manifold Supervised Learning Algorithm

Armando Vieira[1], Bernardete Ribeiro[2], and Ning Chen[1]

[1] Instituto Superior de Engenharia do Porto, 4200 Porto, Portugal
[2] Departamento Engenharia Informática, Universidade de Coimbra, Portugal
{asv,cng}@isep.ipp.pt, bribeiro@dei.uc.pt

Abstract. We propose a credit scoring algorithm based on the supervised ISOMAP to rate SME. By projecting the companies balance sheet data into a one dimensional component we obtain a smoother distribution of ratings while increasing the discriminatory capability of each rate in terms of the probability of default. The method is applied to a large dataset of French SME.

Keywords: Credit Risk, Credit Scoring, Supervised Learning, Isomap.

1 Introduction

Credit risk analysis is a very important and actual topic. In some cases the total loss due to bankruptcies can be as high as 5% of the nominal GDP. The causes of bankruptcy can broadly be assigned in two categories: intrinsic failure and network effects. The first can be originated by anemic sales; irresponsible management; accumulated deficit; low capitalization (lack of working capital or very high interest rates); random cause; build-up of inventory; excessive business investment. The network effects are caused by an aftershock of another company's bankruptcy (excess of bad debts), adverse macro-economic scenario or large non-enforceable accounts receivable [1,2]. In this work we deal with the first type of bankruptcy causes, although the second type is also important and still not well studied.

In order to advice the analyst or the investor, more than detecting the bankruptcy we need to have a credit scorecard to rate the company in terms of how close it is from default or the probability of becoming bankrupt. The difficulty of computing these ratings have several causes, but two are paramount: 1) all algorithms are retrospective, i.e, they project historical data into the future and 2) the only empirical evidence we have to gauge the accuracy of the models is when the company becomes bankrupt, i.e. the labeled data from the training set is either 0 – non-bankrupt or 1 - bankrupt. The algorithm has to infer the real status of the company between those extremes from a classifier that will learn a bimodal distribution picked around 1 and 0.

To address this problem we propose a new scoring algorithm based on semi-supervised manifold learning: the SSA. Many applications of data classification and data mining deals with a large number of unlabeled examples. Supervised nonlinear dimensionality reduction can be used as a preprocessing step before classification.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 763–770, 2012.

The rationale here is to map the high-dimensional data space into a lower dimensional space where classification methods do not suffer from the curse of dimensionality. As the explicit mapping is not found by the algorithm some learning methodology must be used. Our approach uses a set of training labels in the data set to provide a better construction of features and improve learning.

In this paper we address the problem of building a smooth and reliable credit scorecard algorithm. We want to explore the intrinsic structure of the financial data of companies in order to effectively compare them using a meaningful metric. Our aim is to improve credit scoring and classification with unlabeled examples under the assumption that the data resides on a low-dimensional manifold within a high dimensional representation space. In this case we make the most drastic dimensionality reduction, namely to project data onto a 1-dimensional manifold.

This paper is organized as follows. In Section 2 we describe our dataset, Section 3 deals with the topic of manifold learning and describe the algorithm. Section 4 presents the results.

2 Characterization of the Dataset

We used a sample obtained from Diane, a database containing about 100 000 financial statements of French SME companies [9]. The sample consists of financial ratios of industrial French companies, for the years of 2002 to 2007, with at least 10 employees. From these companies, 1511 were declared bankrupted in 2007 and 2272 presented a restructuring plan to the court for approval by the creditors. We decided not to distinguish these two categories as both signal companies in financial distress. From these dataset we build a balanced sample with 600 financial distressed firms, most of them small to medium size, with a number of employees from 10 to 800, corresponding to the year of 2007 – thus we are making bankruptcy prediction one year ahead.

Our database contains many cases with missing values, especially for defaults companies. For this reason we sorted the default cases by the number of missing values and selected the examples with 10 missing values at most. A final set of 600 default examples was obtained. In order to obtain a balanced dataset we selected randomly 600 non-default examples resulting in a set of 1200 examples.

The remaining missing data was treated as follows. For the ratios of the years 2003 and 2006 each missing value was replaced by the value of the closest available year; for 2004 and 2005, if values of the next and previous years were available, each missing value was replaced by their mean, otherwise it was replaced by the remaining value. In some cases there was no data available for a ratio in any of the years. In this very few cases the missing data was replaced by the median value of the ratio in each year. Finally, all ratios were logarithmized and then standardized to zero mean and unity variance.

This dataset includes companies from a wide range of sectors - see for Figure 1. However, due to the large number of attributes available, we used several ranking

algorithms to select the most relevant. From the initial 30 financial ratios defined by COFACE and included in the Diana database, we select only the 8 most relevant, namely: Number of employees, Liquidity ratio, Financial Debt /Equity, Financial Debt/Cashflow, Cashflow/Turnover, Working Capital Needs/Turnover, Total Assets/Turnover and EBITDA Margin. All ratios were normalized to zero mean and unity variance.

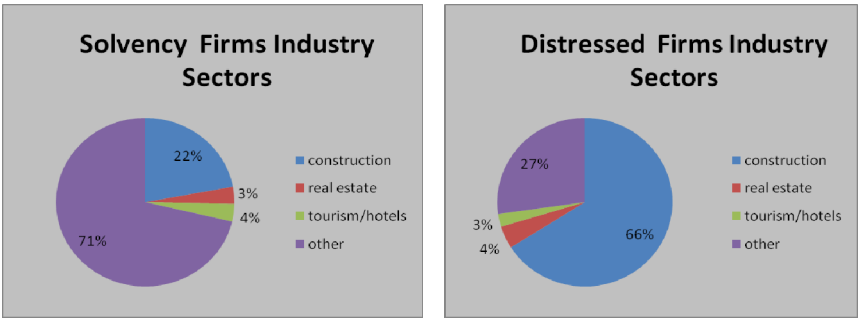


Fig. 1. Characterization of the dataset in terms of sectors of activity

3 Manifold Learning

In many real applications observational high-dimensional data can be cast into low-dimensional manifolds embedded in low-dimensional spaces, provided a suitable representation procedure is found. Instead of working with points in a high-dimensional space, classification and prediction algorithms can be easily used in these low-dimensional spaces sought from the embedded learning process [3, 5, 6].

Attempting to uncover this manifold structure in a data set is known as manifold learning. Manifold methods include a number of nonlinear approaches to data analysis that exploit the geometric properties of the manifold on which the data is supposed to lie. Manifold learning can be seen as an unsupervised feature extraction algorithm. We have previously applied it to the problem of bankruptcy prediction [4].

Although the structure inherent in many real world domains exhibit embedded low dimensional structures, as for example in image data or video frames, much research is being preformed in other areas.

3.1 Unsupervised Isomap

Unsupervised learning does not incorporate domain knowledge but its useful since in many cases we don't have label data. Unsupervised Isomap [3] consists of three main steps:

1) Estimates which points are neighbors on the manifold M, based on the distances $dX(i, j)$ between pairs of points i, j in the input space $\boldsymbol{X}$ by computing the weighted graph G of neighborhood relations given by the edges of weight $dX(i, j)$.

2) Estimates the geodesic distances between all pairs of data points in the manifold *M* by computing the shortest path distance on the *k*'s nearest neighbor graph built on the data set.
3) Applies classical Multi Dimensional Scaling (MDS) to the matrix of graph distances, constructing an embedding of the data in a *d*-dimensional Euclidean space *Y* that best preserves the manifolds estimated intrinsic geometry. Isomap assumes that there is an isometric chart that preserves distances between points.

The ISOMAP algorithm:

input: x_1ᶜ… $x_\backslash$, k

1. Form the *k*-nearest neighbour graph with edge weights $W_{ij} := \left\| x_i - x_j \right\|$ for neighbouring points x_i, x_j .
2. Compute the shortest path distances between all pairs of points using Dijkstra's or Floyd's algorithm. Store the squares of these distances in *D*.
3. Return $Y := MDS(D)$

It is assumed that for nearby points in the high dimensional space the Euclidean distance is a good approximation of the geodesic distance whereas for distant points this may not be true. Therefore, another technique described in ISOMAP is applied. It consists of building a weighted graph a *k*'s nearest neighbours where its edges are weighted by the Euclidean distances between nearby data points. Then a shortest path computation algorithm such as, Dijkstras or Floyds, will complete the calculus of the remainder geodesic distances.

3.2 The Supervised Isomap

The supervised version of the algorithm is based on the assumption that different features of the data can be captured by different dissimilarity measures. The algorithm builds up from a dissimilarity matrix $D(x_i, x_j)$ between two sample points x_i and x_j is defined as [6]:

$$D(x_i, x_j) = \begin{cases} \sqrt{(a-1)/a} & \text{if } c_i = c_j \\ \sqrt{a} - d_0 & \text{if } c_i \neq c_j \end{cases} \quad (1)$$

where $a = 1/e^{-d_{ij}/\sigma}$ with d_{ij} a distance measure (in our case Euclidean), σ a smoothing parameter (set according to the data 'density'), d_0 a constant ($0 \leq d_0 \leq 1$) and c_i, c_j are the data class labels. If dissimilarity between two samples is less than 1, points are in the same class, otherwise points are in different classes. The parameter d_0 allows that points in different classes to have a smaller value of dissimilarity, than those in the same class. In general the inter-class dissimilarity is larger than the intra-class dissimilarity (depending on the parameter d_0) conferring a high

discriminative power on the method, which is very powerful for further classification. This approach was recently applied to our dataset with good results [4].

3.3 The Semi-supervised Algorithm SSA

Since bankruptcy prediction is a high-dimensional and highly unbalanced problem with incomplete information, both supervised and unsupervised approaches have its drawbacks. Supervised manifold learning is a powerful approach. However, it does not incorporate all the data since we don't have information about the real situation of a non-bankrupt company. Furthermore, if we use all points to train the Isomap, we may obtain a crispy map.

Our objective is to build a map that smoothly relates companies; from the worst possible situation to the healthier one. In order to accomplish this, we need a metric system to effectively compare companies through their financial situation. However, supervised Isomap may not be suitable because it forcefully distorts the original structure of the input data no matter whether there is noise in the data or not and how much noise is in it.

Much in the same way as other seeded semi-supervised algorithms, our approach uses a fraction of labeled data (seeds) to constrain the learning of the Isomap [7]. We consider prior information in the form of on-manifold coordinates of certain data points. As a first step we choose a fraction of seed points at random. We run the supervised Isomap to constrain the map on this subset of points. Then we run the unsupervised Isomap to calculate the new MDS distances that does not violate the membership condition imposed by the seed points. Finally, after the implicit mapping Y is created, we train a generalized neural network (GNN) to retrieve an explicit map for fast recover of results. The algorithm is schematically presented in next picture:

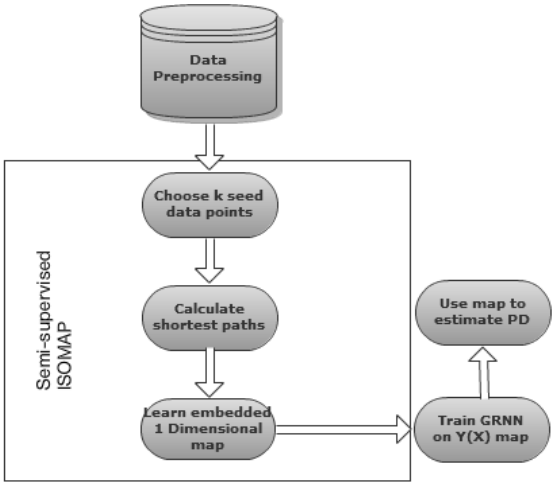


Fig. 2. (Left) Output distributions for the Logistic regression. Red corresponds to bankrupt and blue to healthy companies. Arbitrary unities are used. (Right) Output distributions of the classifier for SSA with $k = 4$. The continuous lines are Gaussians fits to the healthy (blue) and bankrupt (red) companies.

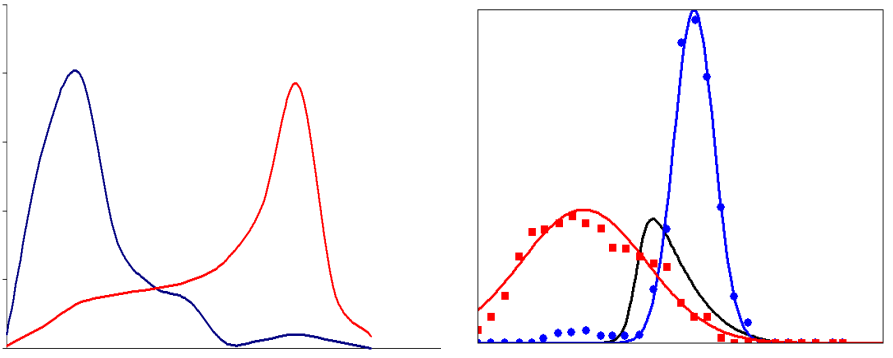


Fig. 2. (*continued*)

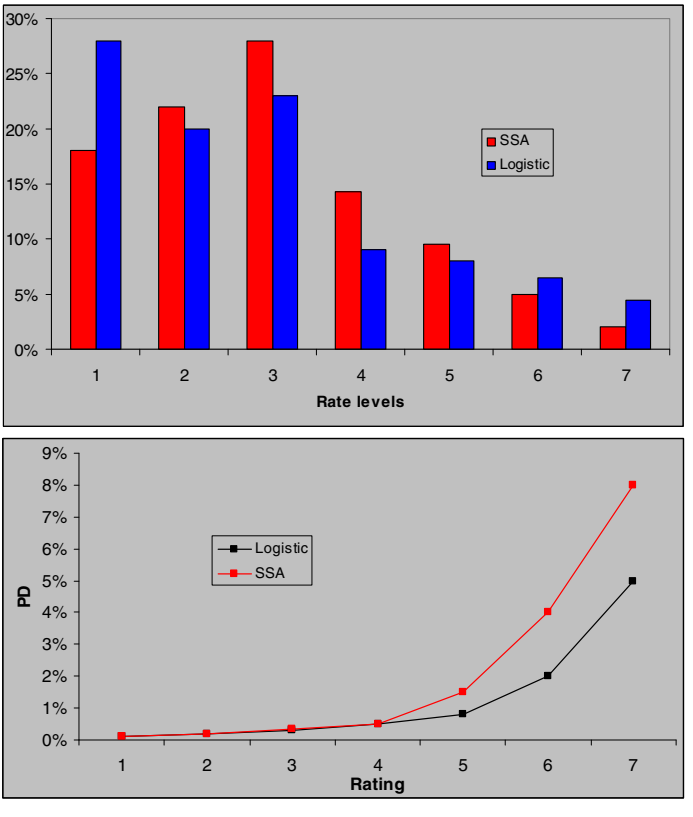


Fig. 3. (Top) Ratings distribution with Logistic and SSA over the full dataset. (**Bottom**) the corresponding probability of default (PD).

4 Results

We run the SSA algorithm using $k = 4$ neighbors and setting the parameter $d_0 = 0.5$. These parameters were optimized in our previous work [4]. We used two different percentages of seeds for semi-supervised algorithm: 10% and 30%. Finally the general regression neural network (GNNR) was trained using a Matlab code.

Table 1. Performance comparison of the algorithms. SSA – 10 and SSA – 30 corresponds to using 10% and 30% of seed points, respectively

	Precision (%)	**Recall (%)**
S-Isomap	87.94	86.79
SSA – 30	85.77	84.44
SSA – 10	84.03	83.05

We test our algorithms using data from 2006 to make one year ahead prediction, i.e., prediction of failures in 2007. First we run the algorithm in order to determine the level of degradation in the accuracy of classification for not using all the labeled data - Table 1. We see that, even with 10% of seed points, the accuracy is not substantially compromised when compared with the full supervised version of Isomap - see [4]. From now on, we use 30% of seed points, so refer to SSA as SSA – 30. Note that our focus is not in the accuracy *per se* but to obtain a smooth and stable map to feed the credit score card algorithm.

In figure 2 we present the distribution of outputs over our dataset for the Logistic regression. Note that the distribution is strongly bimodal with some overlap. In the same figure we also plot the distribution produced by our algorithm - SSA. Note that in both categories we obtain relatively smooth Gaussian distributions – being the bankrupted distribution somehow skewed. From the fits of figure 3 we can estimate the probability of default, defined as

$$PD(x) = \frac{f(x)}{f(x) + g(x)}. \tag{2}$$

where $f(x)$ is the distribution of non-bankrupt (healthy) companies and $g(x)$ the distribution of distressed (bankrupted) companies for the balanced dataset. In this figure we also plot the total expected number of defaults occurring at a given level (black line):

$$N(x) = (f(x) + g(x))PD(x) \sim f(x)PD(x). \tag{3}$$

In practice the most important aspect of the classifier algorithm is to use its discriminatory capability to bind a credit rating scorecard and classify the companies in terms of risk. We have already proposed a stable rating algorithm rating which is more reliable than traditional approaches [10]. In this case we want to produce a smoother algorithm – in the sense of projecting the true distribution of companies ratings - even if we sacrifice the accuracy of the classifier.

In figure 3 we plot the distribution of ratings according to our algorithm compared to the widely used Logistic regression approach. Note that in this case we are using the full dataset according to Section 2. Rating levels were set by slicing the outputs into $M = 7$ equidistant levels. Note that the distribution of ranks is more uniform and the highest risk rates have a steeper decline.

5 Conclusions

We presented a scorecard algorithm for rating SME that can produce smoother rating categories without compromising the accuracy. The algorithm is based on a semi-supervised version of the Isomap, a non-linear dimensionality reduction technique. In future work we want to test the stability of the algorithm over time.

References

1. Vieira, A., Neves, J.C.: Improving Bankruptcy Prediction with Hidden Layer Learning Vector Quantization. European Accounting Review 15(2), 253–271 (2006)
2. Atiya, A.F.: Bankruptcy prediction for credit risk using neural networks: A survey and new results. IEEE Trans. Neural. Net. 12(4) (2001)
3. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
4. Ribeiro, B., Vieira, A., Duarte, J., Silva, C., Neves, J.C., Mukkamala, S., Sung, A.H.: Bankruptcy Analysis for Credit Risk using Manifold Learning. In: ICONIP. LNCS, Springer, Auckland (2008)
5. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
6. Geng, X., Zhan, D.G., Zhou, Z.H.: Supervised nonlinear dimensionality reduction in visualization and classification. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics 35(6), 1098–1107 (2005)
7. Yang, X., Fu, H., Zha, H., Barlow, J.: Semi-supervised nonlinear dimensionality reduction. In: Proceedings of the 23rd International Conference on Machine learning, ICML 2006, pp. 1065–1072. ACM, New York (2006)
8. Chatpatanasiri, R., Kijsirikul, B.: A Unified Semi-Supervised Dimensionality Reduction Framework for Manifold Learning. Neurocomputing (2010)
9. Vieira, A.S., Duarte, J., Ribeiro, B., Neves, J.C.: Accurate Prediction of Financial Distress of Companies with Machine Learning Algorithms. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 569–576. Springer, Heidelberg (2009)
10. Chen, N., Vieira, A., Ribeiro, B., Duarte, J., Neves, J.C.: A stable credit rating model based on learning vector quantization. Intell. Data Anal. 15(2), 237–250 (2011)

Sentiment Proxies: Computing Market Volatility

Stephen Kelly and Khurshid Ahmad

Trinity College Dublin, Ireland
kellys25@scss.tcd.ie

Abstract. Macroeconomic announcements can have an influential effect on the price, and related volatility, of an object traded in financial markets. Modeling the impact of a relevant announcement on a specific commodity is of interest in building financial models of such objects. The announcements may generate false hopes or correctly indicate the ways in which the prices of objects may change. We describe a bootstrap model which is an attempt to analyse the impact of a publicly released oil inventory announcement on the price of oil futures contracts. A comparison with traditional econometric regression model is presented and we perturb the traditional models with: (a) a dummy time series that contains the dates on which the announcement is made, and (b) a sentiment time series that reflects the sentiment of the market. The sentiment time series is generated using natural language processing techniques.

1 Introduction

The movement of prices is the result of varying market forces. These are the so-called endogenous and exogenous variables that impact on the price of entities traded in various markets including for example equities in stock markets and commodities in commodity exchanges. The endogenous variables include past prices, levels of supply and demand for example. The exogenous variables comprise of governmental policies and their impact and the timings of the announcements of such policies related to endogenous variables; increasingly the role of affect, especially sentiment appears to play major role in determining the values of endogenous variables. The computation of the impact of past prices on current price, and indeed future prices, relies on building an econometric model and using numerical analysis techniques for implementing the model. However, the inclusion of the impact of policy announcements on prices relies on a range of heuristics that effectively use proxies[5,3] and the same is true of sentiment[8,9]. The inclusion of market sentiment on prices requires techniques developed in natural language processing, in information extraction and increasingly in machine learning. Sentiments are extracted from news reports and blogs written by traders in specific markets, and thus sentiment extraction involves distributed processing whereby data is obtained from a range of sources especially online data such as RSS feeds.

We focus on the impact of the announcement of estimated reserves of crude oil, in the USA, on the prices of future contracts of oil and use the analysis of

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 771–778, 2012.

the possible impact of the announcement on future prices. In effect, we have developed an empirical distribution of randomness associated with price change for a time frame around the announcement. Our estimates of the price variation, and thus the volatility of future prices, due to the announcement appear to be in agreement with the observed data. We compare the results of our prediction with traditional regression analysis in three stages: pure regression analysis of prices only, regression with prices and announcement proxy as the timing and we introduce an evaluative sentiment series. The sentiment analysis was carried out on a set of oil news and blogs using natural language processing and data mining techniques[1].

Many studies have shown that macroeconomic announcements can have an influential effect on a commodity and its volatility. There are a number of studies that have investigated the volatility, both literal and metaphorical, in the oil markets. Swings in crude oil prices are often accompanied by counter-swings in the inventory of oil supplies. It has been suggested that deviation in the "normal" or expected inventory level, determined by standard market indicators, affected price movement in the short term[12].

The valuation of energy firms is closely related to and driven by commodity prices. It is a consideration then to examine economic factors that can have an impact on these firms and companies (see for example [4]). Our motivation was to investigate the impact of the crude oil reserve announcement on the immediate change in the price of oil future contracts and to use this analysis to build a price estimation model incorporating the impact. This model will perhaps help in understanding the behavior of oil traders as they react to oil inventory announcements by using heuristics learnt over a period of years.

2 Data Set

2.1 Oil Trade: Future and Spot Prices

Oil traders use two instruments whilst trading: First, spot prices for the light sweet crude - the so-called Brent crude oil(Symbol as BFOE) ; Second, futures contracts for light sweet crude. The price of the front month of the light sweet crude oil futures contract is a typical proxy for the cost of imported crude oil. Future contracts can be exchanged between buyers and sellers for an expiration from a minimum of 1 month to a maximum of 8 years; the most extensively traded contract is the Prompt month contract for crude-oil prices of a special blend of light crude oil produced in the US, called the West Texas Intermediate (Symbol as WTI CL1)[10][1].We use a five-minute trading series (2007-2010) for the price quoted for CL1 con-tracts. The prices at the time of the announcement, thirty minutes before and after were noted, as well as the price returns and price differences after the time of the announcement and used in the analysis.

[1] Continuous derived contracts do not focus on a specific delivery month so much as a relative delivery time. In this way it is possible to create a historical time series of the price of the contracts sometime in the future by way of looking at the actual price at the time of delivery relative to the publication date.

2.2 Oil Inventory Announcement

The US Energy Information Administration (EA) releases a weekly report (WPSR)[10] that contains general information and specific data about crude oil sup-ply, production, consumer consumption and refinery utilization. Fluctuations in the data within the report may reflect demand fundamentals, influence market speculation and can have a measurable impact on the price of oil. The identification of significant jump components in oil futures prices is undertaken and an analysis made of the relative contribution of jumps to the realized price change in the futures contracts price. The investigation leads on to see if significant jumps are due to the Energy Information Administration's (EIA) inventory news announcement related by certain factor such as the announcement's timing.

2.3 Sentiment Data Series

The sentiment can be found in a variety of formal and informal publications. The formal comprises of news, specific to oil, and the latter deals with oil blogs. The news was collected using RSS feeds. The timing of the news is gathered from the time stamp carried in the news item.

3 Methods

3.1 Bootstrap Method and Algorithm

To monitor price adjustment due to the announcement's effect the resulting price change post-announcement is looked at with the dependent variable being the time frame chosen. Previous studies [6,9] have suggested a 15-minute window. However it is suggested that the impact is only significant up to two hours after the release of macroeconomic information and is often drowned out in the subsequent random fluctuation[2]. Taking p(0) as the price at the time of the announcement, p(30) is price thirty minutes after the announcement and p(-30) before the announcement. The price change between these times is calculated as well as the price change across the entire hour around the announcement, from 10.00 to 11.00, these figures are then used for analysis and as the working data set that is modeled and used to train the system. A bootstrap method is used to evaluate the properties of the statistical estimators without any model assumptions [7]. The bootstrap algorithm is as follows

1. Select B independent bootstrap samples, each of size n sampled from the empirical distribution $\hat{F}$ where x^* is not the actual observed data set x but a random sample taken with replacement.

$$\hat{F} \rightarrow x^{*1}, x^{*2}, ..., x^{*B} \tag{1}$$

2. The bootstrap replication corresponding to each bootstrap sample, in this case the sample mean, is found for each bootstrap sample. The function *s()* is applied to $x^*$where $b = 1,2,...,B$.

$$\hat{\theta^*}(b) = s(x^{*b}) \tag{2}$$

3. From this calculation the standard error is estimated and the function mentioned in (2) in this case is the sample mean of the bootstrap dataset. The confidence interval for the parameter $\hat{\theta}$ is found at a 95% significance level.

3.2 Sentiment Analysis

The sentiment data is analysed using a bag of words approach and the content of the news is analysed with respect to a glossary of affect words. Typically the glossary of aspect words may contain domain specific words with connotations of affect; for instance, the word *crude* has negative connotations in an affect dictionary whereas it is used in the oil domain as a product category. We have used a general purpose affect dictionary together with a domain specific dictionary. This approach of a hybrid dictionary allows us to look at domain specific terms and use those terms as a filter for news items. The frequency of affect words within a given time interval is used as a basis for the construction of an affect time series.

3.3 Towards a Hybrid Model: Announcement Effects

The intention was to model the announcement's influence on the price series through empirical analysis and also observe the results from a typical regression analysis. It is the motivation to better interpret the announcement's effect on price behavior and to give an estimation of the price volatility. An overview of the intended regression model can be seen from the following equations.

$$p(t) = \sum_{i=1}^{n} \alpha_i p(t-i) + \varepsilon \tag{3}$$

$$p(t) = \sum_{i=1}^{n} \alpha_i p(t-i) + \beta Announcement(t) + \varepsilon' \tag{4}$$

$$p(t) = \sum_{i=1}^{n} \alpha_i p(t-i) + \gamma Sentiment + \varepsilon'' \tag{5}$$

Where *p(t)* is the actual value of the series and $\sum_{i=1}^{n} \alpha_i p(t-i)$ is the expected value with regression coefficients $\alpha_i = \Sigma p(t).p(t-i)$. The first regression model consists of regressing price against itself with some lag n and using this analysis as a bench mark for subsequent regressions. The original regression equation (3) was performed with a time series of daily closing prices for 2011. Each of the models analysis' were repeated for price returns, $r(t) = log[p(t)/p(t-1)]$. To analyse the effects of the oil announcement a simple linear regression model was used but with a (dummy) variable, *Announcement(t)*, such that *Announcement(t)=1*, when t is the time of the announcement (e.g. t=Wednesday for our study), else *Announcement(t)=0* (Eq. 4). For incorporating the impact of sentiment on a given day t, the variable *Sentiment(t)* is the relative frequency count of sentiment on the day normalised by the total number of words comprising all texts published on day t (Eq. 5).

4 Simulation Analysis

An independent program was developed, written in Java, to conduct a series of simulation experiments supplied with price data of the one month oil future contracts (cl1) and a schedule of the oil inventory announcements. The system extracted the relevant prices around the time of the inventory announcement from continuous price data. The change in price of the cl1 contracts around the time of the announcement was found and used to train the system. The bootstrap algorithm was implemented by the program on the supplied data and gave the descriptive statistics used to characterise the distribution of the price changes. The regression analysis of prices alone, with announcements and sentiment was carried out by using *Rocksteady* - an affect analysis program that can compute regressions equations (3-5); the statistical simulation program for the bootstrap model (Eq.1) was integrated into *Rocksteady*. (This system was developed with the help of the Irish Government and Treocht Ltd.)

4.1 Data Used

For the study of the impact of announcements we focused on future contracts (2007-2011, 5 minute data) and for the hybrid study we used both spot and future prices for the year 2011 only. We also perform he analysis on a return series also. The sentiment affect series used in the regression analysis was derived from news collected from 57 news sources from the general area of oil industry related news. The period was during February to November 2011 giving 302 days. There were 13063 news stories collected comprising 8,510,428 single words or tokens; the news flow rate was 1.8 stories per hour for the 10 month period covering the *Arab Spring*.

4.2 Simulating the Bootstrap Model

The data set for each year was prepared and a sample distribution built from this. An estimate of the standard error and confidence interval of 95percent for the estimated value was found. The following table shows the results from the system for the statistical estimates generated for each individual year of data in the form of the year-end mean price change. The announcement's effect on prices was a function of the timing of its occurrence. The results of the impact of the announcements and its effect are good in that the actual value of mean price change, for each of the 4 years, was within two standard deviations of the predicted value.

Given that returns are not serially correlated, an important test of our method will be to see what predictions our system made of the change in the returns: We predicted the results for four weeks in January 2010, based on a bootstrap constructed on 5-minute prices for the year 2009 for the month of January 2010. Price return, logarith of the ratio of prices at 1100 and 1055 hours (Table 2). We appear to have a better agreement with our return predictions, at short time scale, than with the price predictions at a longertime scale.

Table 1. The estimated mean price change($\bar{x}^*$), standard error($\widehat{se}_B$) and confidence at 95% interval for each year end. The forecast is for a week ahead and the result is compared to the actual mean price change($\bar{x}^*$).

Year	$\bar{x}^*$	±95% Confidence	$\widehat{se}_B$	$\bar{x}^*$
2010	0.0514	0.1452	0.0741	0.0481
2009	-0.1194	0.2473	0.1262	-0.0998
2008	-0.3323	0.3631	0.1852	-0.3176
2007	0.6211	0.7936	0.4049	0.5054

Table 2. The estimated mean price return($\bar{x}^*$), standard error($\widehat{se}_B$) and confidence at 95% interval predicted a week before the Announcement. Compared to the actual mean price return($\bar{x}^*$).(All numbers scalled by 10000).

January2010	$\bar{x}^*$	±95% Confidence	$\widehat{se}_B$	$\bar{x}^*$
6th	0.717	3.140	1.604	0.762
13th	0.431	2.840	1.447	0.473
20th	0.382	2.673	1.363	0.387
27th	0.402	2.868	1.464	0.454

4.3 Analysis of Prices and Returns Using Announcement and Sentiment

The computations in this section relate to a daily price, announcemtns and sentiment series covering a period of 10-month (Feb-Nov 2009). The price and return predictions in the case of auto-regressive models are often given in terms of the coefficient of determination (R-squared value). This coefficient determines the amount of variability in a data set that can be explained by a regression model: If the serial correlation in price values is a strong one, then an auto-regressive model comprising past values of prices only should be a good model and R-squared is approximately 1. However, if the oil market has been 'spooked' or 'elated' by exogenous events the the serial correlation will be weakened and perhaps a price-regression model with *announcements* and *sentiment* effects will explain the variability in the data set. We used our regression models on a shorter time series (30 days) rolled or moved over the whole period (282 days), and a longer series of 282 days (cumulative series).

4.4 Moving Window and Cumulative Series

The analysis of both prices and return over a shorter time period (30 days), covering the whole period (282 days) using a moving window, shows that our regression models (i) account for over 65% variability in the price data, for both spot and futures, with announcements and sentiment data increasing R-squared by 3% only; and, (ii) the regression models for returns, both for spot

and future series, only account for 25-30 variability, with announcements and sentiment increasing the coverage by 7% for the spot returns but decrease R-squared marginally by the inclusion of exogenous variable s (Table 3).

Table 3. Shows the average R-squared value for the series for each regression model (Equations 3,4, and 5) for price and returns for moving window. (N=30).

		Price Only (a)	(a)+Announcement (b)	(a)+Sentiment (c)
Price	Brent Spot	0.650	0.688	0.684
	Cl1 Future	0.681	0.694	0.699
Return	Brent Spot	0.250	0.320	0.320
	Cl1 Future	0.315	0.300	0.292

The regression models when simulated using the whole period account for 88% variability in the spot prices and over 90% variability in the future prices. The return series calculations show overall low R-squared (compared with price series) but the variability coverage improves with the inclusion of announcements and sentiment (Table 4).

Table 4. Shows the average R-squared value for the series for each regression model (Equations 3,4, and 5) for price and returns for the whole series (N=282)

		Price Only (a)	(a)+Announcement (b)	(a)+Sentiment (c)
Price	Brent Spot	0.885	0.886	0.886
	Cl1 Future	0.925	0.926	0.926
Return	Brent Spot	0.026	0.027	0.036
	Cl1 Future	0.048	0.051	0.054

5 Concluding Remarks

The empirical distribution used in computing the impact of the announcement of strategic oil reserves in the USA predicts the direction of change in prices correctly over a four year sample that comprised the end of a boom (2007), recession (2008-09) and some green shoots of recovery (2010) (Tables 2 &3). Our model, on average, tends to overestimate the price change (between 5-22%) and underestimates the mean return (1-10%). The linear regression analysis, known to be a poor predictor and equally worse model for computing variance [11], performs better for longer samples of prices (and returns) than is the case for the former, and the performance is improved marginally if we could announcements (through a dummy variable) or include negative sentiment (through frequency

counts). Our focus of work in the future will be to take into account the fractal nature of oil price movement and the concomitant heteroskedastic behaviour of the movements. We are exploring how to incorporate the output of the empirical model of the announcements to replace the dummy variable approach we have had. The inclusion of other dimensions of affect - affirmation or positive affect, affects related to strength and activity, and an improved risk and compliance vocabulary is to be added to our affect dictionaries. The support of Enterprise Ireland (Project Faireachain, IP/2009/0595) is gratefully acknowledged together with that of Prof C. Kearney, now at Monash University, and Mr D. Sciro, CEO Treocht Ltd, especially for providing us with the oil data sets.

References

1. Ahmad, K.: Affective Computing and Sentiment Analysis: Metaphor, Ontology and Terminology. Springer, Heidelberg (2011)
2. Almeida, A., Goodhart, C., Payne, R.: The effects of macroeconomic news on high frequency exchange rate behavior. J. Fin. and Quant. Analysis 22, 383–408 (1998)
3. Andersen, T.G., Bollerslev, T., et al.: Real-time price discovery in global stock, bond and foreign exchange markets. NBER Working Paper Series, No. 11312(2), pp. 251–277 (2007)
4. Boyer, M.M., Filion, D.: Common and Fundamental Factors in Stock Returns of Canadian Oil and Gas Companies. Energy Economics 29(3), 428–453 (2007)
5. Bauwens, L., Ben Omrane, W., Giot, P.: News announcements, market activity and volatility in the $ foreign exchange market. J. Int. Money and Finance 24(7), 1108–1125 (2003)
6. Ederington, L.H., Lee, J.H.: How Markets Process Information: News Releases and Volatility. J. Finance 48(4), 1161 (1993)
7. Efron, B., Tibshirani, R.: An introduction to the bootstrap. Chapman Hall (1993)
8. Groß-Klußmann, A., Hautsch, N.: When machines read the news: Using automated text analytics to quantify high frequency news-[..]. J. Emp. Finance 18(2), 321–340 (2011)
9. Payne, R.: Announcement effects and seasonality in the intra-day foreign exchange market. Discussion paper series, vol. 238. LSE Financial Markets Group (1996)
10. WPSR (Weekly Petroleum Status Report), http://www.eia.gov
11. Xu, B., Ouenniche, J.: A data envelopment analysis-based framework for the relative performance evaluation of competing crude oil prices volatility forecasting models. Energy Economics 34(2), 576–583 (2012)
12. Ye, M., Zyren, J., Shore, J.: Forecasting crude oil spot price using OECD petroleum inventory levels. Int. Adv. in Economic Research 8(4), 324–333 (2002)

Performance Evaluation of Hybrid Implementation of Support Vector Machine

Konrad Gajewski and Michal Wozniak

Department of Systems and Computer Networks, Wroclaw University of Technology
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{konrad.gajewski,michal.wozniak}@pwr.wroc.pl

Abstract. This article focuses on the problem how to shorten the time required for training and decision making by classifiers based on Support Vector Machines techniques. We propose the hybrid implementation of mentioned above algorithm which uses parallel implementation of SVM based on GPU programming model in a distributed computing system using MPI protocol. To estimate the computational efficiency of the proposed model a number of experiments were carried out on the basis of UCI benchmark datasets. Their results show that using parallel model in distributed computing environment can reduce computation time compared to both classical SVM used single processor only and to SVM implementation based on GPU.

Keywords: distributed computing, SVM, parallel classification, GPU, MPI.

1 Introduction

Interest in data mining techniques is a natural phenomenon resulting from the enormous technological progress that has been occurring before our eyes for a few years. At this time the amount of generated information increases, which makes extraction of essential more and more difficult. The demand for better and faster methods of acquiring knowledge is limited by the resources of centralized computing units. A modern economy requires construction of new algorithms designed to use modern equipment more efficiently in processing huge portions of data. The structure of modern computing allows to replace the worn model, used on a single machine, with a parallel processing on a multiple computers.

The main objectives of the work are as follows:

- verifying whether various implementations based on the concept of parallel computing can speed up solving problems of multiclass classification,
- evaluating whether a single graphics unit or network of computers with such units can give a positive performance growth comparing to the classic version of algorithm,
- determining whether it is possible to minimize the time of computing tasks using modern computing systems so that it can process data in real time.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 779–786, 2012.

This study identifies opportunities for future implementation improvements. The content of the work is as follows. The next sections presents theoretical introduction to Support Vector Machines (SVM) technique of solving classification problems, which is the subject of analysis during the comparison of different implementations. The following part focuses on technical aspects of used implementations of the distributed implementations of SVM algorithm. Then the results of computer experiments are discussed. The last section concludes the paper.

2 Support Vector Machines

SVM is a set of related supervised learning methods, used to solve the problems of classification and regression. Having a set of training data in which each instance is assigned to one of two categories, the SVM algorithm builds a model. It tries to predict which category will be assigned to new examples. SVM model is simply a representation of projection of points in the plane. Examples of each of the categories are separated by empty space as broad as possible. New examples are projected on the same plane and are classified depending on which side of empty space they are placed [1, 4, 5, 9].

Multiclass SVM algorithm tries to reduce a single multiclass problem into many binary classification tasks [6]. A two-class classifier is assigned to each task. The classifier generates function which gives large values for examples assigned correctly to the class and relatively low for poorly assigned ones. Two methods commonly used for creating binary classifiers are *one-against-all* and *all-against-all*.

3 Implementation

In order to examine the diversity in computing performance of the proposed hybrid computing model, we compare its performance with two reference implementations.

3.1 Reference Implementation

LibSVM [13] is the simplest implementation of the SVM algorithm. It does not implement advanced techniques to utilize more than one process to solve problems, but it provides a simple interface for its easy integration with other programs. In our research it serves as the reference solution.

Herrera-Lopez et al. [8] propose the implementation of a classifier for multiclass SVM algorithm designed for GPU graphic cards [10] - GPUSVM. The resources and the parallelization level of the latest multi-core hardware can be successfully used on large-scale training of multiclass classification tasks. Despite the fact that there are various methods of problem-solving training part of algorithm, the authors chose the method of SMO (*Sequential Minimal Optimization*) [2]. Its popularity and ability to define patterns in partial data were valued at most [3, 14]. In this work we extend the scope of those experiments.

3.2 NET-GPUSVM

Because our proposition could use Message Passing Protocol (MPI) [14], therefore let's present this concept shortly. MPI is a communication protocol for programs that perform calculations in parallel (called *parallel* programming). It allows applications to run concurrently on a certain number of separate computers connected to the network. MPI programming model is based on the transmission of messages. In the system of such specifications, each of the parallel processing of processes communicating with others by sending information. Unlike the multi-threading, where the various threads share variables of the program, each MPI process has its own local set of variables, where other processes do not have access. Therefore, the MPI protocol processes may be as much fragmented, as a network makes it possible. They may also differ from each by configuration of the unit on which they are running, even in terms of architecture (if application allows).

Most programs are written using MPI in accordance with the principle of "one program, multiple data" (called *Single Program Multiple Data* - SPMD). Each process runs the same program, but works in another part of the data. Such processes usually perform a large part of the calculations on data locally available. MPI supports the SPMD model by allowing users to easily run the same program on many different machines (nodes) in a single command. At the beginning each process is identical, but is assigned a rank that uniquely identifies it. Rank among the MPI processes is an integer from the interval [0, P-1], where P is the number of processes running under the program. Processes can check their rank, which allows them to behave in different ways and send messages to other processes executing the same task through a number of their rank.

We propose the NET-GPUSVM which is the distributed implementation of SVM based on GPUSVM. The modification is made in order to run parallel code for the graphic cards on multiple units connected to a network. The basic algorithm was transformed in such way that the tasks could be done on separate machines. These machines do not share memory and data. Modification of the implementation requires splitting dataset into independent subsets, which are used for calculations on each machine. Let's present the main phases of the proposed modification.

Disaggregating Phase. It is identical for both the training of the algorithm, and classification. A given dataset used in the operation are randomly split into N subsets. The number of subsets is equal to the number of processes that are involved in the calculation. This phase is carried out in the rank 0 process (root). At that time, computational processes are waiting for their subset. Then the method *Scatter* of MPI library is used for distribution subsets to the corresponding processes. Function of scattering data between processes, working in such a way that the *root* sends out an array of elements specified as an argument between all the other nodes. The first element of the process goes to 0, the second to process 1, etc. Considering the fact that implicitly data is send to the MPI processes beginning from rank 0 (which is our unit of management and is not involved in the calculations), we must enrich the array of bogus component. Placing it at the beginning of the array, it will be sent to *root* to keep the order established by the library.

Since the beginning of the program child processes monitor the network all the time using the same Scatter method. As rank 0 process finishes sending data, it goes into the stand by mode. At that time, the other processes receive data and begin processing immediately.

Data Processing Phase. This phase of training differs from testing phase. Although in both cases subsets of main dataset are sent to the nodes, each unit needs whole trained dataset for classification. It is understandable, since the classification cannot be carried out with only a partial result of the training.

Therefore before processing the *root* is using *Broadcast* method to send the whole trained dataset to each node. After finishing the calculation of the training, all processes (including *root)* are retained by *Barrier* method. This is the moment of simple synchronization. Before proceeding to the third phase, all computing nodes have to complete their activities, and the managing node be kept in standby mode.

Data Retrieval Phase. Shortly after synchronization of the system, all nodes call Gather method. This operation will return an array with data only to the unit recognized as root. The element at position i in the array corresponds to the value sent through the process with a given rank.

Phase of Merging Training/Testing Dataset. In last stage of the algorithm computing nodes remain idle, while the main process combines the subsets of the calculated data in a result set.

4 Experiments

The main aim of the experiments was to compare computational time of the SVM implementations: LibSVM, GPUSVM, and NET-GPUSVM.

4.1 Set-Up

The same parameters were used for each algorithm, however they varied for each of the datasets. The data presented in the tables are derived from a single test run, because the specification of tested SVM algorithm does not need to conduct a greater number of runs for verification.

All measurements presented in this section were performed on a machine equipped with dual-core processor Intel DualCore E7400 with 2.67GHz clock speed, 3 GB of memory and Windows 7 (64 bit) operating system. Video card that was used is a NVIDIA GeForce 9800 GT with 112 stream processors, each with a frequency of 600MHz. The card has 512 MB of shared memory and memory bandwidth equal to 57.6 GB/ s.

The comparison of performance between generations of machines were made between machine described above and the computer used in the work of Herrero-Lopez [8]. Its configuration is as follows: Quad Intel Core i7 920 with a speed-core 2.67 GHz, 6 GB of memory and Ubuntu 9.04 (64 bit) operating system. Used video card: a NVIDIA Tesla C1060 with 240 stream processors, each with a frequency of

1.3 GHz. The card has 4 GB of dedicated memory and its bandwidth equal to 102 GB/s.

To carried out the experiment we developed the virtual environment that allows to perform calculations using the communication via the MPI interface. This environment is a simple cluster with a small number of nodes connected in a star topology network Due to the nature of the equipment on which the calculations are carried out (graphic cards), several virtual machines were created on a single powerful computer. This allows to control the delay of transmission of data packets, which is inherent in the exercise of distributed tasks. In case of physically existing network it would be hard to resemble same results. Its limited capacity is a undoubted disadvantage. However the impact of a key elements, which are communication between processes and the distribution of data, on the differences between the performance of individual implementations. The idea of using virtual machines has evolved to the form of starting a program for each process (which is the basic idea of a mechanism based on the MPI interface.) This allowed to eliminate the demand for computing power to run the operating system for each virtual machine and focus resources on solving problems.

All implementations were tested on the same benchmark datasets [12]. Web was used to verify the accuracy of the calculation of binary classification tasks. USPS, Shuttle, Letter datasets were used to analyze the performance of classification multiclass tasks. Size of sets, their parameters used in the training and background are presented in Table 1. RBF kernel method was used for all sets in classification phase.

Table 1. Description of datasets (*C*, *B* - parameters of the RBF method)

Dataset	Background	Training points	Testing points	Features	Classes	C	B
Web	UCI	49749	14951	300	2	64	7.8125
Usps	UCI	7291	2007	256	10	10	1/256
Shuttle	UCI	43500	14500	9	7	2048	8
Letter	UCI	15000	5000	16	26	16	12

Each dataset has been processed for all configurations of machines. As already described in section 3 - each call, in addition to client nodes, includes one process for the managing node - the synchronization server, which delivers and retrieves data. Calculations for NET-GPUSVM were performed for the following virtual network configuration using MPI protocol:

- two separate processes on one host (2P1H),
- four separate processes on one host (4P1H),
- one process for each of two hosts (1P2H),
- one process for each of four hosts (1P4H).

Beside the implementation designed for virtual nodes, we decided to evaluate the performance of additional configuration. It consisted of using the dataset divided the same way as in the case of a network-based approach. The difference is that the

computational steps are executed sequentially on one machine rather than sending partitions to hosts. This approach allows to observe the performance of calculating smaller portions of data. It should also give better results than a regular GPUSVM implementation because operations (which are among the most time-consuming tasks) will be performed on matrices of much smaller dimensions. In this case, the optimal duration of calculation will only be the result of multiplication *NxT,* where *N* is the number of partitions, for which they were divided into a core set of data, and *T* is the time for the task on a single data partition. The experiments were carried out for three (3DP) and five (5DP) partitions of data,

4.2 Results

The results of experiments are presented in Table 2.

Table 2. The results of comparison of MPI-GPUSVM with GPU and LibSVM for the different virtual network configurations

USPS		Training time(s)	Test time (s)	**Web**		Training time(s)	Test time(s)
	LibSVM	98,18	3,62		LibSVM	18916	75,00
	GPUSVM	7,95	**1,40**		GPUSVM	944,50	17,41
MPI	2P1H	19,54	19,54	MPI	2P1H	532,73	856,72
	4P1H	21,82	21,82		4P1H	576,80	2229,59
	1P2H	19,61	19,61		1P2H	526,82	18,75
	2P4H	22,15	22,15		2P4H	547,94	118,15
Single	3DP	3,77	3,77	Single	3DP	394,58	**8,10**
	5DP	**2,31**	2,31		5DP	**89,26**	17,87

Shuttle		Training time(s)	Test time (s)	**Letter**		Training time(s)	Test time (s)
	LibSVM	109,63	**1,43**		LibSVM	408,17	6,77
	GPUSVM	94,14	19,20		GPUSVM	19,74	7,62
MPI	2P1H	260,88	260,88	MPI	2P1H	136,29	7,47
	4P1H	174,31	174,31		4P1H	102,95	**6,42**
	1P2H	218,69	218,69		1P2H	1542,87	7,23
	2P4H	174,31	174,31		2P4H	104,13	6,32
Single	3DP	56,77	56,77	Single	3DP	36,81	8,06
	5DP	**24,86**	24,86		5DP	**14,09**	9,62

The following observations can be made on the basis of the experiments:

- In the case of the USPS the training time and classification one are much worse in the case of a network solution.
- For the Web dataset, even though the problem is binary, it gives the best results in parts of training for networked machines. Calculations speed is about 1.7 times

better than the implementation GPUSVM and even 10 times more favorable for single machines. This means that the size of a subset of the array is inversely proportional to efficiency gains. In the classification stage it is opposite and partitioning the data gives negative results.

- In the case of Shuttle simulates parallelism methods give a pretty good results for the training, but they do not keep pace with the original version of the GPUSVM during the test phase.
- In the case of Letter dataset MPI-GPUSVM handles the worst (even worse than LibSVM) on the training part on MPI environment with fewer nodes. For best results, the most partitioned version of the single machines. For classification stage network solution gives a slight better results than GPUSVM.

As expected, solutions based on graphic cards were much better than reference algorithm - LibSVM (Shuttle dataset was an exception). Classification time for LibSVM was much smaller in this case. This could mean that parallelization of algorithm adversely affects the calculation time for small datasets (such as Shuttle). It could be also noticed that solutions based on MPI protocol were worse in the training stage for smaller tasks. This is due to delays in communication between nodes. Despite these delays, MPI-based solution was best for longer tasks. For the classification stage, our algorithm performed better than or comparable to the time achieved by original method (GPUSVM).

5 Final Remarks

The appearance of the GPU as a unit of the highly paralleled grid of processors opens up all sorts of opportunities for acceleration in calculations and scaling learning algorithms. In addition, previous attempts to speed up the SVM by parallel processors and solutions based on classic clusters can be successfully transferred to cheaper and smaller graphic units or combined multiple GPUs. In such solutions the memory speed is great and communication is relatively faster compared to the environment based solely on network connections.

The experiments confirmed that the implementation of multiclass classifier running on a single unit of the GPU can lead to acceleration of the time solving problems compared to classical version of the algorithm. These results allowed us to reduce training time by order of magnitude, while maintaining the accuracy of the solution to the problem of classification. In case of using a specialized unit in parallel computing (such as Tesla) man can obtain a further increase in efficiency, reducing time to resolution of the problem with yet another order of magnitude. In addition, the work showed that even a simple virtual structure which parallelizes calculations by dividing and distributing the smaller subsets of data gave good acceleration for some datasets.

The described implementation NET-GPUSVM for MPI-communicated graphic cards should be running on a real network structure. Then the computing power of GPUs working in parallel would achieve significant performance gains while solving problems represented by used datasets.

Not only algorithms such as Support Vector Machines can be adapted to modern programming model. Other algorithms having components susceptible to parallelism of calculations could also benefit from such a modification [15].

Acknowledgment. This work is supported in part by The Polish National Science Centre under the grant which is being realized in years 2010-2013.

References

1. Bugres, C.J.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
2. Cao, L., Keerthi, S.: Parallel sequential minimal optimization for the training of support vector machines. IEEE Transactions on Neural Networks 17(4), 1039–1049 (2006)
3. Catanzano, B., Sundaram, N., Keutzer, K.: Fast support vector machine training and classification on graphics processors. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 104–111. ACM, New York (2008)
4. Alphaydin, E.: Introduction to Machine Learning, 2nd edn. The MIT Press, London (2011)
5. Cristianini, N.: Support Vector and Kernel Machines. Tutorial at ICML (2001)
6. Duan, K.-B., Keerthi, S.S.: Which Is the Best Multiclass SVM Method? An Empirical Study. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 278–285. Springer, Heidelberg (2005)
7. Graf, H., Cosatto, E.: Parallel support vector machines: The cascade svm. In: Advances in Neural Information Processing Systems, pp. 521–528. MIT Press (2005)
8. Herrero-Lopez, S., Williams, J.R., Sanchez, A.: Parallel Multiclass Classification using SVMs on GPUs, Pittsburgh, PA, USA (2010)
9. Joachims, T.: Making large-scale support vector machine learning practical. MIT Press, Cambridge (1999)
10. Lindholm, E., Nickolls, J., Oberman, S., Montrym, J.: Nvidia tesla: A unified graphics and computing architecture. IEEE Micro. 28(2), 39–55 (2008)
11. Nickolls, J., Buck, I., Garland, M., Skadron, K.: Scalable parallel programming with CUDA. Queue 6(2), 40–53 (2008)
12. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
13. Johnson, M.: SVM.NET Home Page, `http://www.matthewajohnson.org/`
14. Message Passing Interface Forum, `http://www.mpi-forum.org`
15. Wilk, T., Woźniak, M.: Complexity and Multithreaded Implementation Analysis of One Class-Classifiers Fuzzy Combiner. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS, vol. 6679, pp. 237–244. Springer, Heidelberg (2011)

Advanced CT and MR Image Processing with FPGA

Vladimir Kasik[1], Martin Cerny[1], Marek Penhaker[1], Václav Snášel[1], Vilem Novak[2], and Radka Pustkova[1]

[1] VSB - Technical University of Ostrava,
Faculty of Electrical Engineering and Computer Science,
Department of Cybernetics and Biomedical Engineering,
17. listopadu 15. 70833 Ostrava, Czech Republic
{vladimir.kasik,martin.cerny,
marek.penhaker,vaclav.snasel,radka.pustkova}@vsb.cz
[2] Faculty Hospital Ostrava, Child Neurology Clinic, 17. listopadu 1790/5, Ostrava,
Czech Republic
vilem.novak@fno.cz

Abstract. The CT and MRI image processing is a task with high requirement of computation time and computation speed. The calculation process in the most cases consists of various steps that are inherently parallel. There is opportunity to use parallel programming implemented in standard computer with more than one core. The more efficient way is to use real parallel signal processing. The only hardware that allows parallel data processing is FPGA. This article deals about this possibility. It is shown in the application on real problem in the image processing field. The first there is described the methodology of image processing in the application on CT (MRI) images of the head. The task is to count the ratio of intracranial fluid in the skull. It is important for child neurology diagnostic. The second part deals about the FPGA and its contribution to solving this task in the real parallel hardware system.

Keywords: FPGA, Image Processing, parallel process, calculation speed.

1 Introduction

The abnormal accumulation of cerebrospinal fluid (CSF) in the ventricles, or cavities, of the brain is diagnosed as hydrocephalus. This illness can cause progressive enlargement of the head, tunnel vision, and mental disability or in very serious cases the death. In infants and toddlers it causes enlargement of the head in adults it leads to brain tissue oppression and necrosis.[1]

The diagnosis of this disease is possible by monitoring the size of cerebral ventricles (see fig.1). This is made by the evaluation of that size and ratio of these ventricles in the skull. There is requirement to make this calculation automatically to we can compare the development of the ratio of intracranial fluid in the time progress. It is because of the automatic image processing is used. The proposed procedure allows the automatic assessment of cerebral ventricular volume in CT images.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 787–793, 2012.

The FPGA (field-programmable gate array) is an integrated circuit designed to be configured by the designer in the hardware layer of application. FPGA contains programmable logic components (gates), and allows to user to build his own logic structure of gates. It allows making original electronic device, which is intended for a special application. The FPGA allows building real hardware parallel systems. It is because it is good for use it to solve image processing tasks.[2]

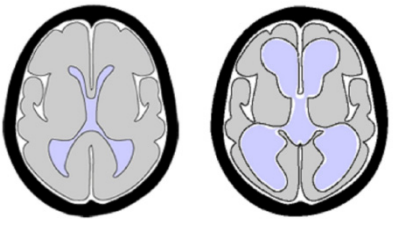


Fig. 1. Cerebral ventricles size (left) normal (right) enlarged [1]

The results presented in this article demonstrate the possibility of using FPGAs for image processing tasks and not only in presented application.

2 Methods

The designed algorithm can work with the images in DICOM and jpeg format. Even if the jpeg compression causes the image quality degradation – the color depth is on 8bit it could be used in our case.[3] The intracranial fluid differs at the images well from other tissues. Before DICOM files were processed the information about the measured person and every text were extracted due to it caused misdetections in further image processing.

The next step was the image crop. Only the cropped area was processed in the following steps. This modification reduces the computation time and it avoids problems during further image processing.

The edge is a place where there is a step change of brightness. The edge detection should be done by second derivation. This approach is very sensitive to the noise in the image. It was necessary to filter images. This work uses Gaussian filter, convolution mask which is defined by eq.1., where x and y are coordinates in the image and σ^2 is the mean square deviation (in this work $\sigma^2 = 1.4$).[4]

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

In the next step there was done brightness correction. Initial distribution of brightness in the image was adjusted by moving average method. The results see on figure 2.

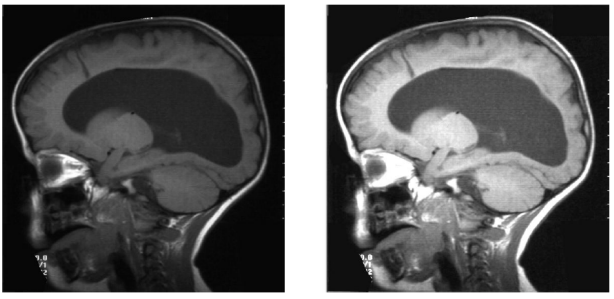


Fig. 2. Detail of brightness correction

The edge detection in the picture is the next step to we can recognize the intracranial fluid and other tissues. The Sobel mask with the dimension 5 x 5 was used and it was proofed experimentally as the suitable for this task. After edge detection it is necessary to control the results semi - manually and it is necessary to eliminate the structures which are not in the field of our interest. [5] The mask for intracranial fluid and the rest of tissues in the skull is done. The result of these steps you can see on figure 3.

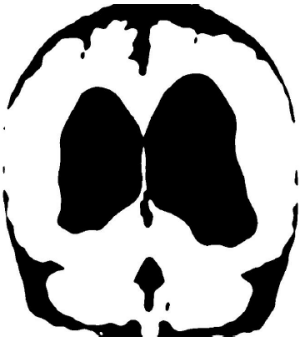


Fig. 3. The edge detection result

The calculation of the areas of interest is determined by the number of black points in the each mask. It is done by the algorithm defined in the equation 2.

$$S_L = \sum_{i=0}^{i}\sum_{j=0}^{j} x_{i,jPRK} + \sum_{i=0}^{i}\sum_{j=0}^{j} x_{i,jNPRK} \tag{2}$$

These points are overlapping each other, when comparing the two images. This information is important for the calculation rounded. Ranking data for cerebrospinal fluid, as is the scheme for overlapping and overlapping area is responsible for block comp. The procedure is applied to all data in the series.[4] [6]

3 FPGA Unit

The unit for acceleration of image processing is designed as a digital module in the FPGA. For this work has been used an ML403 board with a Virtex-4 FPGA. Image processing algorithms are generally solved by parallel methods of Digital Signal Processing, which use multiple adders and multipliers, so the resulting pixel is usually evaluated in single clock cycle.[7] The individual functional blocks are therefore designed as a separate hardware blocks (see Fig.4).

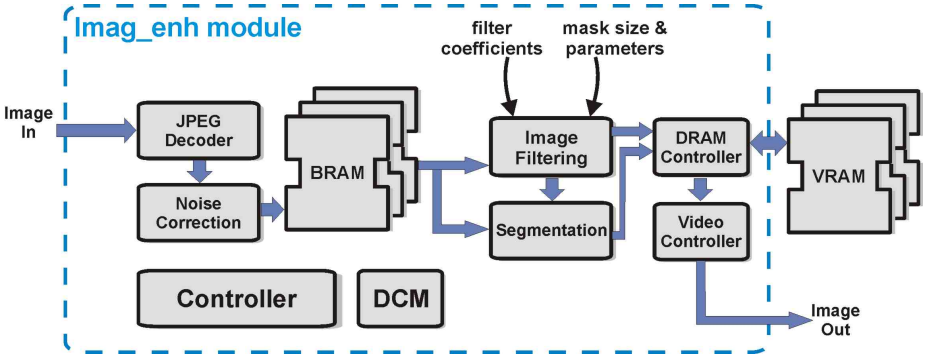


Fig. 4. The digital design module for image processing inside the FPGA

This process uses a significant amount of FPGA resources, in particular hardware multipliers. In the Fig. 4 the images enters the FPGA in JPG format, but the JPEG Decoder block can be replaced for different image formats reading from any data source. A block RAM inside the FPGA is used as a buffer for the just processed images. The memory has sufficiently short access time and can be configured in an appropriate type of organization. The image cropping and noise reduction is done before loading the image into the memory. The main part of the digital design is a digital image filtering block, which consists mainly of convolution filters and FIR filters. Another important block is a segmentation and edge enhancement block.

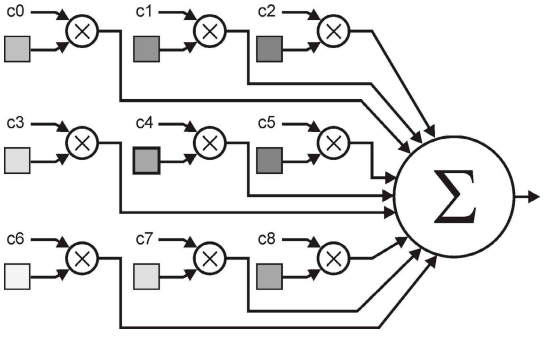
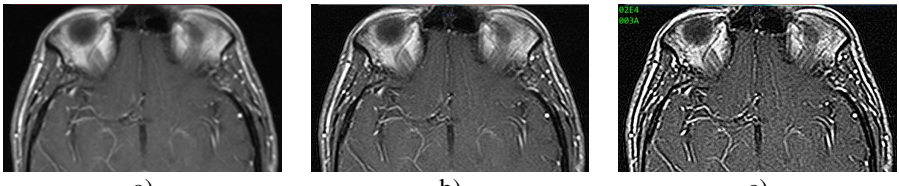


Fig. 5. The parallel structure of the 3x3 image convolution filter inside the FPGA

Both these blocks are configurable through the controller, which communicates with the host system via serial link. There can be set not only filter coefficients. We can also choose a few different configurations of filters that can be changed by changing the internal structure. Multipliers for DSP algorithms are used as the standard 18-bit, while adders can be created in any width of the fundamental elements of the FPGA logic. The method of processing data, processing speed and size of the resulting image is set by the controller, which is clocked with a clock signal from the Digital Clock Manager.



a) b) c)

Fig. 6. Part of the skull: a) source image b) sharpened image c) image with enhanced edges

Processed data are stored in a VideoRAM, which is designed as a DDR SDRAM chips outside the FPGA. The hardware solution of the filtration allows a significant acceleration of the image processing. This enables real-time image inspection including a surface evaluation of the picture segments. When configuring filter parameters there is used a host system, which allows to manually select the areas of evaluation using the mouse. Parameters to change the brightness of the image can be also set easily.

Table 1. Logic utilization of the Virtex-4 FPGA

Logic Utilization	Resources		
	Used	**Available**	**Utilization**
Number of Slices	1528	5472	28%
Number of Slice Flip Flops	2289	10944	21%
Number of 4 input LUTs	2628	10944	24%
Number of Block RAMs	32	36	89%

DCM frequency synthesizer block generates a clock frequency of 150 MHz, which allows processing of one image at a resolution of 800x600 pixels in units of milliseconds. Data path in Figure 1 represent buses with a width corresponding to the bit depth of the images. Currently, 12-bits data paths are used primarily because of the size of VRAM.

4 Results

This work has been done for Children neurology clinic of University Hospital of Ostrava. It was tested on the set of patients, children with hydrocephalus diagnosis. The

set of patients included patients which were not treated only surgically. The usability of image processing for this task was evidence. The good example is the results of surgically treated patient (see Fig. 7.)

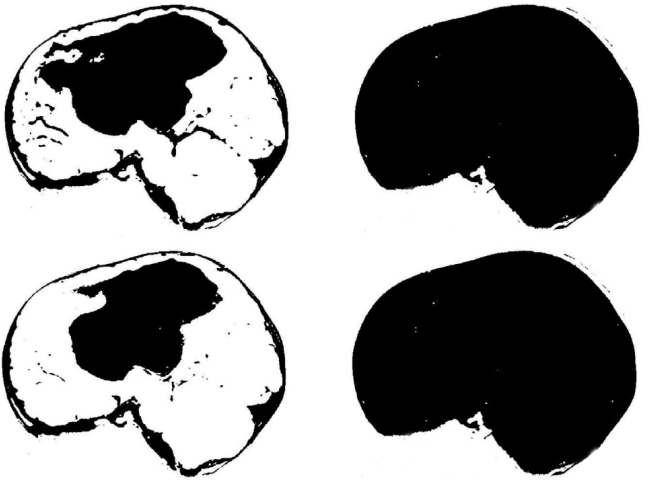


Fig. 7. The image processing results before (top) and after (bottom) surgery

5 Conclusion

This article deals about methodology for image processing in the application for diagnosis of the evaluation of hydrocephalus. The method was tested by physicians and it was identified as an effective tool for objective evaluation of disease development.

The using of modern hardware like FPGA for this type of tasks was described in this article too. It is clear, that parallel hardware using occurs to reduce the computational demands of processing in such complex tasks as that what was presented in this article.

Acknowledgement. The work and the contributions were supported by the project SV SP 2012/114 “Biomedical engineering systems VIII” and TACR TA01010632 “SCADA system for control and measurement of process in real time. The paper has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic. We also acknowledge support by student Frantisek Kutalek in development of the application testing.

References

1. Macak, J., Macakova, J.: Patologie, Prague (2004) ISBN 80-247-0785-03
2. Kasik, V.: FPGA based security system with remote control functions. In: 5th IFAC Workshop on Programmable Devices and Systems, Gliwice, pp. 277–280 (2002) ISBN:0-08-044081-9

3. Bernabucci, I., Conforto, S., Schmid, M., D'Alessio, T.: A bio-inspired controller of an upper arm model in a perturbed environment. In: Proceedings the 2007 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp. 549–553 (2007) ISBN: 978-1-4244-1501-4
4. Pustkova, R., Kutalek, F., Penhaker, M., Novak, V.: Measurement and calculation of cerebrospinal fluid in proportion to the skull. In: Proceedings - 9th RoEduNet IEEE International Conference, RoEduNet 2010, Romania, pp. 95–99 (2010)
5. Krejcar, O.: Large Multimedia Artifacts Prebuffering in Mobile Information Systems as Location Context Awareness. In: 4th International Symposium on Wireless Pervasive Computing, ISWPC 2009, pp. 216–220 (2009) ISBN 978-1-4244-4299-7, doi:10.1109/ISWPC.2009.4800591
6. Krištof, M., Hudak, R., Takacova, A., Zivcak, J., Fialka, L., Takač, R.: Contact pressure measurement in trunk orthoses. In: ICCC-CONTI 2010 - IEEE International Joint Conferences on Computational Cybernetics and Technical Informatics, p. 175 (2010) ISBN: 978-142447433-2, doi:10.1109/ICCCYB.2010.5491304
7. Krejcar, O.: Problem Solving of Low Data Throughput on Mobile Devices by Artefacts Prebuffering. EURASIP Journal on Wireless Communications and Networking (2009), doi:10.1155/2009/802523

New AdaBoost Algorithm Based on Interval-Valued Fuzzy Sets

Robert Burduk

Department of Systems and Computer Networks, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
robert.burduk@pwr.wroc.pl

Abstract. This paper presents a new extension of AdaBoost algorithm based on interval-valued fuzzy sets. This extension is for the weights used in samples of the training sets. The original weights are the real number from the interval [0, 1]. In our approach the weights are represented by the interval-valued fuzzy set, that is any weight has a lower and upper membership function. The same value of lower and upper membership function has a weight of the appropriate weak classifier. In our study we use the boosting by the reweighting method where each weak classifier is based on the recursive partitioning method. The described algorithm was tested on two generation data sets and two sets from UCI repository. The obtained results are compared with the original AdaBoost algorithm.

Keywords: AdaBoost algorithm, interval-valued fuzzy sets.

1 Introduction

Boosting is a machine learning effective method of producing a very accurate classification rule by combining a weak classifier [1]. The weak classifier is defined to be a classifier which is only slightly correlated with the true classification i.e. it can classify the object better than random classifier. In boosting, the weak classifier is learned on various training examples sampled from the original learning set. The sampling procedure is based on the weight of each example. In each iteration, the weights of examples are changing. The final decision of boosting algorithm is determined on the ensemble of classifiers derived from each iteration of the algorithm. One of the fundamental problems of the development of different boosting algorithms is choosing the weights and to define rules for an ensemble of classifiers. In recent years, many authors presented various concepts based on the boosting idea [2], [3], [4], [5]. There are also many studies showing the application of this method in the medical diagnosis problem [6]. In this article we present a new extension of AdaBoost [7] algorithm based on interval-valued fuzzy sets.

Since Zadeh introduced fuzzy sets in 1965 [8], many new approaches and theories treating imprecision and uncertainty have been proposed [9], [10]. Interval-valued fuzzy sets (IVFS) were proposed as a natural extension of fuzzy sets [11]. There are fuzzy sets in which the membership degree of each element of the fuzzy set is given by a closed subinterval of the interval $[0, 1]$. Various aspects of

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 794–801, 2012.

IVFSs have been considered for pattern recognition and decision making. In [12], [13] applications of IVFSs in pattern recognition are presented, IVFSs were also applied in image processing [14].

In this paper we consider a new extension of AdaBoost algorithm. This extension is for the weights used in samples of the training sets. The original weights are the real number from the interval [0, 1]. In our approach the weights are represented by the interval-valued fuzzy set. The same value of the lower and upper membership function has a weight of the appropriate weak classifier used in AdaBoost algorithm.

This paper is organized as follows: Section 2 introduces the necessary terms of IVFSs. In section 3 AdaBoost algorithm is presented. In the next section there is our modification of the presented algorithm. Section 4 presents the experiment results comparing AdaBoost with our modification. Finally, some conclusions are presented.

2 Interval-Valued Fuzzy Sets

An interval-valued fuzzy set $\bar{A}$ on a universe X is defined as [11]:

$$\bar{A} = \{\langle x, M_{\bar{A}}(x)\rangle : x \in X\} \tag{1}$$

where the function $M_{\bar{A}}(x) : X \to D[0,1]$ defines the degree of membership of an element $x \in X$ to $\bar{A}$. The $D[0,1]$ denotes the set of all closed subintervals of the interval $[0,1]$.

A more practical definitions for IVFS can be given as follows:

$$\bar{A} = \{(x, \overline{\mu}(x), \underline{\mu}(x))| \quad \forall x \in X\}, \tag{2}$$

where $\underline{\mu}(x) \leq \mu(x) \leq \overline{\mu}(x), \quad \mu \in [0,1]$.

The upper $\overline{\mu}(x)$ and lower $\underline{\mu}(x)$ membership functions define the footprint of uncertainty (FOU). The FOU is bounded from above by $\overline{\mu}(x)$ and from below by $\underline{\mu}(x)$. The $\overline{\mu}(x)$ and $\underline{\mu}(x)$ are fuzzy sets, which implies that we can use fuzzy set mathematics to characterize and work with IVFSs.

3 AdaBoost Algorithm

In the work [7] weak and strong learning algorithms were discussed. The weak algorithms can classify the object better than random, on the other hand strong algorithms can classify the object accurately. Schapire formulated the first algorithm to "boost" a weak classifier. The main idea in boosting is to improve the prediction of weak learning algorithms by creating a set of weak classifiers which is a single strong classifier. The well-known and widely applied is AdaBoost algorithm. Its main steps are as follows [15]:

1. Let $w_{1,1} = ... = w_{1,n} = 1/n$
2. For $t = 1, 2, ...T$ do:

a. Fit f_t using weights $w_{t,1}, ..., w_{t,n}$, and compute the error e_t
b. Compute $c_t = \ln((1 - e_t)/e_t)$.
c. Update the observations weights:

$$w_{t+1,i} = w_{t,i} \exp(c_t, I_{t,i}) / \sum_{j=1}^{n} (w_{t,i} \exp(c_t, I_{t,i})), \quad i = 1, ..n.$$

3. Output the final classifier:

$$\hat{y}_i = F(x_i) = sign(\sum_{t=1}^{T} c_t f_t(x_i)).$$

Table 1. Notation of the AdaBoost algorithm

i	Observation number, $i = 1, ..., n$.
t	Stage number, $t = 1, ..., T$.
x_i	A p-dimensional vector containing the quantitative variables of the ith observation.
y_i	A scalar quantity representing the class membership of the ith observation, $y_i = -1, 1$.
f_t	The weak classifier at the tth stage.
$f_t(x_i)$	The class estimate of the ith observation at the tth stage.
$w_{t,i}$	The weight of the ith observation at the tth stage, $\sum_i w_{t,i} = 1$.
$I_{t,i}$	Indicator function, $I(f_t(x_i) \neq y_i)$.
e_t	The classification error at the tth stage, $\sum_i w_{t,i} I_{t,i}$.
c_t	The weight of f_t.
sign(x)	$= 1$ if $x \geq 0$ and $= -1$ otherwise.

One of the main steps in the algorithm is to maintain a distribution of the training set using the weights. Initially, all weights of the training set observations are set equally. If an observation is incorrectly classified (at the current stage) the weight of this observation is increased. Similarly, the correctly classified observation receives less weight in the next step. For this reason the weak learner is forced to focus on the ard examples from the training set in each next step of the algorithm. In the each step of AdaBoost algorithm the best weak classifier according to the current distribution of the observation weights is found. The goodness of a weak classifier is measured by its error. Based on the value of this error the ratio is calculated. The final prediction of AdaBoost algorithm is a weighted majority vote of all weak classifiers.

4 AdaBoost Algorithm with Interval-Valued Fuzzy Weights

As we have previously described one of the main problems of the development of different boosting algorithms is the choice of weights. They concern the weights

of the observation $w_{t,i}$ and are needed to determine the weighted error e_t of each learned classifier. Without the loss of generality we can restrict $w_i(t)$ within the interval $[0, 1]$, where now the weight is otherwise expressed for the unity of the determinations. Now we define the $FOU_{e_t}(t)$ as uncertainty in the value of the classification error at the tth stage of the algorithm. The $FOU_{e_t}(t)$ should be constant for all steps of the algorithm or can be computed for each step. From this assumption, the weight is IVFS and can be represented as $\overline{\mu}_{w_i}(t)$ and $\underline{\mu}_{w_i}(t)$. The proposed AdaBoost algorithm with IVF weights is named IVFw-AdaBoost. Its main steps are as follows:

1. Enter the values: $FOU_{e_t}(t)$ $t = 1, 2, ...T$, and λ.
2. Let $w_1(1) = ... = w_n(1) = 1/n$.
3. For $t = 1, 2, ...T$ do:
a. Fit f_t using weights $w_{t,1}, ..., w_{t,n}$, and compute the error e_t.
b. Compute $\overline{\mu}_{e_t}(t) = e_t + |FOU_{e_t}(t)|/2$, $\underline{\mu}_{e_t}(t) = e_t - |FOU_{e_t}(t)|/2$.
c. Compute $\overline{\mu}_c(t) = \ln((1 - \overline{\mu}_{e_t}(t))/\overline{\mu}_{e_t}(t))$, $\underline{\mu}_c(t) = \ln((1 - \underline{\mu}_{e_t}(t))/\underline{\mu}_{e_t}(t))$.
d. Compute $c_t = \lambda * \overline{\mu}_c(t) + (1 - \lambda)\underline{\mu}_c(t)$
e. Update the observations weights:

$$\overline{\mu}_{w_i}(t+1) = w_i(t) \exp(\overline{\mu}_c(t), I_{t,i}) / \sum_{j=1}^{n} (w_i(t) \exp(\overline{\mu}_c(t), I_{t,i})), \quad i = 1, ..n,$$

$$\underline{\mu}_{w_i}(t+1) = w_i(t) \exp(\underline{\mu}_c(t), I_{t,i}) / \sum_{j=1}^{n} (w_i(t) \exp(\underline{\mu}_c(t), I_{t,i})), \quad i = 1, ..n.$$

f. Compute $w_i(t+1) = \lambda * \overline{\mu}_{w_i}(t+1) + (1 - \lambda)\underline{\mu}_{w_i}(t+1)$.
4. Output the final classifier:

$$\hat{y}_i = F(x_i) = sign(\sum_{t=1}^{T} c_t f_t(x_i)).$$

The parameter λ is needed for the deffuzyfication of weights. In this proposition it is a linear combination of lower and upper membership functions. For the λ greater than 0.5 the wrong classified observation is to get the greater weight in comparison to the original AdaBoost algorithm. For the λ less than 0.5 the wrong classified observation is to get the lower weight.

5 Experiments

To test IVFw-AdaBoost, we performed experiments on four data sets: the Banana set and the Highleyman set which were generated with the prtools Matlab toolbox, and two sets from UCI repository. There are the Pima Indians Diabetes and the Connectionist Bench (Sonar, Mines vs. Rocks) sets. Table 2 gives the properites of the data sets which we used in experiments.

Table 2. Properites of the data sets used in experiments

Data sets	Features	Classes	Observations
Banana	2	2	400
Highleyman	2	2	400
Sonar	60	2	208
Pima	8	2	768

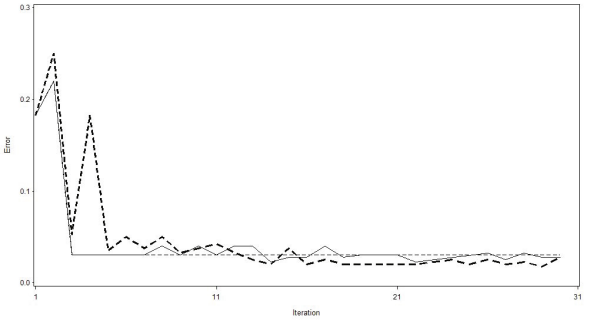


Fig. 1. The error for the Banana set

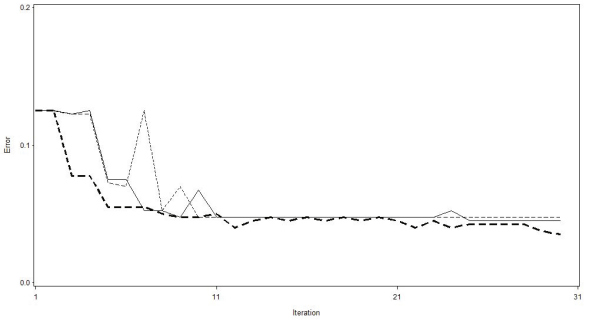


Fig. 2. The error for the Highleyman set

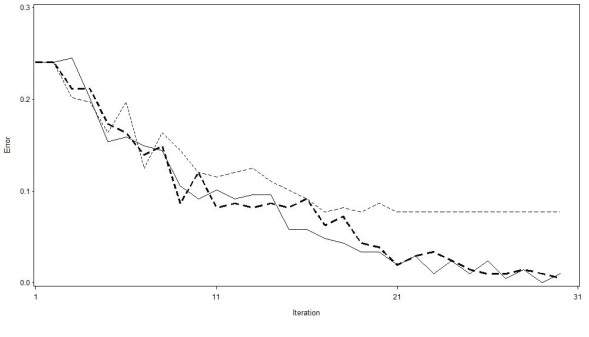


Fig. 3. The error for the Sonar set

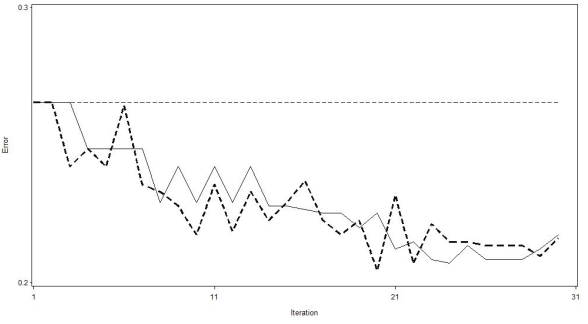


Fig. 4. The error for the Pima set

Results for the thirty iterations of the algorithm are shown in Fig. 1-4 and the numerical values are presented in Tab. 3-4.

The results for AdaBoost algorithm are represented by solid lines. The results for IVFw-AdaBoost algorithms with $\lambda = 1$ are represented by the thin dashed line and with $\lambda = 0$ are represented by the thick dashed line. Three out of four (Highleyman, Sonar and Pima) data sets obtained better results in comparison with AdaBoost algorithm (if we take into account only the thirtieth iteration of the algorithm). Generally, for the parameter $\lambda = 0$ we often obtain lesser value of error compared to AdaBoost algorithm.

Table 3. The results of experiments - misclassification error for Banana and Highleyman sets

Iteration	Banana set			Highleyman set		
	AdaBoost	IVFw-AB $\lambda = 0$	IVFw-AB $\lambda = 1$	AdaBoost	IVFw-AB $\lambda = 0$	IVFw-AB $\lambda = 1$
1	0.1825	0.1825	0.1825	0.1250	0.1250	0.1250
2	0.2200	0.2500	0.2200	0.1250	0.1250	0.1250
3	0.0300	0.0525	0.0300	0.1225	0.0775	0.1225
4	0.0300	0.1825	0.0300	0.1250	0.0775	0.1225
5	0.0300	0.0350	0.0300	0.0750	0.0550	0.0725
6	0.0300	0.0500	0.0300	0.0750	0.0550	0.0700
7	0.0300	0.0375	0.0300	0.0525	0.0550	0.1250
8	0.0400	0.0500	0.0300	0.0525	0.0500	0.0525
9	0.0300	0.0325	0.0300	0.0475	0.0475	0.0700
10	0.0400	0.0375	0.0300	0.0675	0.0475	0.0475
15	0.0275	0.0375	0.0300	0.0475	0.0450	0.0475
20	0.0300	0.0200	0.0300	0.0475	0.0475	0.0475
25	0.0300	0.0200	0.0300	0.0450	0.0425	0.0475
30	0.0275	0.0275	0.0300	0.0450	0.0350	0.0475

Table 4. The results of experiments - misclassification error for Sonar and Pima sets

Iteration	Sonar set			Pima set		
	AdaBoost	IVFw-AB $\lambda = 0$	IVFw-AB $\lambda = 1$	AdaBoost	IVFw-AB $\lambda = 0$	IVFw-AB $\lambda = 1$
1	0.24038	0.24038	0.24038	0.26563	0.26563	0.26563
2	0.24038	0.24038	0.24038	0.26563	0.26563	0.26563
3	0.24519	0.21154	0.20192	0.26563	0.24219	0.26563
4	0.20192	0.21154	0.19712	0.24870	0.24870	0.26563
5	0.15385	0.17308	0.16346	0.24870	0.24219	0.26563
6	0.15865	0.16346	0.19712	0.24870	0.26432	0.26563
7	0.14904	0.13942	0.12500	0.24870	0.23568	0.26563
8	0.14423	0.14904	0.16346	0.22917	0.23307	0.26563
9	0.10577	0.08654	0.14423	0.24219	0.22786	0.26563
10	0.09135	0.12019	0.12019	0.22917	0.21745	0.26563
15	0.05769	0.08173	0.10096	0.22786	0.22917	0.26563
20	0.03365	0.03846	0.08654	0.22526	0.20443	0.26563
25	0.00962	0.01442	0.07692	0.21354	0.21484	0.26563
30	0.00962	0.00481	0.07692	0.21745	0.21615	0.26563

6 Conclusions

In this paper we presented new IVFw-AdaBoost algorithm. It is a modification of AdaBoost algorithm where we use IVFSs. We consider the situation where the weights are represented by IVFS. For the deffuzyfication of weights we propose the linear combination of lower and upper membership functions of weights. In our study we use boosting by the reweighting method where each weak classifier is based on the recursive partitioning method. The received results for four data sets show that thanks to the choice of the parameter lambda we can get better results of the final classification measured by the value of the error.

In future work we can apply the $FOU_{e_t}(t)$ which can be computed on each step. Additionally, you can check the behavior of the proposed algorithm in the multistage classifier. In particular, you should discuss the choice of the parameter λ in different nodes of the decision tree of the multistage classifier.

Acknowledgments. This work is supported in part by the National Science Centre under the grant which is realizing in years 2011–2014.

References

1. Kearns, M., Valiant, L.: Cryptographic limitations on learning boolean formulae and finite automata. J. Assoc. Comput. Mach. 41(1), 67–95 (1994)
2. Chunhua, S., Hanxi, L.: On the Dual Formulation of Boosting Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(12), 2216–2231 (2010)
3. Oza, N.C.: Boosting with Averaged Weight Vectors. In: Windeatt, T., Roli, F. (eds.) MCS 2003. LNCS, vol. 2709, pp. 15–24. Springer, Heidelberg (2003)

4. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning, Bari, Italy, pp. 148–156 (1996)
5. Wozniak, M.: Proposition of Boosting Algorithm for Probabilistic Decision Support System. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004, Part I. LNCS, vol. 3036, pp. 675–678. Springer, Heidelberg (2004)
6. Wozniak, M.: Boosted Decision Trees for Diagnosis Type of Hypertension. In: Oliveira, J.L., Maojo, V., Martín-Sánchez, F., Pereira, A.S. (eds.) ISBMDA 2005. LNCS (LNBI), vol. 3745, pp. 223–230. Springer, Heidelberg (2005)
7. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boostin. Journal of Computer and System Scienses 55(1), 119–139 (1997)
8. Zadeh, L.A.: Probability measures of fuzzy events. Journal of Mathematical Analysis and Applications 23, 421–427 (1968)
9. Goguen, J.: L-fuzzy sets. Journal of Mathematical Analysis and Applications 18(1), 145–174 (1967)
10. Pawlak, Z.: Rough sets and fuzzy sets. Fuzzy Sets and Systems 17, 99–102 (1985)
11. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning - I. Information Science 8, 199–249 (1975)
12. Burduk, R.: Imprecise information in Bayes classifier. Pattern Analysis and Applications 15(2), 147–153 (2012)
13. Mitchell, H.B.: Pattern recognition using type-II fuzzy sets. Information Science 170, 409–418 (2005)
14. Melin, P.: Image Processing and Pattern Recognition with Mamdani Interval Type-2 Fuzzy Inference Systems. In: Trillas, E., Bonissone, P.P., Magdalena, L., Kacprzyk, J. (eds.) Combining Experimentation and Theory. STUDFUZZ, vol. 271, pp. 179–190. Springer, Heidelberg (2011)
15. Dmitrienko, A., Chuang-Stein, C.: Pharmaceutical Statistics Using SAS: A Practical Guide. SAS Press (2007)

Application of an Hybrid Bio-inspired Meta-heuristic in the Optimization of Two-Dimensional Guillotine Cutting in an Glass Industry

Flávio Moreira da Costa and Renato José Sassi

Industrial Engineering Post Graduation Program, Nove de Julho University, Sao Paulo, Brazil
flavio.costa@iduo.com.br, renato.sassi@ieee.org

Abstract. The optimization of two-dimensional guillotined cutting consists in determining a parts arrangement to be cut from a larger piece, maximizing the material use, but respecting the restrictions imposed by the cutting equipment and the production flow. An optimized cutting process maximizes the materials use and is an important factor for production systems performance at glassworks industries, impacting directly in the products final cost formation and, thus, increasing the company's competitiveness in glass market. Several studies have shown that combinations of bio-inspired meta-heuristics, more specifically, the Genetic Algorithms (GA) and Ant Colony Optimization (ACO) are efficient techniques to solving constraint satisfaction problems and combinatorial optimization problems. GA and ACO are bio-inspired meta-heuristics techniques suitable for random guided solutions in problems with large search spaces. GA are search methods inspired by the natural evolution theory, presenting good results in global searches. ACO is based on the attraction of ants by pheromone trails while searching for food and uses a feedback system that enables rapid convergence in good solutions. The initial results from the combination of these two techniques when compared with the results each technique individually applied are encouraging and have presented interesting solutions to the problem of optimizing two-dimensional guillotined cutting.

Keywords: Genetic Algorithms, Ant Colony Optimization, Two-dimensional Cutting.

1 Introduction

The resource allocation problems have been a great impact factor on the industrial production systems performance and can be found in various industrial or production processes, such as task allocation, packaging problems in logistics processes and material cutting processes.

Due to practical applications potential to optimize industrial processes and the difficulties to obtain exact solutions, this problems category has been the subject

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 802–809, 2012.

of intenses research in the Operational Research (OR) and Artificial Intelligence (AI).The study of these problems provides a common basis for analysis and solution of other problems that belong to the same category [5] and, in this context, the bio-inspired meta-heuristics techniques is being increasingly used in solving such problems.

Genetic Algorithms (GA) and Ant Colony Optimization (ACO) techniques are bio-inspired meta-heuristics suitable for random guided solutions in large search spaces problems, due to difficulties of deterministic methods in such problems types [11].

GA are nondeterministic techniques inspired by natural evolution theory and known as robust and efficient methods of searching for irregular, multidimensional and complex search spaces [10].

ACO is based on the attraction of ants by pheromone trails while searching for food [6] and uses a feedback system that enables rapid convergence to good solu-tions [11].

The Hybrid architectures, has been widely researched and applied in solving combinatorial optimization problems [3], [9], [11], [13], [14], [15], [16].

Besides this introduction the paper is organized as follows: section 2 describes the the two-dimensional cutting problem, section 3 presents the hybrid bio-inspired meta-heuristics approach used in this research, in section 4 we discuss the methodology, sections 5 and 6 dealing, respectively, on the initial results from this research and the conclusions.

2 Two-Dimensional Guillotine Cutting Problem

The two-dimensional cutting is a combinatorial optimization problem which consists in determining a cutting pattern of material pieces, so as to produce a set of smaller pieces, satisfying certain constraints.

The main limitations of two-dimensional cutting process in an glass industry are the size of the glass plates to be cut and such cuts have to be guillotined, where possible, to maximize the plates used in the process. The Fig. 1 illustrates two-dimensional guillotine cutting process.

The two-dimensional cutting is a complex problem, which the time required to find a solution takes a factorial increases in relation to the increased amount of pieces to be cut from the same plate.

With the increasing complexity, it is impractical to calculate the exact solution (deterministic) for this problem, since the time required to solve it becomes unacceptable to the requirements of the solution.

3 An Hybrid Bio-inspired Meta-Heuristic

The great advantage of hybrid systems is due to synergy produced by combining a two or more techniques. This synergy reflects in obtaining a more powerful and less disability system [14].

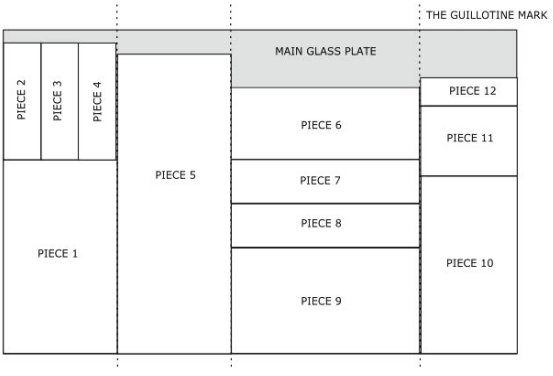


Fig. 1. The two-dimensional guillotine cutting

In a similar study, which combines the search capabilities of the two meta-heuristics (GA and ACO) for solving another combinatorial optimization problem (Travelling Salesman Problem), the author uses a cooperative hybrid approach [3], where :

1. whenever the ACO cannot leave from a local optimum point, it exports a predetermined amount of solutions to create the initial population of GA;
2. the ACO iterations are interrupted;
3. the GA starts to find a better solution than the ACO;
4. since the AG found this solution, the ACO algorithm updates the pheromone on that basis, reinforcing the search alternatives for the next iteration of ACO.

This article approaches the combination of bio-inspired meta-heuristics techniques, known as Genetic Algorithms and Ant Colony Optimization, to solve the two-dimensional guillotine cutting problem. This hybrid algorithm, resulting from the combination of Genetic Algorithms and Ant Colony Optimization, has been called GA + ACO.

4 The Packing Algorithm

The Assessment of GA individuals is performed with the aid of the packing algorithm called First Fit Decreasing Width Decreasing Height (FFDWDH). The calculation consists in fit the pieces in master blade, simulating the cutting process and discovering the unused area (waste).

For FFDWDH algorithm implementation:

1. the pieces are sorted in decreasing order by width and height then, as can be seen in Fig. 2.
2. then the pieces are allocated from left to right within vertical stripes (columns).

3. attempts are made to insert the part in the first available column of glass slide, if it does not fit, then tries to insert it in the next column and so on, until the method finds some column with sufficient space to allocate it.
4. if the piece does not fit into any existing column, then creates a new column.

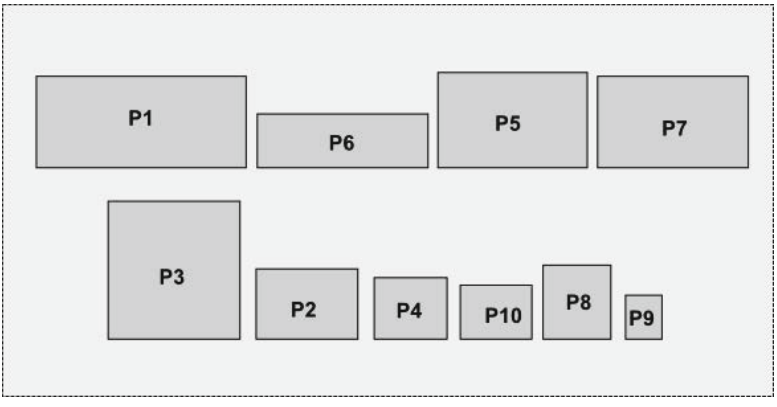


Fig. 2. Pieces sorted in decreasing order by width and height, when. (adapted from [17]).

As the parts are pre-sorted before insertion, the width of the columns is determined by the first part to be inserted. Below can be seen an example of pseudocode of FFDWDH algorithm and in Fig. 3, the outcome of algorithm applied.

Generic pseudocode of FFDWDH algorithm.

```
procedure FFDWDH()
   var
     i,j : Real;
     W : Real; {sheet width}
     H : Real; {available sheet heigth}
     hi : Real; {piece i heigth}
     wi : Real; {piece i width}
     wdlj : Real; {column j available width}
     hdlj : Real; {column j available heigth}
   begin
     wdlj := wi;
     hdlj := H;
     j := 1;
     for i:= 1 to n do
     begin
        while (wi > wdlj) or (hi > hdlj) do
        begin
```

```
            j := j+1;
        end;
        wdlj := wdlj-wi;  {include i piece  in column j}
        hdlj := hdlj-hi
    end;
end.
```

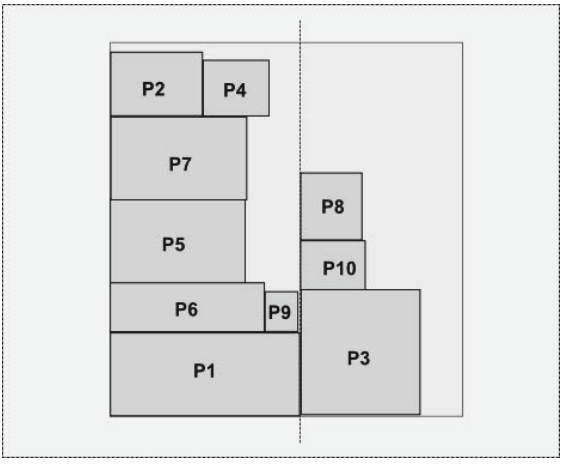


Fig. 3. A example result of FFDWDH algorithm applied

5 Methodology

The development of this paper is based on the study, revision and implementation of bio-inspired computing techniques for solving combinatorial optimization problems. It was made a literature review of two-dimensional guillotine cutting problems derived from production systems, as well as the options for combining Genetic Algorithms and Ant Colony Optimization.

To implement the hybrid algorithm (GA+ACO) are used the programming languages Java and Java Script.

Are being performed several experiments using a data set of requests from a glass industry in São Paulo city. During these experiments, the results of hybrid algorithm (GA+ACO) are compared with the results of each individual algorithm (GA and ACO).

6 The Initial Results of the Experiments

Experiments are still being tested several different variations in the algorithms parameters, as the chromosomes quantity and generations number of the GA and

the artificial ants quantity for the ACO, in order to find the best combination for each presented data sets.

The Fig. 4 shows a processing time comparison between individual algorithms (AG, ACO) and hybrid algorithm (AG+ACO) applied to the same data set, from different amounts of pieces (in the hybrid algorithm, the amount of iterations of the GA and ACO has been reduced for the half, without evidence of loss in quality of results). The initial results from the Genetic Algorithms and Ant Colony Optimization combination, taking as input the same data sets, present solutions faster than those presented by the application of algorithms individually. The Fig. 5 shows the output of hybrid algorithm applied to a small data set (32 pieces).

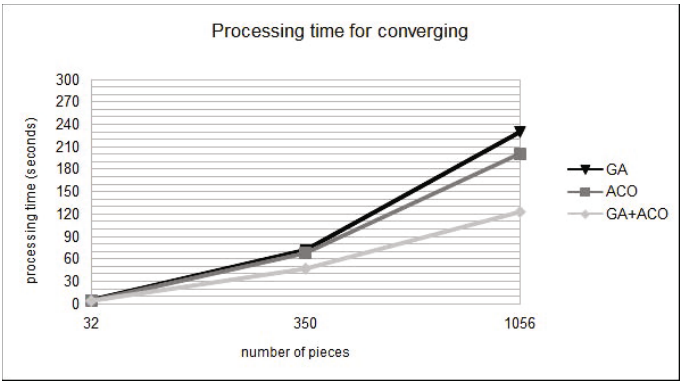


Fig. 4. A processing time comparison between the hybrid algorithm and individual ones

Fig. 5. Output of the hybrid algorithm applied to a two-dimensional guillotine cutting problem

7 Conclusions

An optimized cutting process maximizes the materials use and is an important factor for production systems performance at glassworks industries, impacting directly in the products final cost formation and, thus, increasing the company's competitiveness in glass market.

Initial results from combination of GA+ACO are encouraging and shown that hybrid bio-inspired meta-heuristics are efficient techniques for solving two-dimensional guillotine cutting problems and, consequently, for resolving similar combinatorial optimization problems.

The purpose is to continue the research by changing the parameters of the GA+ACO and experimenting with other different ways of combination between these algorithms, as, for exemple, using the ACO in the AG crossover process.

Acknowledgments. This work was supported by the Nove de Julho University.

References

1. Blum, C., Ibáñez, M.L.: The Industrial Electronics Handbook, Intelligent systems, 2nd edn. CRC Press (2011)
2. Canto, N., Sassi, R.J., Costa, F.M.: Aplicação de um algoritmo genético para solução do problema do corte bidimensional em lâminas de vidro. In 5th Conferência Ibérica de Sistemas e Tecnologias de Informação, CISTI 2010, vol. 1, pp. 1–4 (2010)
3. Carvalho, M.B.: Aplicações de Meta-heurística Genética e Fuzzy no Sistema de Colônia de Formigas para o Problema do Caixeiro Viajante. Dissertação de Mestrado em Engenharia Elétrica, UNICAMP (2007)
4. de Castro, L.N., von Zuben, F.J.: Recent Developments in Biologically Inspired Computing. Idea Group Publishing (2005)
5. Costa, F.M., Canto, N., Sassi, R.J.: Study of the Application of Genetic Algorithms in Optimization of Cutting Glass Sheets. In: Proceedings of the 9th IEEE/IAS International Conference on Industry Applications, INDUSCON, São Paulo, vol. 1, pp. 1–3 (2010)
6. Dorigo, M., Stutzle, T.: Ant Colony Optimization. Bradford Book (2004)
7. Dyckhoff, H.: A typology of cutting and packing problems. European Journal of Operational Research 44(2), 145–159 (1990)
8. Goldgerg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, MA (1989)
9. Guangdong, H., Ping, L., Qun, W.: A Hybrid Metaheuristic ACO-GA with an Application in Sports Competition Scheduling. In: 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 55–57 (2007)
10. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)
11. Hoseini, P., Shayesteh, M.G.: Hybrid Ant Colony Optimization, Genetic Algorithm, and Simulated Annealing for Image Contrast Enhancement. In: 2010 IEEE Congress on Evolutionary Computation, CEC, pp. 1–6 (2010)

12. Linden, R.: Algoritmos Genéticos: Aplicação em Bioinformática e no setor elétrico, programação Genética. Estratégias Evolucionárias e Algoritmos Meméticos. Brasport (2008)
13. Liu, B., Meng, P.: Hybrid Algorithm Combining Ant Colony Algorithm with Genetic Algorithm for Continuous Domain. In: Proceedings of the 9th International Conference for Young Computer Scientists, ICYCS, pp. 1819–1824 (2008)
14. Sassi, R.J.: Uma Arquitetura Hbrida para Descoberta de Conhecimento em Bases de Dados: Teoria dos Rough Sets e Redes Neurais Artificiais Mapas Auto-Organizáveis. Tese de Doutorado, Universidade de São Paulo (2006)
15. Zhang, D., Du, L.: Hybrid Ant Colony Optimization Based on Genetic Algorithm for Container Loading Problem. In: International Conference of Soft Computing and Pattern Recognition, SoCPaR, pp. 10–14 (2011)
16. Grosan, C., Abraham, A.: Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews. SCI, vol. 75, pp. 1–17. Springer, Heidelberg (2007)
17. Temponi, E.C.: Uma Proposta de Resolução do Problema de Corte Bidimensional via Abordagem Metaheurística. Dissertação de Mestrado. Diretoria de Pesquisa e Pós-Graduação, CEFET-MG (2007)

Photovoltaic Power Plant Output Estimation by Neural Networks and Fuzzy Inference

Lukáš Prokop[1], Stanislav Mišák[1], Tomáš Novosád[1], Pavel Krömer[1,2], Jan Platoš[1,2], and Václav Snášel[1,2]

[1] Dept. of Computer Science, VŠB-Technical University of Ostrava, Czech Republic
{lukas.prokop,stanislav.misak,tomas.novosad,pavel.kromer, jan.platos,vaclav.snasel}@vsb.cz
[2] IT4Innovations, Ostrava, Czech Republic

Abstract. The stochastic production of renewable energy sources puts increased demands on power grids worldwide. Neurocomputing methods can be used for the forecast of electric energy production of renewable resources and contribute to the reliability of energy systems. This study compares two neurocomputing methods as predictors of a selected photovoltaic power plant in the Czech Republic that meets the real world criterion of high output variance and relatively large installed power.

Keywords: prediction, photovoltaic power plant, neural networks, adaptive network-based fuzzy inference systems.

1 Introduction

The ever increasing installed power output of power sources providing stochastic supply as well as the increasing share of distributed electric power puts greater demands on the preparation of the relevant operations and management of distribution networks. The influence of each type of production units varies depending on the type of the unit as well as on the volume of the installed power output provided by the monitored power system.

Prediction of the production of electric power is important for the preparation and management of distribution networks. Moreover, the results provided by predictive models can be used as one of the controlling variables to control off-grid power units, which use photovoltaic technology for the production of electric power. Photovoltaic power plants offer stochastic supply of electricity and deterministic prediction methods often do not provide sufficient accuracy. Therefore, we shall utilize soft computing methods that are able to predict the production of electricity within short time intervals, even for production units offering stochastic production of electricity.

Evaluation and initial applications of methods able to predict the stochastic production may be found in [7]. The study used for the prediction of the production of electricity created by photovoltaic power plant fuzzy logic and fuzzy classifiers. Artificial neural networks were used for short-term and mid-term solar power prediction in [4,1]. Gaussian equations were tested and applied in [11]

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 810–817, 2012.

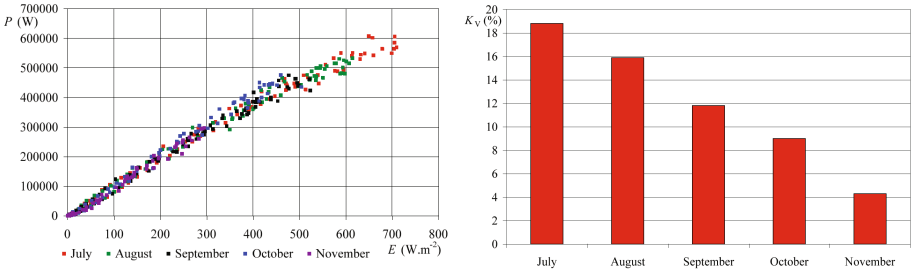
to estimate solar power output. In [10] was shown a model that relates short-circuit current with radiation, considering the true nonlinear behavior of the relationship between variables. A prediction method based on this relationship and genetic programming was proposed. Statistical short-term forecasting system for grid-connected photovoltaic plant was presented in [3]. This system was based on combination of three modules, two numerical weather prediction models and an artificial neural network based model. Application of hybrid multilayer feed forward neural network technique for photovoltaic power plant output estimation was presented in [13]. A practical method for solar irradiance prediction was shown in [8]. The study utilized a multilayer perceptron neural network for a 24-hour forecast of solar irradiance based on actual values of mean daily solar irradiance and air temperature.

The above mentioned techniques were able to predict the production of electric power, but they are based on the optimization of prediction techniques and procedures developed specifically for given areas or power plants. Currently, there is no such predictive system localized or optimised for the Czech Republic. However, localized and optimized general prediction methods may be used. To test and verify individual prediction techniques we have selected a photovoltaic power plant (PVPP) with capacity/power output of 1100 kWp. The PVPP was equipped with monocrystalline silicon panels capable of using only the direct component of solar radiation. The measured value of the supplied electrical power and the intensity of the solar radiation are used not only to test the prediction methods, but are also used to determine basic parameters of any photovoltaic power plant. Figure 1a shows the power output curve for a selected period of about five months and we can see the availability of the electrical power produced during various meteorological / weather conditions. The shape of the curve shows the differences in the amount of the supplied power during summer and winter time.

Another important parameter which characterizes the operation of a photovoltaic power plant is the so-called capacity factor of a power plant, which is defined as the ratio between the actual amount of generated electricity and the amount of electric energy that would have been generated if the photovoltaic plant supplied power continuously. Values shown in fig. 1b demonstrate a considerable difference between the amount of produced electricity in July and the amount of electricity produced in November. It is very likely that this difference will have an impact on the error in the prediction while using different methods, whereas it is expected that prediction methods will show larger variations for winter period.

2 Multilayer Perceptron

Artificial neural networks (ANNs) constitute a family of computing models based on the emulation of electrochemical processes observed in neural systems of living organisms [6,2].



(a) Power plant power curve (b) Capacity factor

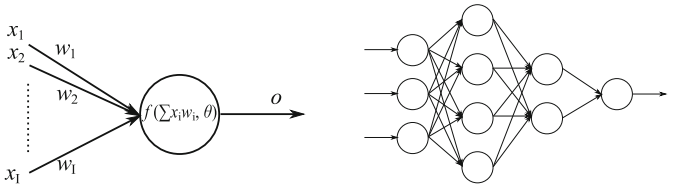
Fig. 1. Power output curve and capacity factor

Various types of complex general-purpose artificial neural networks composed of simple computing units - artificial neurons - have been proposed. Artificial neurons emulate the behavior of biological neurons in terms of signal processing (aggregation, thresholding, modification, propagation etc.) and information storage (input weights, activation function parameters). A single artificial neuron (perceptron) represents a non-linear mapping $f : \mathbb{R}^I \rightarrow \mathbb{R}$ and it can be used to solve linearly separable problems. A schematic view of a perceptron is shown in fig. 2a. In the figure, x_i represents input signal, w_i input connection weights, $f(\sum x_i w_i, \theta)$ activation function and o the output signal. The perceptron in fig. 2a is a summation unit (i.e. it performs a weighted sum of input signals) and it can be used to solve linearly separable problems. The perceptron must be set-up (trained) before it can be used to process data. The training involves setting the input connection weights w_i and activation function parameters.

An ANN is a layered network of artificial neurons with certain topology [2]. It implements a non-linear mapping $f_{ann} : \mathbb{R}^I \rightarrow \mathbb{R}^K$ from I-dimensional input space to K-dimensional output space [2]. In contrast to single perceptron, ANN is able to solve problems that are not linearly separable and in general to provide an approximation of a function. ANNs have been used for countless applications in data mining, pattern recognition, data classification, control and so fort. Multilayer perceptron is a basic type of multilayer feed-forward ANN [2] that consists of multiple fully-connected layers of perceptrons. The MLP consists of the input layer, one or more hidden layers, and the output layer (see fig. 2b).

The ANN training methods include supervised, unsupervised, and reinforced learning. The well-known backpropagation (BP) algorithm is an example of supervised learning method based on gradient descent optimization of network parameters [2]. The BP training algorithm consists of a number of iterations in which the network (that was originally randomly initialized) first processes training patterns and computes the error for each pattern and second performs

the backward propagation of error in which the weights are adjusted as a function of the error signal. The training is finished when the terminating criteria (i.e. the maximum number of iterations was reached, training error was small enough) are met. Other parameters of the BP algorithm include the learning rate which is the size of each learning step and momentum that controls how the network avoids fluctuations when processing different training patterns during stochastic learning. For more details on the MLPs and the BP algorithm see e.g. [2,6].



(a) A perceptron

(b) An example of a 3-4-2-1 MLP

Fig. 2. Perceptron and multilayer perceptron (feed-forward network)

3 Adaptive Network-Based Fuzzy Inference System

ANFIS (Adaptive Network-based Fuzzy Inference System) is fuzzy inference system which uses the design and features of multilayer feed-forward ANNs proposed by Jyh-Shing Roger Jang in 1993 [5]. Fuzzy inference systems, sometimes referred to as fuzzy rule-based systems, fuzzy models, fuzzy associative memories (FAM), or fuzzy controllers (when used as controllers), are special type of fuzzy modeling or fuzzy identification systems, first proposed by Takagi and Sugeno [14]. Fuzzy modeling systems have various practical applications in approximation, prediction, detection, control, inference and more [12,9]. Common fuzzy inference systems are composed of five basic functional parts as shown in fig. 3 [14]. The fuzzy inference system compares in the fuzzyfication step the input variables with the membership functions on premise part to gain the fuzzy membership values of each input. The membership values are then combined

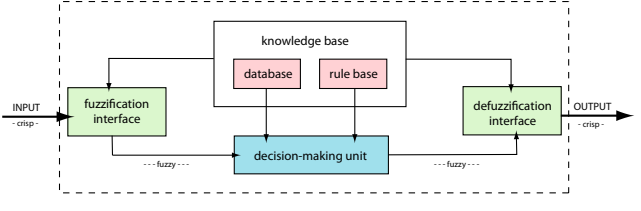


Fig. 3. Fuzzy Inference System

on the premise part by a specific t-norm operator (minimum or multiplication). This assigns a firing strength (weight) to each rule. For each rule is then generated the qualified output on consequent part. Finally, the qualified outputs are combined to produce a crisp output in the defuzzyfication step.

ANFIS is a fuzzy inference systems based on an a simple multilayer feed-forward ANN. Each node of this network applies a specific function (activation function) on incoming signals with activation function parameters which belong to this node. For further reference we call this type of node a flexible node. Every such node can have a different activation function with different parameters. There are also fixed nodes in the adaptive network which only transforms incoming signals without any parameters.

ANFIS can be seen as the multilayer feed-forward ANN which satisfies the conditions of fuzzy inference system mentioned above and is composed of five layers [14]. Every node in the *first layer* is a flexible node with an arbitrary continuous and piecewise differentiable activation function selected depending on the data. For example, activation functions can be Gaussian type functions, triangular function, trapezoidal function etc. The *second layer* is composed of fixed nodes labeled $\prod$ when the input signals are multiplied (i.e. multiplication units) or min when is taken the minimum of the input signal values. The product or minimum of input signal values is then propagated as the output to the next layer. The *third layer* consists of fixed nodes labeled N - normalization nodes. The ith node in this layer normalizes the output of ith node in second layer (ith fuzzy rule node firing strength) against the sum of all node outputs of layer 2. This layer is commonly called normalization layer. *Fourth layer* consists of flexible nodes whose activation function parameters are tuned during the ANFIS learning phase. The overall output of the adaptive network is computed in the *fifth layer* composed of a single fixed node $\sum$, which sums all incoming signals from previous layer.

The parameter set of an adaptive network is the union of the parameter of each adaptive (flexible) node. In order to achieve a desired input-output mapping, these parameters must be updated according to given training data. The batch (off-line) learning and pattern (on-line) learning can be used to setup the adaptive network. The batch learning procedure updates parameters after all of input-output learning pairs have been presented and the pattern learning procedure updates adaptive node parameters after each input-output learning pair has been proceed. ANFIS adaptive node parameters are commonly updated with hybrid learning algorithm which compose of least square estimate method (LSE) and back-propagation. Least square estimate algorithm is used for optimizing the consequent parameters and back-propagation is used for optimization of premise parameters. In many cases there is no need for hybrid learning algorithm and only consequent parameters are updated using LSE method, and the premise parameters are set according to training input data. For detailed information see [5,14].

4 Experiments

We have used a MLP and ANFIS for the estimation of power output of a photovolatic power plant located in the Czech Republic. We have recorded the volumes of electrical energy produced by the PVPP and the values of solar radiation in the same location. The values were recorded in 10 minute intervals between November 2010 and April 2011. The data set contained after cleaning 20513 records that were divided in two halves. The first part containing 10257 records was used as the training data set for predictor evolution. The second part containing 10256 records was used as testing data set. The parameters used for the training of the MLP and ANFIS are shown in table 1a. They were selected on the basis of initial experiments and our past experience with the methods.

The average prediction error obtained by both methods for the training data set and for the test data set is shown in table 1b. The training error shows how the model managed to learn the training data and the prediction error shows how well could the model evolved using the training data predict the power produced by the PVPP in previously unknown period covered by the test data set. The prediction error is around 1% of the installed power for the training data set and around 2% for the test data set, which is a satisfactory result, considering that only a single input variable (current solar radiation intensity), its past values and past output forecasts were available. We can also observe that both methods have achieved similar average prediction error for both, training and test data set. Both methods managed to estimate the PVPP output very well (despite imperfections in the training and test data) for some days and less accurately for some other days. An example of good predictions is shown in fig. 4 and an example of less accurate predictions is shown in fig. 5. Let us note that both neurocomputing methods have provided very similar forecasts which indicates that they have discovered in the data set similar trends and hidden dependencies.

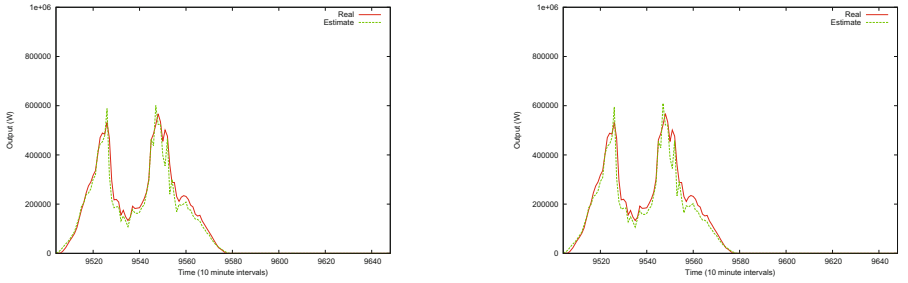
Table 1. Parameters

(a) MLP and ANFIS parameters

Method	Parameters
MLP	Training for 5000 iterations, three-layered 2-2-1 network, learning rate 0.3, momentum 0.2
ANFIS	Single input node, 10 fuzzy-input-membership nodes with Gaussian functions with a,b,c parameters for width, height and location of the function peak, 10 fuzzy rule nodes with multiplication t-norm, 10 normalization nodes, 10 fuzzy output function nodes with 2 premise paremeters each, 1 Σ node. Batch LSE learning without backpropagation.

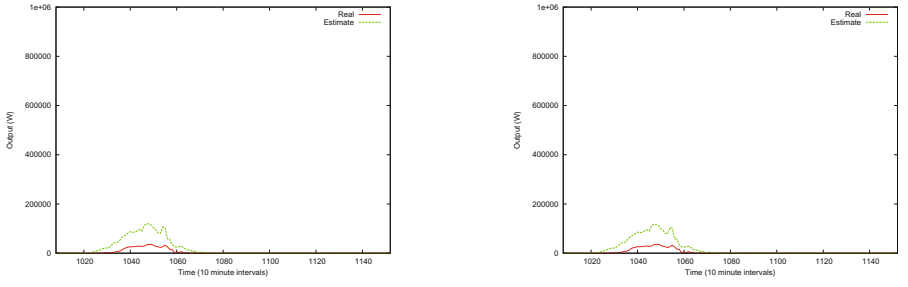
(b) Average PVPP output prediction error

Method	Training error (W)	Prediction error (W)
MLP	10000	22033.2
ANFIS	*9957.39*	*21323.8*



(a) Prediction by MLP (b) Prediction by ANFIS

Fig. 4. An example of day with good prediction



(a) Prediction by MLP (b) Prediction by ANFIS

Fig. 5. An example of day with bad prediction

5 Conclusions

MLP and ANFIS were used as predictors to estimate the output of a PVPP. An experiment with a real world photovoltaic power plant was conducted and a models based on the data describing more than three months of the operations of the PVPP were generated. The data contained only the information about the intensity of solar radiation in the location of the facility. Both methods have reached similar average prediction errors of about 2 % of the peak output of the power plant. However, the more sophisticated ANFIS was shown to provide slightly more accurate predictions than a simple feed-forward MLP. This is a promise for future development of custom predictors tailored to the needs of specific PVPPs as every PVPP is unique (because of e.g. the solar panel technology, configuration, age, location, geographical setting, and so on).

Acknowledgement. This paper has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic and supported by the Ministry of Industry and Trade of the

Czech Republic, under the grant no. FR-TI1/420 , and by SGS, VŠB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/58.

References

1. Almonacid, F., Rus, C., Prez, P., Hontoria, L.: Estimation of the energy of a pv generator using artificial neural network. Renewable Energy 34(12), 2743–2750 (2009)
2. Engelbrecht, A.: Computational Intelligence: An Introduction, 2nd edn. Wiley, New York (2007)
3. Fernandez-Jimenez, L.A., Muoz-Jimenez, A., Falces, A., Mendoza-Villena, M., Garcia-Garrido, E., Lara-Santillan, P.M., Zorzano-Alba, E., Zorzano-Santamaria, P.J.: Short-term power forecasting system for photovoltaic plants. Renewable Energy 44, 311–317 (2012)
4. İzgi, E., Öztopal, A., Yerli, B., Kaymak, M.K., Şahin, A.D.: Short mid-term solar power prediction by using artificial neural networks. Solar Energy 86(2), 725–733 (2012)
5. Jang, J.S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23(3), 665–685 (2002)
6. Konar, A.: Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain. CRC Press, Inc., Boca Raton (2000)
7. Krömer, P., Platoš, J., Snášel, V., Abraham, A., Prokop, L., Mišák, S.: Genetically evolved fuzzy predictor for photovoltaic power output estimation. In: INCoS 2011, pp. 41–46. IEEE (2011)
8. Mellit, A., Pavan, A.M.: A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected pv plant at trieste, italy. Solar Energy 84(5), 807–821 (2010)
9. Pedrycz, W.: Fuzzy Control and Fuzzy Systems, 2nd edn. John Wiley & Sons, Inc., New York (1993)
10. Reinoso, C.S., Cutrera, M., Battioni, M., Milone, D., Buitrago, R.: Photovoltaic generation model as a function of weather variables using artificial intelligence techniques. International Journal of Hydrogen Energy (2012)
11. Su, Y., Chan, L.C., Shu, L., Tsui, K.L.: Real-time prediction models for output power and efficiency of grid-connected solar photovoltaic systems. Applied Energy 93, 319–326 (2012)
12. Sugeno, M.: Industrial Applications of Fuzzy Control. Elsevier Science Inc., New York (1985)
13. Sulaiman, S.I., Rahman, T.K.A., Musirin, I., Shaari, S.: An artificial immune-based hybrid multi-layer feedforward neural network for predicting grid-connected photovoltaic system output. Energy Procedia 14, 260–264 (2012)
14. Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE Trans. on Systems, Man, and Cybernetics 15(1), 116–132 (1985)

An Intelligent System for Managing the Isolation of Patients Suspected of Pulmonary Tuberculosis

João Baptista de Oliveira e Souza Filho[1], Ana Paula Pereira Vieira[1], José Manoel de Seixas[2], Fábio Silva Aguiar[3], Fernanda Carvalho de Queiroz Mello[3], and Afrânio Lineu Kritski[3]

[1] Federal Center of Technological Education Celso Suckow da Fonseca (CEFET/RJ)
jsouza@cefet-rj.br, anappv2000@yahoo.com
[2] Signal Processing Lab – POLI/PEE/COPPE – Federal University of Rio de Janeiro (UFRJ)
seixas@lps.ufrj.br
[3] Tuberculosis Academic Program - Medical School – Federal University of Rio de Janeiro (UFRJ)
{aguiarmd,kritskia}@gmail.com, fcqmellho@hucff.ufrj.br

Abstract. In hospital internment context, patients suspect to have pulmonary tuberculosis, especially those which have a higher transmission risk, should be allocated in isolation respiratory rooms in order to reduce infection risk. Due to high implementation costs, these rooms are only available in a restricted quantity and have to be shared with patients having other infectious diseases. Currently applied criteria have been resulting in a large number of unnecessary patient isolations. This work proposes a decision support system based on neural networks to guide patient respiratory isolation. The system identifies the probability of a patient to have pulmonary tuberculosis and was developed using medical records data from 290 Pulmonary TB suspect patients who were admitted to isolation rooms in an AIDS/TB reference hospital (IDT-HUCFF-UFRJ). Based on 26 clinical-radiological variables, the system achieved a sensitivity of 94% and specificity of 89%. This system should be validated in other settings in order to confirm this high performance and its usefulness by avoiding unnecessary patient isolation as providing an earlier diagnosis, which may reduce the contamination risk.

Keywords: Decision support systems, Artificial Neural Networks, Tuberculosis Diagnosis.

1 Introduction

Tuberculosis (TB) is one of the main infectious diseases in developing countries and is associated with high morbidity and mortality. The perpetuation of TB worldwide is mainly caused by disease transmission, which is closely linked to social inequality, the advent of HIV infection and population aging [1].

One of the key factors related to the emergence of new TB cases is the time lag during which a patient having pulmonary tuberculosis sustains the transmission chain

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 818–825, 2012.

[2]. Thus, a rapid diagnosis is mandatory to epidemic control, since it allows the start of anti-TB treatment and diminishes the time during which the disease may be transmitted, reducing the number of possible new infected individuals [3].

The laboratory tests most commonly used to confirm TB diagnosis are smear microscopy and mycobacterial culture. Smear microscopy has low sensitivity (40% to 60%), especially in patients with low bacterial load in respiratory samples, in HIV-infected ones or for those which have other immunosuppressive diseases (30%) [4]. Mycobacterial culture has higher sensitivity (80% to 90%), however it takes long to be concluded (4 to 6 weeks), besides to be only available on reference or research health units in high TB burden countries. So, the development and the evaluation of new diagnostic methodologies are urgent. In a recent article, Marais et al propose to evaluate clinical and clinical-radiological scores to pulmonary TB diagnosis on different epidemiological scenarios [5].

In United States, the Centers for Disease Control and Prevention (CDC) recommends that patients with suspicion of pulmonary TB, when admitted to a hospital, should be allocated to respiratory isolation rooms [6]. However, this practice is only feasible in places with low TB incidence, since these rooms exist is a restricted number, their implementation is expensive and hospital management should also consider their use by patients having others pathologies.

There is not a standard criterion used for an early identification of in-patients having a high risk of pulmonary tuberculosis. Therefore, there is a diagnosis delay of smear positive TB patients and unnecessary isolation of those with low chance of having the disease. Besides, atypical clinical features found in patients with specific diseases may postpone the diagnosis and increases the risk of intra-hospital transmission [7].

In the hospital complex Thoracic Diseases Institute (IDT) – Clementino Fraga Filho Hospital (HUCFF), which belongs to the Federal University of Rio de Janeiro (UFRJ), according to strict criteria established by the Tuberculosis Control Program (PCTH), just 26.6% of the patients admitted to isolation rooms have a positive TB diagnosis. Some predictive models have been described to optimize use of respiratory isolation rooms in HUCFF [8], showing that patient isolation can be reduced by one third without a increase in risk of nosocomial transmission.

This work proposes a clinical-radiological score to aid the decision making task of respiratory isolation. The proposed system identifies the probability of a patient to have pulmonary TB based strictly on clinical-radiological information. The system employs a Multilayer Perceptron Artificial Neural Network (MLP) [9] due to its well-known ability on solving complex problems belonging to diverse domains and was developed using medical records of patients attended on hospital complex IDT-HUCFF-UFRJ.

The major contributions of the proposed system to hospital operation are: (i) provides a quick diagnosis tool, (ii) reduces the chance of TB transmission, (iii) permits an earlier detection of TB and, therefore, treatment in a less advanced stage, (iii) reduces the number of unnecessary patient isolations, (iv) permits a better management of respiratory isolation rooms, (v) improves the quality of patient attendance.

This article is structured as follows: initially, relevant aspects to the development of efficient neural networks considering the problem under study are addressed. Later, dataset used is described. Finally, results and conclusion are presented.

2 Neural Support to TB Diagnosis

Neural networks [9] have been adopted in the solution of a growing number of problems belonging to diverse knowledge areas. Their ability to extract relevant information and explore relationships (linear and non-linear) among descriptive variables to problem solution usually results on efficient models even in complex and high dimensional data spaces.

As neural networks do not presume any statistical assumption concerning problem variables, they are quite adequate to the development of models to support diagnosis, prognosis and therapy. Commonly, in these contexts, the number of patients having accessible records is restricted so as the quality of the available information is not adequate. Thus, these datasets may have severe statistical restrictions concerning a proper problem description.

Mathematical models applied to diagnosis support have to show high accuracy due to influence decisions which imply on life risk. Due to the exiting statistical restrictions, special care must be taken in the development of these systems, especially with respect to the selection of problem descriptive variables as the performance assessment criteria. Thus, the adoption of a methodology which properly evaluates the generalization skills of the system, so as provide a realistic estimation of its performance in real settings is necessary [10]. These aspects will be briefly discussed on the following.

2.1 Variable Selection

In order to develop an efficient classification system, the selection of high quality variables with good discriminative power is mandatory. Roughly speaking, this selection can be realized by automatic methods, guided by experts or explore both approaches [11]. Variable selection is usually both problem and dataset dependent.

Hybrid approaches, merging expert knowledge with the use of machine learning techniques, are quite useful to medical applications. In this direction, statistical analysis methods, heuristic procedures and computation intelligence techniques may be used to refine a set of variables selected by specialists, resulting in more effective and clinically plausible models, which easies their practical adoption.

Considering the problem in question, 29 variables were selected by TB experts. This set comprises signals, symptoms, the presence of specific diseases and information referent to clinical-radiological evaluation of the patient as follows: cough, fever, sweating anorexia, hemoptysis, hemoptoic, headache, face pain, sore throat, sputum, loss of weight, dyspnea, malnutrition, respiratory symptoms, chest pain, previous hospital admission, sex, previous pulmonary TB, alcoholism, chronic obstructive pulmonary disease, chronic kidney disease, diabetes mellitus, smoking, HIV, as the

presence of any following images on x-ray chest: cavitation, upper lobe or diffuse lesions.

Considering this set of variables, a basic statistical analysis (heuristic selection approach) was conducted upon the dataset to refine this selection. This procedure aimed to exclude variables both showing low quality and which did not contribute to system efficiency. Thus, groups of candidate variables to be excluded were established and evaluated on a defined sequence. Each group was formed by variables which exclusively attended the following characteristics: (i) low probability to be positive (low prevalence), (ii) high percentage of ignored cases and (iii) low difference on variable positivity between positive and negative TB patients (low odds ratio). Variables belonging to each group were also ranked in ascending (groups i and iii) or descending (group ii) order. The evaluation of variable relevance was based on MLP network performance assessment. One variable was excluded if this action did not impact on classifier performance.

2.2 Classifier Development

Special caution must be taken on neural network design choices, especially when the network is developed upon datasets showing restrictions relative to a proper problem description as have strict performance requirements.

A commonly adopted procedure explored to neural network development is hold-out [12]. This technique consists on splitting the dataset into two sets: training and test, the first used to obtain the model and the last to estimate their performance (a measure of generalization ability). For datasets with statistical restrictions, this split should be properly done, otherwise the model obtained may do not have an adequate quality (poor learning) or the estimate of its performance may be not realistic (too much optimistic or pessimistic).

Aiming to define training and test sets conjugating learning and generalization skills, a random splitting procedure may be adopted. This procedure is extensively applied on statistical model production, is easy to be implemented, as usually results on models exhibiting good performance. Following this approach, several pairs of sets are formed. For each pair, patients are randomly distributed between training and test sets. Models are built and evaluated based on these sets. The pair of sets which exhibits the highest generalization ability is selected.

Another factor which impacts on system performance is the size of the neural network, which is defined by the number of layers and neurons by each layer. Due to universal approximation theorem [9], three layer MLP networks are the most common adopted topology. In this case, only the number of hidden neuron has to be established, since the number of input nodes and output neurons are defined with basis on problem variables and target vectors, respectively. The number of hidden neurons is directly related to network power on mapping input-output relationships, i.e. their ability to learn. However, an increase on this number may affect learning, due to the growth in number of network parameters, especially for datasets having restrictions on problem description. Thus, this design parameter should be defined considering the

parsimonious principle [13], i.e. the minimal number of neurons which maximizes performance should be employed.

Neural networks are prone to overtraining phenomena [9], demanding a proper learning mechanism control. Early stop is usually used to control network learning. This procedure consists in interrupt training on the epoch for which network starts to loose generalization.

Finally, the network performance is susceptible to the value of the parameters (bias and weights) used at the start of training procedure. This is caused by the multiple local minima problem [9]. Usually, in order to mitigate this effect, several models are produced, each one based on a random set of start parameters. The best initialization set is defined by the highest performance network.

2.3 Evaluating Neural Networks

Cross-validation procedures developed during neural network development demand split dataset into three sets: training, validation and test [9]. Both validation and test sets are used to evaluate model generalization, but on different development stages. While validation set is used to choose design parameters as the number of hidden neurons and to early stop procedure, the test set is employed to assess the performance of the final model. For applications having datasets with small number of samples, this may be not convenient or even feasible as may result in too small sets. So, in these cases, data is just split in training and test sets, the last one also used for validation purposes.

Sensitivity and specificity, which identifies the percentage of TB positive and negative patients correctly diagnosed by the system, are relevant performance attributes. The maximization of both indexes is desirable, but they represent conflicting interests. Thus, in order to evaluate model performance, the following index was adopted:

$$I_{SP} = \sqrt{(S + E)/2\sqrt{S.E}} \tag{1}$$

This index was inspired on the sum-product index [14] and benefits solutions with balanced values of sensitivity (S) and specificity (E).

2.4 Dataset Description

Dataset is constituted by hospital patients suspicious of have active pulmonary tuberculosis which were allocated to respiratory isolation rooms. They were attended in hospital complex IDT/HUCFF/UFRJ, which belongs to the Brazilian National Health Service (SUS), during the period of March 2003 to December 2004 [7]. Patients were followed-up during clinical evolution for at least 60 days. For each one, a standard form containing demographic data, clinical symptoms and risk factors to pulmonary TB, as other information was filled. The sample has 290 patients whose estimated TB prevalence is 26.6% (77/290).

3 Results

The proposed decision support system explored two-layer MLP totally connected networks having hyperbolic tangent as activation functions and one neuron on output layer. All variables, except age, were coded as -1 (absence), 0 (unknown) and +1 (presence). Age was normalized to be inside the -1 and +1 interval, so as to have zero mean. Target values were +1 (positive pulmonary TB) and -1 (negative). The training algorithm was Resilient propagation – RPROP [15], using batch gradient estimation. Due to the existing imbalance between the number of positive and negative TB patients (class-imbalance), the following objective function was used during training:

$$E = E_{TB+} + E_{TB-} \tag{2}$$

where E_{TB+} and E_{TB-} correspond to the mean square error evaluated for positive and negative TB patients.

Performance assessment employed stratified hold-out approach [16], considering for each group of positive and negative TB patients, 80% of them to training and 20% to test set. In order to define the training and test set contents, 10 pairs of sets were produced according the random selection approach. MLP networks from 1 to 20 hidden neurons were built for each pair of these sets. For each network, 50 models were also produced, each on based on a different set of start parameters. The best overall network was selected according I_{SP} value evaluated upon test sets. This index was also used in early stop procedure.

Initially, a MLP model based on variables selected by TB experts was build (model M1). In the sequence, the elimination criteria described in Section 2.1 was conducted. The variables face pain, CKD and throat pain, in this order, was identified as satisfying group (i) criterion, for which a prevalence limit of 7% was considered. Additional models (M2 to M4) were developed to evaluate the exclusion of each one of these variables, so as a new model (M5) for which the variables hemoptoic and hemoptysis symptoms were fused according expert suggestion (a positive value was taken if any of them was positive). The number of hidden neurons, the sensitivity and specificity values obtained for these models are summarized in Table 1.

Table 1. Number of hidden neurons (#), sensitivity and specificity values for the different models evaluated

Models	# hidden neurons	Sensitivity	Specificity
M1	11	94%	79%
M2	11	94%	84%
M3	18	94%	81%
M4	8	94%	86%
M5	**10**	**94%**	**89%**

Results show that the exclusion of face pain (model M2) resulted on a model with higher specificity. However, CKD variable should not to be excluded (model M3) due to imply on specificity reduction. Sore throat was also excluded (model M4), which

resulted on an increase on specificity. The fusion of hemoptoic and hemoptysis symptoms was benefic too, since resulted on a new increase on specificity.

In the following, variables meeting group (ii) and (iii) criteria were identified. For both cases, however, the exclusion of any of these variables reduced model performance.

4 Conclusion

In this work, a decision support system based on an artificial neural network for managing the respiratory isolation of patients suspicious of pulmonary TB was developed. One major benefit of its adoption is to provide an early and easy detection tool, which is extremely valuable for hospitals, prisons and hostels in high burden countries where there are places lacking of diagnostic resources. This system may reduce the number of unnecessary isolations, which provides a more effective usage of resources and allows significant operational cost reduction.

Based on a set of variables selected by experts that has been refined through an auxiliary statistical analysis, the system reached 94% of sensitivity and 89% of specificity using 26 descriptive variables.

As a future work, we intend to evaluate specific model production for pulmonary TB detection on positive and negative HIV patients. In addition the system will be evaluated in terms of cost effectiveness in different sites in Brazil.

Acknowledgements. We are thankful to hospital complex IDT-HUCFF-UFRJ for providing the database used in this work and for their technical cooperation. We also thank CNPq and FAPERJ for financial founding to this work.

References

1. World Health Organization: Global Tuberculosis Control: surveillance, planning, financing. World Health Organization, Geneva (2010)
2. Dye, C., Williams, B.G., Espinal, M.A., Raviglione, M.C.: Erasing the world's slow stain: strategies to beat multidrug-resistant tuberculosis. Science 295, 2042–2046 (2002)
3. Fair, E., Hopewell, P.C., Pai, M.: International Standards for Tuberculosis Care: revisiting the cornerstones of tuberculosis care and control. Expert Rev. Anti. Infect. Ther. 5, 61–65 (2007)
4. Sester, M., Giehl, C., McNerney, R., Kampmann, B., Walzl, G., Cuchí, P., Wingfield, C., Migliori, G.B., Kritski, A.L., Meyerhans, A.: Challenges and perspectives for improved management of HIV/TB coinfection. Eur. Respir. Journal 36, 1242–1247 (2010)
5. Marais, B.J., Raviglione, M., Donald, P.R., Harries, A.D., Kritski, A.L., Graham, S.M., El-Sadr, W., Harrington, M., Mwaba, P., Churchyard, G., Kauffman, S.H.E., Whitty, C.J.M., Atun, R., Zumla, A.: Scale-up of services and research priorities for TB diagnosis, management and control: Call to action. Lancet. 375, 2179–2191 (2010)
6. CDC: Guidelines for preventing the transmission of Mycobacterium tuberculosis in healthcare facilities. MMWR 43, 1–132 (1994)

7. Selig, L., Kritski, A., Lapa e Silva, J.R., Guedes, R., Braga, J.U., Trajman, A.: Tuberculosis death surveillance in Rio de Janeiro Hospitals. Int. J. Tuberc. Lung. Dis. 13, 982–988 (2009)
8. Almeida, L.L.: Evaluation of predictive models to pulmonar TB diagnosis of patients attended on a tertiary hospital – classification trees. Master thesis. Medical Faculty. Federal University of Rio de Janeiro (2005) (in Portuguese)
9. Haykin, S.: Neural networks and learning machines. Prentice Hall, USA (2008)
10. Sahiner, B., Chan, H.P., Hadjiiski, L.: Classifier performance estimation under the constraint of a finite sample size: resampling schemes applied to neural network classifiers. Neural Networks 21, 476–483 (2008)
11. Liu, H., Motoda, H.: Computational methods of feature selection. Chapman and Hall, USA (2007)
12. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Annals of International Joint Conference on Artificial Intelligence, vol. 2, pp. 1137–1143 (2007)
13. Medeiros, M.C., Terävirta, T., Rech, G.: Building neural network models for time series: a statistical approach. Journal of Forecasting 25, 49–75 (2006)
14. Seixas, J.M., Damazio, D.O.: A neural particle discriminator for calorimetry in high energy physics. In: Annals of "The Third Brazilian Congress on Neural Networks", Florianópolis, Brazil, vol. 1 (1997)
15. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: Proceedings of the IEEE International Conference on Neural Networks, ICNN, vol. 1, pp. 586–591 (1993)

Application of a Perceptron Artificial Neural Network for Building the Stability of a Mining Process

Anna Burduk[1] and Paweł Stefaniak[2]

[1] Wrocław University of Technology, 27 Wybrzeże Wyspiańskiego St, 50-370 Wrocław, Poland
[2] KGHM Cuprum Sp. z o.o. Research and Development Centre

Abstract. The paper describes a method of ensuring the stability of the selected mining process (loading and haulage of copper ore) taking place under variable environmental conditions. Four models of a multilayer perceptron neural network were built for this purpose. Travel times and the condition of transport roads were adopted as input parameters. The output of the network is the cycle time of the analysed process. On the basis of an analysis of learning errors, a model with two hidden layers was selected. A series of experiments was conducted on the selected model. An assessment was also performed to determine at which values of input parameters the stability of the analysed process could be ensured.

Keywords: production system, production process stability, neural networks, perceptron.

1 Introduction

A condition for efficient management of an enterprise is to ensure its predictability and stability. This problem is particularly important in the case of mining processes that occur in a variable and unpredictable rock mass environment. Considering the number of elements involved in the process and affecting it, an adequate control of the process can be ensured only through the application of intelligent methods to the process model.

A problem of contemporary enterprises is not a lack of data on production processes, but their acquisition and processing into the information that can be then used for better management of a production process. Production systems are measured and monitored to a significant degree. Many organizational, process-related, cost-related and other data are stored every day in databases of IT systems. After some time, these databases contain millions of data collected from processes, which provide historical information concerning the course of these processes. However in most cases the problem consists in their proper processing and understanding. Intelligent methods, such as artificial neural networks (ANN), are used for this purpose more and more often.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 826–833, 2012.

2 Production Process Stability

There are many methods for building the stability of production processes. However all of them are associated with building a model, determining the parameters and their ranges, as well as with monitoring and control of parameters, if the parameters exceed the set range [3, 6]. Process control takes place by appropriate selection of input values so that a desired output value is obtained.

The purpose of parameterisation of a process is its control. Selection of appropriate parameters, which are to be monitored, provides a basis for building and improving the stability of processes in the mining industry. The problem of monitoring the parameters, which determine the proper execution of a production process, is crucial for the operational management. Monitoring can be regarded as the most important element in the process of control. The monitoring frequency varies and depends on the variability of a given parameter and its significance for the process [9].

From the viewpoint of the whole loading and haulage process, building the stability will consist in determination of values of the parameters for specific states at the moments $t_0, t_1, \ldots, t_n, \ldots, t_{n+1}$. For example, the operation of loading the output on a haulage vehicle is described by the values at the moment t_n, while the features of the process at the moment t_{n+1} will concern the operation of unloading the output on the grate. If we want a process to be stable, we must ensure the stability of its individual stages, and therefore the measures in specific moments of the process should approach to the specific x values from the standard.

$$P_n(t_n) \rightarrow x \tag{1}$$

By comparing the values of the parameters with the established standard it can be concluded whether the transport of the output to the grate is stable. Fig 1 shows a sample diagram presenting how the stability of a process is ensured by monitoring the parameters and comparing them with the established standard.

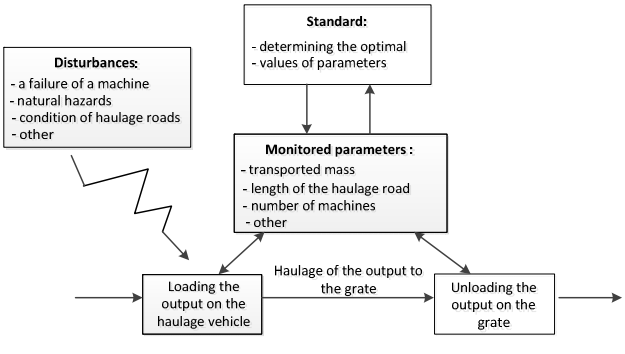


Fig. 1. Diagram presenting how the stability of a loading and haulage process is ensured

Optimal functioning of processes, i.e. such functioning that enables individual indicators of the parameters to reach the most desired value while fulfilling all the restricting conditions, is associated with defining the standard of values. The standard covers the range of values that will allow maintaining the stability of processes. Continuous monitoring of the parameters will allow determining an optimum standard.

For example, the parameter of the transported mass should be constant - both too small and too large mass of the transported ore can disrupt the operation of the entire system. Therefore, it is important to select such parameters and such ranges of their values that will ensure the invariability of the process and, over a longer period of time, also its optimisation.

3 Possible Applications of Neural Networks in Production Process Modelling

Neural networks (NN) are very popular in almost all fields of science and technology. They offer a new methodology for modelling complex processes and are able to reproduce the complicated relationships between the elements without the necessity to build a complex mathematical model. They can be used for solving problems associated with prediction, classification and control.

The primary objective of modelling the dynamics of a production process is to identify the temporal variability of its physical quantities or states [8]. To this end, a time series, i.e. an ordered sequence of values of a certain variable over time, should be determined. A time series can have a form of a vector $[y(t_1), y(t_2), \ldots, y(t_N)]$. Due to the fact that process parameters may differ in separate phases of the process, the time series vector can take the form of a vector defined in N-dimensional space. Individual components of this vector will be the states of the production process stages in the past, which in turn can be regarded as points in a multi-dimensional output space. Thus the task of analysing the temporal variability of the production process can be reduced to searching N-dimensional space for a certain trajectory, on which the analysed output variable of the process "moves". Therefore, a given quantity in the form of a time series is determined in order to predict its value in future moments.

A unidirectional neural network can describe the regularities occurring in a time series and allows predicting its future values. Usually, a future value of time series $y(t+1)$ is predicted on the basis of the current and k past values y as well as the current and 'l' of past values of input variables x according to the formula [5, 8]:

$$y(t+1) = f(y(t), y(t-1), \ldots, y(t-k), x(t), x(t-1), \ldots, x(t-l), w) \tag{2}$$

where f is the transformation effected by the network, while w is a set of weights.

The method of predicting future values of a time series is schematically presented in Fig. 2.

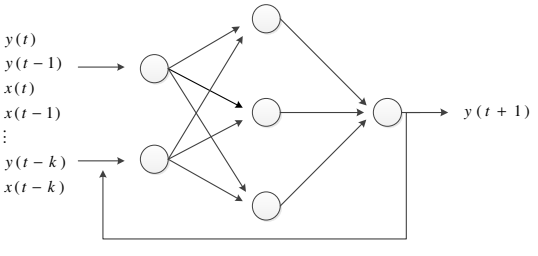


Fig. 2. Method of transforming a time series by a unidirectional network [8]

After predicting the value y at the moment (t+1), this value is entered to the network input and the value of the variable at the moment (t+2) is calculated. In this way the network can execute the dynamic model of the entire production process stage by stage.

The perceptron, due to its activation function, takes only two different output values, so it can classify the signals entered at the input in the form of vectors $X = [x_1, x_2, \ldots, x_n]^T$ into one of two classes. A perceptron with one input can assess whether the input signal is positive or negative. In the case of two inputs x_1 and x_2 the perceptron divides the plane into two parts. This division is determined by a straight line defined by the equation [2, 7]:

$$x_1 w_1 + x_2 w_2 + b = 0 \tag{3}$$

where b is the value of the bias.

For n inputs, the perceptron divides n–dimensional space of input vectors X into two half-spaces. They are separated by $n - 1$-dimensional hyperplane, called a decision boundary.

The perceptron learning algorithm is based on error correction, i.e. the change in the difference of weights is proportional to the magnitude of the error made by perceptron. By comparing the current output value y_i with the set value, d_i the update of weights presented below is performed. The course of the perceptron learning procedure looks as follows [1]:

- if $y_i = d_i$, then the weights w_{ij} remain unchanged,
- if $y_i = 0$ and $d_i = 1$, then the weights are updated according to the formula:

$$w_{ij}(t+1) = w_{ij}(t) + x_j \tag{4}$$

where t denotes the previous period, while $t + 1$ - the current period,

- if $y_i = 1$, and $d_i = 0$, then weight update is performed according to the formula:

$$w_{ij}(t+1) = w_{ij}(t) - x_j \tag{5}$$

The weights change procedure is of an iterative character and aims at zeroing the error.

4 Unidirectional Multi-layer Perceptron Network for Determining the Stability of the Loading and Haulage Process

For further analysis of the problem of building the stability of a mining system, the process of loading and haulage was selected. This is one of the stages of the most important process, i.e. mining. The amount of copper ore transported to processing plants depends on the stability of this process. Too small amount of copper ore can cause downtime in the processing plant, whereas too large amount may lead to overfilling the holding tanks or to a failure of belt conveyors.

The purpose of the loading and haulage process is to move the ore from the mine face to the grate, where the ore is crushed and discharged onto a belt conveyor. Main stages of this process are shown in Fig. 3.

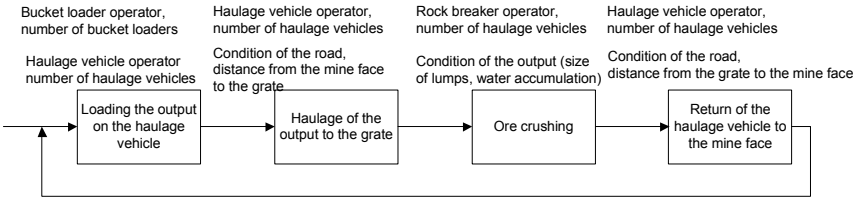


Fig. 3. Main stages of the loading and haulage process

1. Loading a haulage vehicle. This stage consists in loading the copper ore on a haulage vehicle with the use of a bucket loader. It takes place at the mine face, where the copper ore is mined.
2. Haulage of the ore to the grate. The loaded haulage vehicle goes to the discharge point, i.e. to the grate. The transport of the ore takes place on roads, the condition of which may vary depending on the type of rocks making up a given road. If the rocks are fairly soft, the road becomes muddy over time and ruts are formed in it. As time goes on, the transport on such a road is becoming more and more difficult and therefore longer.
3. Discharge of the ore. The haulage vehicle discharges the ore on the grate. The ore is then crushed by a rock breaker and goes through holes in the grate onto a belt conveyor. The time of operation of the rock breaker depends on the size of the rocks brought in and the degree of water accumulation in the ore.
4. The haulage vehicle, after unloading on the grate, goes back to the mine face to be loaded again.

The main purpose of the loading and haulage process is to provide an appropriate amount of the copper ore (output) to the discharge points. Disturbing factors include variable environmental conditions, which cause that the time of transport of the output and the time of return of the haulage vehicle from the grate to the mine face are very variable.

In order to consider the mining process to be stable, it should deliver the established amount of copper ore to the processing plants. Since the haulage vehicle has a constant and limited capacity, the amount of the ore getting to the processing plants depends on the number of haulage vehicles unloaded on the grate. Although the number of the haulage vehicles involved in the process can be changed, the travel time depends on the length and condition of the haulage road.

Continuous monitoring and measurements are necessary to control a process, maintain its constancy over time and in the long term enable its optimization. The main factor that should be controlled in the process of the ore transportation is the real production value, because it is an important parameter for the whole copper mine and is easily measurable.

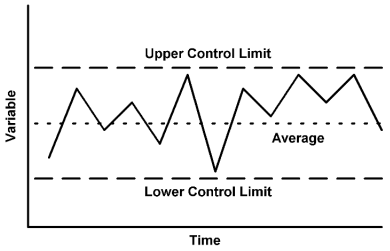


Fig. 4. Parameter values of process in stability state

Control of the stability state by inspection of the real production value seems to be one of the most important manner of monitoring this process stability. Comparison the resulting value of production with the developed standard for the analysed division will allow to assess whether the process is stable. Manoeuvring of the number of haulage vehicles appears to be the most obvious action.

In order to predict the cycle time of the loading and haulage process at the set input values, a unidirectional perceptron neural network was built. The results of the observations and measurements of the times in the loading and haulage process were used as a set of learning data. In total, 60 measurements were performed in three working shifts. As the inputs of the neural network, there were selected parameters such as:

x_1 - time of the travel of a haulage vehicle to the grate,
x_2 - time of the return of the haulage vehicle to the mine face, and
x_3 - condition of the haulage road.

Considering the safety regulations that prohibit walking on haulage roads, the data on the road condition were collected by obtaining information from operators of the haulage vehicles. They characterized the condition of roads using the descriptions presented in Table 1.

Table 1. Possible parameters of haulage roads and their descriptions

good	3	- road covered with mud up to 20 cm, without ruts and with a small inclination angle,
average	2	- road covered with mud up to 40 cm, with small ruts and with a medium inclination angle,
poor	1	- road covered with mud with a height over 40 cm, with a large inclination angle.

Other input parameters, such as the time of operation of the rock breaker, the loading time, or failures of haulage vehicles, did not affect the output values of the network.

The time of one process cycle (from loading a haulage vehicle to its return to the mine face) was adopted as the network output. It has been assumed that the process will be stable if the cycle time is less than 15 minutes and greater than 10 minutes.

If the time of a process cycle is greater than 15 minutes, then an adequate amount of output will not be delivered to the processing plant during one production shift.

Too short time of a process cycle will result in delivering too much copper ore, which in turn may lead to overfilling the holding tanks and blocking the belt conveyors.

Four models of a multilayer perceptron neural network were built in total:

Model 1 - with one hidden layer,
Model 2 - with two hidden layers,
Model 3 - with three hidden layers,
Model 4 - generalized linear model.

Values of learning errors for the tested networks are presented in Table 2.

Table 2. Values of learning errors in the network models built

Model	**Mean squared error**	**Final prediction error**	**Sum of squared errors**
Model 1	4.208	2.918	172.509
Model 2	3.729	2.543	152.909
Model 3	3.733	2.918	153.068
Model 4	4.388	2.610	179.921

As it appears from Table 2, the lowest learning errors were obtained in Model 2, i.e. a multi-layer perceptron with two hidden layers. Model 2 was therefore used for predictions of the cycle time in the loading and haulage process depending on the travel time and road conditions. Sample results of experiments on this model for the shortest, average and longest times of transport are shown in Table 3.

Table 3. Results of experiments on Model 2 for the selected input data

Test No.	We_1 Time of travel to the grate [min.]	We_2 Time of return to the mine face [min.]	We_3 Condition of the haulage road	Wy Time of process cycle [min.]
1.	3.0	2.0	3	8.197759607
			2	8.897754296
			1	10.05702046
2.	4.5	3.0	3	11.47988548
			2	14.38787992
			1	18.56142567
3.	8.7	7.9	3	12.59800458
			2	16.77015184
			1	21.71128117

In the test No. 1, due to low values of the inputs x_1 and x_2, the process is stable only in the case of a haulage road of poor quality. This is caused by a small distance between the mine face and the grate, and thus a short time of transport and return even on a road of average quality.

In the test No. 2, which was conducted for mean values of the transport and return times, the process is stable only in the case if a road of good or average quality. Average times represented the vast majority in the set of measurement data, and hence

it can be assumed that the road condition has a significant impact on the process stability.

In the test No. 3, which was conducted for the maximum time values obtained from measurements, it is visible that the process can be stable only at a good condition of the haulage road.

The results obtained from the neural network model allow the persons responsible for managing the loading and haulage process to make decisions regarding the process inputs so that the process is stable over a longer period of time.

References

1. Bilski, J.: The UD RLS algorithm for training feedforward neural networks. International Journal of Applied Mathematics and Computer Science 15(1), 115–123 (2005)
2. Bilski, J., Rutkowski, L.: A fast training algorithm for neural networks. IEEE Trans. Circuits Syst. II 45(6), 749–753 (1998)
3. Burduk, A., Chlebus, E.: Methods of risk evaluation in manufacturing systems. Archives of Civil and Mechanical Engineering 9(3), 17–30 (2009)
4. Frumusanu, G., Epureanu, A., Ionut, C.: Cutting process stability evaluation by process parameters monitoring. In: Proceedings of the 8th WSEAS International Conference on Non-linear Analysis, Non-linear Systems and Chaos, NOLASC 2009, vol. 1, pp. 345–350 (2009)
5. Mahmood, K., Zidouri, A., Zerguine, A.: Performance analysis of a RLS-based MLP-DFE in time-invariant and time-varying channels. Digital Signal Processing 18, 307–320 (2008)
6. Sankar, N., Prabhu, B.: Modified approach for prioritization of failures in a system failure mode and effects analysis. International Journal of Quality & Reliability Management 18(3), 324–336 (2001)
7. Rutkowski, L.: Adaptive Probabilistic Neural Networks for Pattern Classification in Time-Varying Environment. IEEE Transactions on Neural Networks 15(4), 811–827 (2004)
8. Rutkowski, L.: Computational Intelligence. Methods and Techniques. Springer, Heidelberg (2008)
9. Tung-Hsu, H., Wang-Lin, L., Li, L.: Intelligent remote monitoring and diagnosis of manufacturing processes using an integrated approach of neural networks and rough sets. Journal of Intelligent Manufacturing 18(2), 239–253 (2003)

Merge Method for Shape-Based Clustering in Time Series Microarray Analysis

Irene Barbero[1], Camelia Chira[1,2], Javier Sedano[1,3], Carlos Prieto[4], José R. Villar[5], and Emilio Corchado[6]

[1] Instituto Tecnológico de Castilla y León, Burgos, Spain
{irene.barbero,camelia.chira,javier.sedano}@itcl.es
[2] Department of Computer Science, Babes-Bolyai University, Cluj-Napoca, Romania
[3] Department of Electromechanical Engineering, University of Burgos, Spain
[4] Instituto de Biotecnología de León, Spain
carlos.prieto@unileon.es
[5] University of Oviedo, Gijón, Spain
villarjose@uniovi.es
[6] University of Salamanca, Spain
escorchado@usal.es

Abstract. A challenging task in time-course microarray data analysis is to combine the information provided by multiple time series in order to cluster genes meaningfully. This paper proposes a novel merge method to accomplish this goal obtaining clusters with highly correlated genes. The main idea of the proposed method is to generate a clustering, starting from clusterings created from different time series individually, that takes into account the number of times each clustering assemble two genes into the same group. Computational experiments are performed for real-world time series microarray with the purpose of finding co-expressed genes related to the production and growth of a certain bacteria. The results obtained by the introduced merge method are compared with clusterings generated by time series individually and averaged as well as interpreted biologically.

Keywords: microarray analysis, time series, clustering, merge methods.

1 Introduction

Nowadays, microarray technology provides the possibility to measure gene expression levels of thousands of genes. Some difficulties to face with this type of data are the cost and the high dimensionality. Microarray data can be analysed from a static viewpoint [1, 2, 3, 4], ignoring the temporal information of time series, or from a temporal perspective [5, 6, 7, 8]. In this paper, we focus on temporal microarray data analysis.

A premise widely accepted when analysing microarray data is that genes showing similar behaviour in their expression levels, i.e. co-expressed, are possibly functionally related [9]. Hence, the task of clustering genes based on time series

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 834–841, 2012.

is extremely important. Clustering time series microarray data has been intensively investigated and many algorithms have been proposed to carry out this task without any clear overall best performing method. These methods can be classified in point-wise distance based, feature based clustering and model based clustering [6]. Point-wise distance methods compute the clusterings according to an objective function based on a distance measure between gene pairs, feature based clustering methods detect some features among the expression profiles and model based clustering methods create models capable of describing the data. In this paper, we engage two feature based clustering methods with the purpose of merging their results for different time series. These clustering methods are shape index clustering [10] and an extension of it introducing the correlation with an output called Shape Output Clustering.

A merge method is proposed to blend clusterings created independently from time series in order to get a new more meaningful clustering. In the first phase, several clusterings of the same genes are produced using the shape clustering methods based on individual time series data. In the second phase, one single clustering is obtained by merging the groups resulted from applying the same shape-based clustering method to different time series data. The results show how the proposed method is able to generate more restrictive and highly correlated gene clusters. Moreover, this method can help in future work to overcome the problem of detecting significant genes that promote a certain event or variable.

The paper is structured as follows: section two describes in detail the shape-based clustering methods; section three explains thoroughly the merge clustering method proposed; section four discusses the experimental results obtained for the real-world time series microarray data considered, giving also a biological perspective of the results and section five contains the conclusions of paper.

2 Shape-Based Clustering Methods

The grouping of genes based on values from individual time series is performed using two shape-based clustering methods: Shape-based Clustering (SC) and Shape Output Clustering (SOC). Both methods rely on modelling the change of expression value between consecutive time points using a shape index. The first method engaged is described in [10] where it is used as the first step of a clustering methodology for time series microarray data based on similar rate of change and modulation patterns in gene expression profiles. SOC extends the functionality of SC by further taking into account the gene correlation with the output. The main idea of SOC is to group two genes together if they follow a similar pattern of changes in gene expression over time and with regard to the output.

Let S be the number of time points considered (samples available). For each sample $i, i = 1 \ldots S$, the value of the current time point is given by t_i and the corresponding gene expression level is denoted by x_i. Let g_step be the rate of change in the gene profile calculated at each time interval as the difference

between the two consecutive gene expression levels x_{i+1} and x_i divided to the difference between the time points. Similarly, the rate of change in the output *y_step* is computed using the value of output instead of the gene expression level.

The values of *g_step* and *y_step* (*step* for short) are used to decide how significant is the change in gene expression level from one time point to the next and how correlated with the change in the output at the same time interval. This is achieved through the use of a threshold ψ which indicates the level of acceptable difference between two consecutive values. Based on the relationship between the step values and the threshold, *g_class* and *y_class* (*class* for short) levels are assigned to each time interval indicating the rate of change in the gene/output pattern. If $step \in (-\psi, \psi)$, the level of change is not significant and the gene/output category *class* has a 'no change' (*stable*) meaning associated. If $step \geq \psi$ (respectively $step \leq -\psi$) then we have an increase (respectively decrease) of the gene expression level.

Once all *g_class* and *y_class* values are assigned, the *g_index* corresponding to a gene is calculated following Eq. 1 for SC, respectively Eq. 2 for SOC (l is the number of different categories that can be assigned to a gene).

$$g\_index^{SC} = \sum_{i=1}^{S-1} l^i * g\_class(t_i, t_{i+1}) \tag{1}$$

$$g\_index^{SOC} = \sum_{i=1}^{S-1} l^i * g\_class(t_i, t_{i+1}) * y\_class(t_i, t_{i+1}) \tag{2}$$

Two genes are placed in the same group if they have the same *g_index* value. In SOC, the rate of change in the output level is allowed to directly influence the gene shape index and, consequently, the gene clustering process. The final result is a set groups of genes having the same *g_index* and therefore similar shape of change and output correlation.

3 Merging Shape-Based Clusterings

The proposed merge method combines the information provided by multiple different time series clustered separately by either SC or SOC method. The aim is to create a combination of all of them resulting in a more meaningful clustering. In the context of the considered microarray problem, this merge method has been applied to three time series but it can be generalised to any number of time series.

Let N be the number of genes and S the number of time series in the microarray data. The input of the merge method is represented by the clusterings created by SC/SOC. Let C be a matrix that encompasses this information: c_{ij} is the cluster assigned to a gene g_j when applying SC/SOC to the time series i for $i = 1...S$, $j = 1...N$. The merge method computes a value denoted by v_{ij} for each pair of genes (g_i, g_j) such that $i < j$. These values are saved in a triangular matrix V.

The main steps of the merge shape index clustering algorithm are as follows:

(i) Assign a value to each possible combination of two genes (all v_{ij} of $V, i < j$) depending on the number of time series that group together that pair of genes according to Eq.(3).

$$v_{ij} = \frac{1}{S} \sum_{s=1}^{S} d(c_{si}, c_{sj}) \qquad d(c_{si}, c_{sj}) = \begin{cases} 1 & \text{if } c_{si} = c_{sj} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

(ii) Obtain the unique values of v_{ij} for all $i < j$ in descending order, which are denoted by $\alpha = \{\alpha_1, \alpha_2, \alpha_3 \ldots, \alpha_k\}$, so that α_1 is the greater value and α_k the smallest.

(iii) Generate the merged clustering M which consists in a list of clusters, initially empty, as outlined in the pseudocode below. The process consists in creating first clusters of genes that were detected as co-expressed by the clustering in all time series. In the second step, clusters are formed with the genes that were detected as co-expressed by the clustering in all time series except one and which are not formed in the previous step. This process continues iteratively until the last clusters formed correspond to genes that are considered to be co-expressed only by the clustering in one time series.

Merge Clustering Procedure:

$M = \emptyset$
$genesConsidered = \{g_1, g_2, g_3, ..., g_N\}$
for each α_l, $l = 1...k$
  $genesConsidered = genesConsidered - M$
  for each v_{ij} such that $g_i, g_j \in genesConsidered$
    if $v_{ij} >= \alpha_l$
      if $g_i \notin M$ and $g_j \notin M$
        create new group $M_{new} = \{g_i, g_j\}$;
        add M_{new} to M
      elseif $g_i \notin M$ but $\exists u \in [1, size(M)] \mid g_j \in M_u$
        $M_u = M_u \cup \{g_i\}$
      elseif $g_j \notin M$ but $\exists u \in [1, size(M)] \mid g_i \in M_u$
        $M_u = M_u \cup \{g_j\}$
      elseif $\exists p \in [1, size(M)] \mid g_j \in M_p$ and
      $\exists q \in [1, size(M)] \mid g_i \in M_q$
        create new group $M_{new} = M_p \cup M_q$
        remove M_p and M_q from M
        add M_{new} to M
      endif
    endif
  endfor
endfor
$M_0 = \{g_1, g_2, g_3, ..., g_N\} - M$

The result is a set of clusters of co-expressed genes denoted by $M = \{M_1, M_2, M_3, \ldots, M_m\}$. Every cluster has an associated α value that created it. This α value will provide information about the number of times that the genes of the group were clustered together by all time series, so that the greater α, the more correlated the genes are. Furthermore, the merge method also generates a cluster M_0 which contains the unclustered genes considered not to be correlated with any other.

4 Computational Experiments and Results

The microarray design was performed with the software eArray 5.0 and has provided the capacity to measure the expression of 8848 genes. Biological samples were extracted as time series of 12 time points that belonged to three different cell lines. The microarray dataset was preprocessed prior to applying a clustering method and combining the clusterings using the proposed merge method.

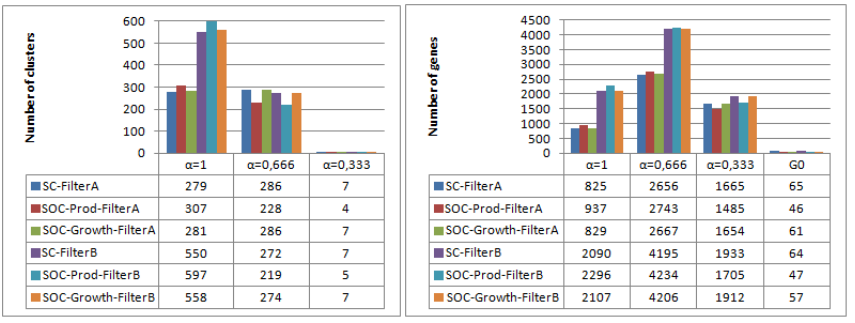
A normalization process was performed with the limma package [11]. Median and none background correction and Cyclic Loess normalization methods were applied in order to determine the best input for the implemented methods. Furthermore, a filtering phase with the aim of eliminating noisy genes was engaged. Two different filters were used as follows: (i) *FilterA* removes those genes for which the difference between the maximum and minimum value taken through the samples is greater than 1.5, and (ii) *FilterB* removes those genes for which difference between the maximum and minimum value taken is less than 0.75 and its standard deviation is less than 0.25. Eventually, FilterA selected 5211 genes and FilterB 8282 genes (out of the total 8848 genes).

The analysis of results compares the proposed method with the outcomes of SC and SOC methods individually applied over the time series as well as applied to the mean of the three time series.

4.1 Merge Clustering Results

To perform the experiments, the three time series (denoted by series1, series2 and series3) have been used independently to generate three different clusterings and averaged (called Mean) to produce one single clustering. Over these data, the SC and SOC methods are applied with threshold $\psi = 0.08$. SOC can use one of the two outputs available in the considered microarray data: production (SOC-Prod) or growth (SOC-Growth). The proposed merge method has been applied over the clusterings created from the time series independently.

Analysing the results of the merge method, the largest number of clusters have an associated α value of 1 and only few clusters emerged from an $\alpha = 0.333$ value corresponding to genes correlated only in one of the three time series (see Fig. 1 left). Nevertheless, if we consider the amount of genes classified within all clusters created according to each α value, by far, the greatest number of genes corresponds to $\alpha = 0.666$ (see Fig. 1 right). Generally, the size of clusters corresponding to $\alpha = 0.333$ and $\alpha = 0.666$ is high while their number is really low. However, the remaining groups contain rather few genes compared to the



	α=1	α=0,666	α=0,333
SC-FilterA	279	286	7
SOC-Prod-FilterA	307	228	4
SOC-Growth-FilterA	281	286	7
SC-FilterB	550	272	7
SOC-Prod-FilterB	597	219	5
SOC-Growth-FilterB	558	274	7

	α=1	α=0,666	α=0,333	G0
SC-FilterA	825	2656	1665	65
SOC-Prod-FilterA	937	2743	1485	46
SOC-Growth-FilterA	829	2667	1654	61
SC-FilterB	2090	4195	1933	64
SOC-Prod-FilterB	2296	4234	1705	47
SOC-Growth-FilterB	2107	4206	1912	57

Fig. 1. Number of clusters (left) and number of genes (right) obtained

size of the groups generated by each time series and mean. It should be noted that the merge method considers very few genes as uncorrelated with other genes (genes in M_0) compared to the rest of clusterings. The lower the α value, the more probable is to get large groups and the more different the time series, the larger the obtained clusters. Generalizing, we can say that the proposed merge method generates a few more clusters compared to the mean clustering results and less than clusterings obtained by SC/SOC for individual time series.

Regarding the gene profiles, meaningful groups are obtained applying the merge method over the considered clustering methods. A cluster of four genes obtained by the merge method is depicted as an example in Fig. 2. As it can be seen, the merge method groups those genes that have similar genes profiles over time for all time series. The same four genes have been observed clustered in FilterB-SOC-Growth-Mean but included in a larger cluster of 131 genes.

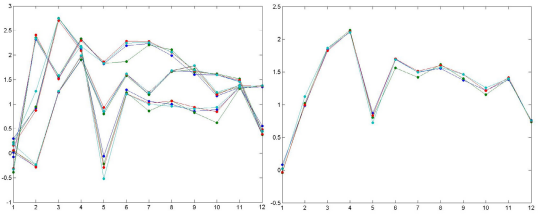


Fig. 2. Genes profiles of a cluster with four genes obtained by the merge method. Profiles for each time series (left) and the mean of the gene values (right) are depicted.

On the other hand, Fig. 3 shows a cluster of 3 genes created by $\alpha = 0.666$ where it can be clearly observed that the correlation between genes is a bit less strict than in previous clusters created by $\alpha = 1$. In this case, for FilterB-SOC-Growth, series2 considers the three genes as uncorrelated while series1 and series3 classify the three genes in the same group of 5 and 4 genes respectively. Unlike the other example, the clustering over the mean of the time series considers that one of the genes is not correlated with any one and the other genes are classified in separate clusters of 2 genes each.

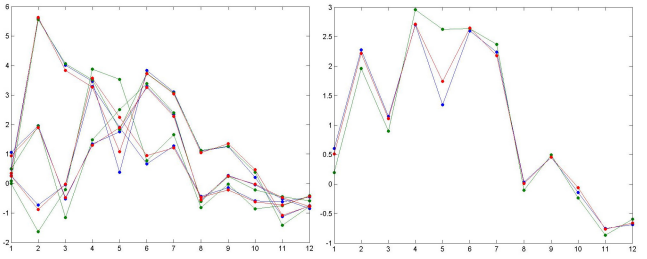


Fig. 3. Genes profiles of a cluster of 3 genes obtained by the merge method. The gene profiles of each time series (left) and the mean of the gene profiles (right) are depicted.

4.2 Biological Perspective

The method proposed has reached the final objective of clustering genes in groups with similar expression profiles. It has been seen that production genes were clustered with others which have similar expression and their expression profiles follow a common pattern which is coherent with the production rate.

Moreover, it has been shown how some genes have different expression values between the three time series. This is not a usual case in these time series because they are well synchronized, but it is interesting to consider that some genes could have high expression variations during time. These genes can be related to tight regulation processes in which the expression changes are quite frequent. In order to deal with these profiles, it has been shown how the series can be analyzed without previous merging. In addition, this approach could be interesting for the analysis of bad synchronized or noisy samples, in order to avoid potential errors in the clustering calculation process.

5 Conclusions and Future Work

A clustering merge method was proposed to combine the information provided by different time series for time-course microarray analysis. The results obtained indicate that the merge method is able to generate valuable clusters containing genes highly correlated with regard to all time series compared to the clustering methods applied over the averaged or individual time series. It has been shown how the mean of the series is unable to find any relationship between genes despite very similar expression profiles with regard to some of the time series.

The proposed method can be useful in further analysis of time series microarrays for detecting important genes that promote a certain event or variable. Because the merge method provides an indicator of the correlation between the genes of the generated clusters, it is possible to analyze groups that have obtained a low indicator of correlation in order to detect those genes with high expression variations over time. This type of genes, characterized by frequent expression changes, are not regular in synchronized time series but they are likely to be fairly common in noisy samples. Therefore, the proposed merge method can be employed for detecting genes related to tight regulation and for analysis of poorly synchronized or noisy samples.

Acknowledgements. This work was supported by a grant of the Agencia de Inversiones y Servicios de Castilla y León (record CCTT/10/BU/0002 and CCTT/10/LE/0001) and the Spanish Ministry of Science and Innovation [PID 560300-2009-11]. Carlos Prieto was supported by a Juan de la Cierva programme (JCI-2009-05444) of the Ministry of Science and Innovation (Spain).

References

[1] Lee, C.-P., Lin, W.-S., Chen, Y.-M., Kuo, B.-J.: Gene selection and sample classification on microarray data based on adaptive genetic algorithm/k-nearest neighbor method. Expert Systems with Applications 38(5), 4661–4667 (2011)

[2] Liu, H., Liu, L., Zhang, H.: Ensemble gene selection by grouping for microarray data classification. Journal of Biomedical Informatics 43(1), 81–87 (2010); PMID: 19699316

[3] Wang, Y., Tetko, I.V., Hall, M.A., Frank, E., Facius, A., Mayer, K.F.X., Mewes, H.W.: Gene selection from microarray data for cancer classification–a machine learning approach. Computational Biology and Chemistry 29(1), 37–46 (2005)

[4] Wei, J.S., Greer, B.T., Westermann, F., Steinberg, S.M., Son, C.-G., Chen, Q.-R., Whiteford, C.C., Bilke, S., Krasnoselsky, A.L., Cenacchi, N., Catchpoole, D., Berthold, F., Schwab, M., Khan, J.: Prediction of clinical outcome using gene expression profiling and artificial neural networks for patients with neuroblastoma. Cancer Research 64(19), 6883–6891 (2004)

[5] Coffey, N., Hinde, J.: Analyzing time-course microarray data using functional data analysis - a review. Statistical Applications in Genetics and Molecular Biology 10 (2011); peer-reviewed

[6] Krishna, R., Li, C.-T., Buchanan-Wollaston, V.: A temporal precedence based clustering method for gene expression microarray data. BMC Bioinformatics 11(1), 68 (2010)

[7] Yi, S.-G., Joo, Y.-J., Park, T.: Rank-based clustering analysis for the time-course microarray data. Journal of Bioinformatics and Computational Biology 7(1), 75–91 (2009); PMID: 19226661

[8] Storey, J., Xiao, W., Leek, J., Tompkins, R., Davis, R.: Significance analysis of time course microarray experiments. UW Biostatistics Working Paper Series (August 2004)

[9] Wolfe, C.J., Kohane, I.S., Butte, A.J.: Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. BMC Bioinformatics 6, 227 (2005); PMID: 16162296

[10] Phan, S., Famili, F., Tang, Z., Pan, Y., Liu, Z., Ouyang, J., Lenferink, A., McCourt O'connor, M.: A novel pattern based clustering methodology for time-series microarray data. International Journal of Computer Mathematics 84(5), 585–597 (2007)

[11] Smyth, G.K., Speed, T.: Normalization of cdna microarray data. Methods 31(4), 265–273 (2003)

Prediction of Dental Milling Time-Error by Flexible Neural Trees and Fuzzy Rules

Pavel Krömer[1,2], Tomáš Novosád[1], Václav Snášel[1,2], Vicente Vera[4], Beatriz Hernando[4], Laura García-Hernandez[7], Héctor Quintián[3], Emilio Corchado[2,3], Raquel Redondo[5], Javier Sedano[6], and Alvaro E. García[4]

[1] Dept. of Computer Science, VŠB-Technical University of Ostrava, Czech Republic
{pavel.kromer,tomas.novosad,vaclav.snasel}@vsb.cz
[2] IT4Innovations, Ostrava, Czech Republic
[3] Departamento de Informática y Automática, Universidad de Salamanca, Spain
escorchado@usal.es
[4] Facultad de Odontología, UCM, Madrid, Spain
{vicentevera,aegarcia}@odon.ucm.es
[5] Department of Civil Engineering, University of Burgos, Burgos, Spain
rredondo@ubu.es
[6] Dept. of AI & Applied Electronics, Castilla y León Technological Institute, Burgos, Spain
javier.sedano@itcl.es
[7] Area of Project Engineering, University of Cordoba, Spain
ir1gahel@uco.es

Abstract. This multidisciplinary study presents the application of two soft computing methods utilizing the artificial evolution of symbolic structures – evolutionary fuzzy rules and flexible neural trees – for the prediction of dental milling time-error, i.e. the error between real dental milling time and forecast given by the dental milling machine. In this study a real data set obtained by a dynamic machining center with five axes simultaneously is analyzed to empirically test the novel system in order to optimize the time error.

Keywords: dental milling, prediction, evolutionary algorithms, flexible neural trees, fuzzy rules, industrial applications.

1 Introduction

Accurate scheduling and planning becomes increasingly important part of modern industrial processes. To optimize the manufacturing of products and schedule the utilization of devices, the product manufacturing time has to be known in advance. However, the predictions given by traditional methods and tools are often less accurate. Precise prediction of product manufacturing time is important for industrial production planning in order to meet, industrial, technological, and economical objectives [2,14]. The goal of a production process is to deliver products on time and utilize the resources at maximum during production cycles. Production time estimates provided either by production models (i.e. by

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 842–849, 2012.

auxiliary software) or human experts are often less accurate than desirable [2]. Soft computing techniques can be used for flexible and detailed modelling of production processes [5]. The area of soft computing represents a set of various technologies involving non-linear dynamics, computational intelligence, ideas drawn from physics, physicology and several other computational frameworks. It investigates, simulates and analyzes very complex issues and phenomena in order to solve real-world problems: such as the failures detection in dental milling process, which requires a multidisciplinary approach [12].

In this study, a real data set obtained by a dynamic machining center with five axes simultaneously is analyzed by means of two soft computing techniques to empirically test the system in order to optimize the time error.

2 Flexible Neural Tree

Flexible neural tree (FNT) [3] is a hierarchical neural network, which is automatically created in order to solve given problem. Its structure is usually determined using some adaptive mechanism and it is intended to adapt to the problem and data under investigation [11,10,4]. Due to this property of the FNTs, it is not necessary to setup some generic static network structure not related to the problem domain beforehand.

A general and enhanced FNT model can be used for problem solving. Based on the predefined instruction/operator sets, a FNT model can be created and evolved. In this approach, over-layer connections, different activation functions for different nodes and input variables selection are allowed. The hierarchical structure could be evolved by using genetic programming. The fine tuning of the parameters encoded in the structure could be accomplished by using parameter optimization algorithms. The FNT evolution used in this study combines both approaches. Starting with random structures and corresponding parameters, it first tries to improve the structure and then as soon as an improved structure is found, it fine tunes its parameters. It then goes back to improving the structure again and, provided it finds a better structure, it again fine tunes the rules' parameters. This loop continues until a satisfactory solution is found or a time limit is reached. A tree-structural based encoding method with specific instruction set is selected for representing a FNT model in this research. The reason for choosing the representation is that the tree can be created and evolved using the existing or modified tree-structure-based approaches.

The fitness function maps the FNT to a scalar, real-valued fitness values that reflect the FNT's performances on a given task. Firstly the fitness functions should be seen as error measures, i.e. mean square error (MSE) or root mean square error ($RMSE$). A secondary non-user-defined objective for which algorithm always optimizes FNTs is FNT size as measured by number of nodes. Among FNTs with equal fitness smaller ones are always preferred. MSE and $RMSE$ are given by:

$$MSE(i) = \frac{1}{P}\sum_{j=1}^{P}(y_1^j - y_2^j)^2, \qquad RMSE(i) = \sqrt{MSE(i)} \tag{1}$$

where P is the total number of samples, y_1^j and y_2^j are the actual time-series and the FNT model output of j-th sample. $MSE(i)$ and $RMSE(i)$ denotes the fitness value of i-th individual.

2.1 Tree Structure and Parameter Learning

Finding an optimal or near-optimal flexible neural tree can be accomplished by various evolutionary and bio-inspired algorithms [11,10,4]. The general learning procedure for constructing the FNT model can be described in high level as follows [3]:

1. Set the initial values of parameters used in the GA algorithms. Set the elitist program as NULL and its fitness value as a biggest positive real number of the computer at hand. Create a random initial population (flexible neural trees and their corresponding parameters)
2. Structure optimization by genetic algorithm, in which the fitness function is calculated by MSE or RMSE
3. If a better structure is found and no better structure is found for certain number of generations, then go to step (4), otherwise go to step (2)
4. Parameter optimization by genetic algorithms. In this stage, the tree structure or architecture of flexible neural tree model is fixed, and it is the best tree taken from the sorted population of trees. All of the parameters used in the best tree formulated a parameter vector to be optimized by local search
5. If the maximum number of local search is reached, or no better parameter vector is found for a significantly long time then go to step (6); otherwise go to step (4);
6. If satisfactory solution is found, then the algorithm is stopped; otherwise go to step (2).

Evolutionary methods [1] are in this study used for FNT structure optimization as well as for activation function parameters and tree nodes weights optimization. The selection, crossover and mutation operators used are same as those of standard genetic programming [1]. A genetic algorithm starts with selection of two parents from current population. The product of crossover operator can be one or more offspring - two in this study. The mutation of offspring is performed at the last step of genetic algorithm. After these three steps we have new offspring which is placed into a newly created population. The process is repeated until desired new population is built. As soon as the new population is built, the new population is evaluated and sorted according to the fitness function. Selection is in the FNT evolution implemented using the weighted roulette wheel

algorithm and the tree structure crossover is implemented as an exchange of randomly selected subtrees of parent chromosomes. The crossover of node weights and activation function parameters is done in a similar way as in previous studies applying genetic algorithms to neural network training [6]. A variety of FNT mutation types were used:

1. Changing one terminal node: randomly select one terminal node in the neural tree and replace it with another terminal node.
2. Changing one function node: randomly select one function node and replace it with a newly generated subtree.
3. Growing: select a random function node in hidden layer of the neural tree and add newly generated subtree as a new child.
4. Pruning: randomly select a node in the neural tree and delete it in the case the parent node has more than two child nodes.

The mutation of tree weights and activation function parameters is the same as in the genetic algorithms for artificial neural networks [6].

3 Fuzzy Rules Evolved by Genetic Programming

Fuzzy rules (FR) [7,8,13] inspired by the area of fuzzy information retrieval (IR) [9] and evolved by genetic programming have been shown to achieve interesting results in the area of data mining and pattern analysis.

The fuzzy rules use similar data structures, basic concepts, and operations as the fuzzy information retrieval but they can be used for the analysis (i.e. classification, prediction) of general data. A fuzzy rule has the form of a weighted symbolic expression roughly corresponding to an extended Boolean query in the fuzzy IR analogy. The rule consists of weighted feature (attribute) names and weighted aggregation operators. The evaluation of such an expression assigns a real value from the range $[0, 1]$ to each data record. Such a valuation can be interpreted as an ordering or a fuzzy set over the data records. The fuzzy rule is a symbolic expression that can be parsed into a tree structure. The tree structure consists of nodes and leaves (i.e. terminal nodes). An example of fuzzy rule is give below:

feature1:0.5 and:0.4 (feature2[1]:0.3 or:0.1 ([1]:0.1 and:0.2 [2]:0.3))

In the fuzzy rule syntax can be seen three types of nodes: the feature node is defined by feature name and its weight (*feature1:0.5*) and represents a requirement on current value of a feature, past feature node is defined by feature name, index of previous record, and weight (*feature2[1]:0.3*) and it is requirement on previous value of a feature. Finally, the past output node is defined by the index of previous output and weight (*[1]:0.5*) and represents a requirement on previous value of the predicted output variable. Clearly, such a fuzzy rule can be used for the analysis of both, data sets consisting of independent records and time series.

The fuzzy rules are evaluated using the formulas and equations from the area of fuzzy IR and fuzzy sets (see e.g. [7,8,13]). The terminal node weights are

interpreted as threshold for data feature values and operator nodes are mapped to fuzzy set operators. The fuzzy rule predicting certain value for a given data set is found using standard genetic programming that evolves a population of tree representations of the rules in a supervised manner. The whole procedure is very similar to the evolution of the FNT structure described in section 2.1 but it differs in the choice of the fitness function which is taken from the area of fuzzy IR. The correctness of search results in IR can be evaluated using the measures precision P and recall R. Precision corresponds to the probability of retrieved document to be relevant and recall can be seen as the probability of retrieving a relevant document. Precision and recall in the extended Boolean IR model can be defined using the $\Sigma-$count $\|A\|$ [15]:

$$\rho(X|Y) = \begin{cases} \frac{\|X \cap Y\|}{\|Y\|} & \|Y\| \neq 0 \\ 1 & \|Y\| = 0 \end{cases}, \; P = \rho(REL|RET), \; R = \rho(RET|REL) \quad (2)$$

where REL stands for the fuzzy set of all relevant documents, RET for the fuzzy set of all retrieved documents, and $\|A\|$ is the $\Sigma-$count, i.e. the sum of the values of characteristic function μ_A for all members of the fuzzy set $\|A\| = \sum_{x \in A} \mu_A(x)$ [15]. The F-score F is among the most used scalar combinations of P and R:

$$F = \frac{(1 + \beta^2)PR}{\beta^2 P + R} \quad (3)$$

For the evolution of fuzzy rules [7,8,13] we map the prediction given for training data set by the fuzzy rule to RET and the desired values to REL. F corresponds to the similarity of two fuzzy sets and a fuzzy rule with high F provides good approximation of the output value.

4 Industrial Case Study and Experiments

FNTs and FRs were evolved for the estimation of time-error in a real dental milling process. The data was gathered by means of a Machining Milling Center of HERMLE type-C 20 U (iTNC 530), with swivelling rotary (280 mm), with a control system using high precision drills and bits (see fig. 1) and it was used for the evolution of FNTs and FRs by the genetic programming. The models were trained using an initial data set of 98 samples obtained by the dental scanner in the manufacturing of dental pieces with different tool types (plane, toric, spherical and drill). The data set contained records consisting of 8 input variables (Tool, Radius, Revolutions, Feed rate X, Y and Z, Thickness, Initial Temperature) and 1 output variable (Time Error for manufacturing) as shown in table 1. Time error for manufacturing is the difference between the time estimated by the machine itself and real production time. Negative values indicate that real time exceeds estimated time. The goal of this study was to evaluate the ability of evolutionary evolved FNT and FR to predict the dental milling time-error from the data. The parameters used for the evolution of the FNT and FR are

(a) Machining/ Milling Center of HERMLE type-C 20 U (iTNC 530)

(b) Manufacturing of metal pieces

Fig. 1. Machining/ Milling Center of HERMLE type-C 20 U (iTNC 530)

Table 1. Description of variables in the data set

Variable (Units)	Range of values
Type of tool	Plane, toric, spherical and drill
Radius (mm.)	0.25 to 1.5
Revolutions per minute (RPM)	7,500 to 38,000
Feed rate X (mm. by minute)	0 to 3,000
Feed rate Y (mm. by minute)	0 to 3,000
Feed rate Z (mm. by minute)	50 to 2,000
Thickness (mm.)	10 to 18
Temperature (°C)	24.1 to 31
Real time of work (s)	6 to 1,794
Time errors for manufacturing (s)	-28 to -255

Table 2. FNT and FR evolution parameters

Method	Parameters
FNT	pop. size 100, crossover probability P_C 0.8, mutation probability P_M 0.2, limiting number of 10 generations, fitness function $RMSE$, Gaussian activation function with a, b, and weights from the range $[0, 1]$
FR	pop. size 100, crossover probability P_C 0.8, mutation probability P_M 0.2, limiting number of 1000 generations, no past feature nodes and no past output nodes allowed, fitness function F-Score with $\beta = 1$

Table 3. Dental milling time-error prediction accuracy

Method	Prediction error (sec)	Prediction accuracy (pct)
FNT	4.52	98.00
FR	5.55	97.55

shown in table 2. They were selected on the basis of initial experiments and past experience with the methods. Because the data set was rather small, all of it was used to train the predictors. The average prediction error ($RMSE$) for both methods on the full dental milling data set is shown in table 3. The table shows that both methods were able to learn the trends in the data set with a similar accuracy of 97.55 % and 98.0%. Visual illustration of the time-error prediction by FNT and FR is shown in fig. 2 and fig. 3 respectively.

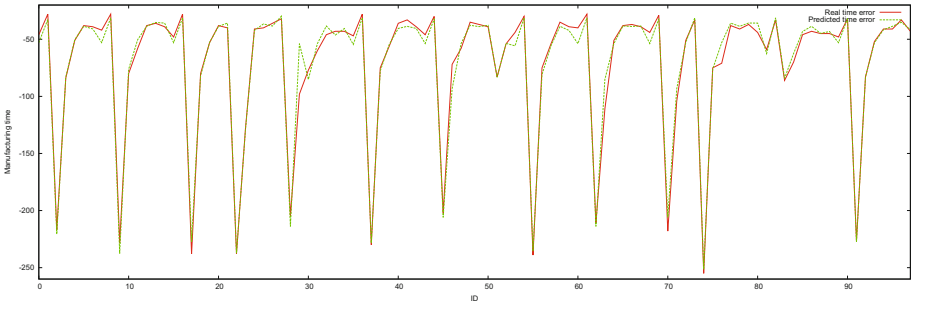


Fig. 2. Visual results of the prediction by FNT

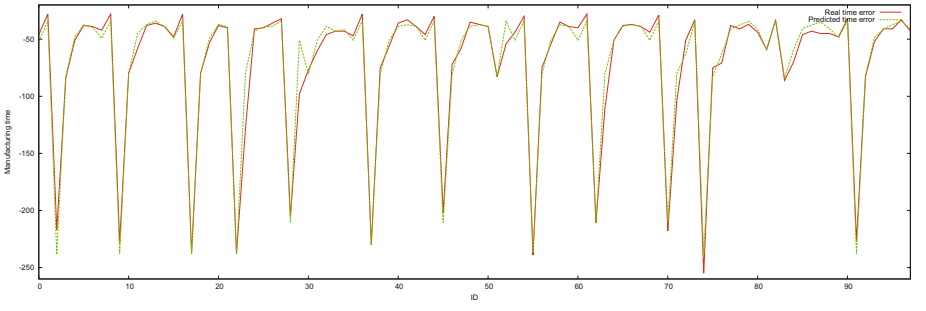


Fig. 3. Visual results of the prediction by FR

5 Conclusions

This study presents initial results on the prediction of dental milling time-error by flexible neural trees and fuzzy rules evolved by artificial evolution. Both soft computing models were trained on a real-world data set describing the production of a dental milling machine and their ability to adapt to the data was compared. Both methods have shown very similar ability to predict the dental milling time-error, the former reaching 0.45 % higher prediction accuracy.

In the future, the prediction of dental milling time-error by both methods will be evaluated in a statistically sounder manner (e.g. by n-fold cross validation) to obtain better comparison of both methods. Moreover, precise production models can be used for further optimization of the industrial processes and device parameters.

Acknowledgements. This research is partially supported through project of the Spanish Ministry of Science and Innovation [ref:TIN2010-21272-C02-01]. The authors would also like to thank ESTUDIO PREVIO (Madrid-Spain) and TARAMI (Madrid-Spain) for their collaboration in this research. This work was also supported in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/ 02.0070 by operational programme "Research and Development for Innovations" funded by the Structural Funds of the European Union and

state budget of the Czech Republic, by the Ministry of Industry and Trade of the Czech Republic, under the grant no. FR-TI1/420, and by SGS, VSB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/58.

References

1. Affenzeller, M., Winkler, S., Wagner, S., Beham, A.: Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications. Chapman & Hall/CRC (2009)
2. Chang, P., Liao, T.: Combining som and fuzzy rule base for flow time prediction in semiconductor manufacturing factory. Applied Soft Computing 6(2), 198–206 (2006)
3. Chen, Y., Abraham, A.: Flexible Neural Tree: Foundations and Applications. In: Chen, Y., Abraham, A. (eds.) Tree-Structure Based Hybrid Computational Intelligence. ISRL, vol. 2, pp. 39–96. Springer, Heidelberg (2010)
4. Chen, Y., Yang, B., Meng, Q.: Small-time scale network traffic prediction based on flexible neural tree. Appl. Soft Comput. 12(1), 274–279 (2012)
5. Custodio, L.M.M., Sentieiro, J.J.S., Bispo, C.F.G.: Production planning and scheduling using a fuzzy decision system. IEEE Trans. on Robotics and Automation 10(2), 160–168 (1994)
6. Ding, S., Li, H., Su, C., Yu, J., Jin, F.: Evolutionary artificial neural networks: a review. Artificial Intelligence Review, 1–10 (2011)
7. Krömer, P., Platoš, J., Snášel, V., Abraham, A.: Fuzzy classification by evolutionary algorithms. In: IEEE Int. Conf. on Systems, Man, and Cybernetics, pp. 313–318. IEEE SMC Society (2011)
8. Krömer, P., Platoš, J., Snášel, V., Abraham, A., Prokop, L., Mišák, S.: Genetically evolved fuzzy predictor for photovoltaic power output estimation. In: INCoS 2011, pp. 41–46. IEEE (2011)
9. Pasi, G.: Fuzzy Sets in Information Retrieval: State of the Art and Research Trends. In: Bustince, H., Herrera, F., Montero, J. (eds.) Fuzzy Sets and Their Extensions: Representation, Aggregation and Models. STUDFUZZ, vol. 220, pp. 517–535. Springer, Heidelberg (2008)
10. Peng, L., Yang, B., Zhang, L., Chen, Y.: A parallel evolving algorithm for flexible neural tree. Parallel Computing 37(1011), 653–666 (2011)
11. Qi, F., Liu, X., Ma, Y.: Synthesis of neural tree models by improved breeder genetic programming. Neural Computing & Applications 21, 515–521 (2012)
12. Sedano, J., Curiel, L., Corchado, E., de la Cal, E., Villar, J.R.: A soft computing method for detecting lifetime building thermal insulation failures. Integr. Comput.-Aided Eng. 17(2), 103–115 (2010)
13. Snášel, V., Krömer, P., Platoš, J., Abraham, A.: The Evolution of Fuzzy Classifier for Data Mining with Applications. In: Deb, K., Bhattacharya, A., Chakraborti, N., Chakroborty, P., Das, S., Dutta, J., Gupta, S.K., Jain, A., Aggarwal, V., Branke, J., Louis, S.J., Tan, K.C. (eds.) SEAL 2010. LNCS, vol. 6457, pp. 349–358. Springer, Heidelberg (2010)
14. Vera, V., Corchado, E., Redondo, R., Sedano, J., Garcia, A.: Applying soft computing techniques to optimize a dental milling process. Neurocomputing (submitted)
15. Zadeh, L.A.: Test-score semantics dor natural languages and meaning representation via Pruf. In: Empirical Semantics, Quantitative Semantics, vol. 12, 1, pp. 281–349. Studienverlag Brockmeyer, Bochum (1981)

Cost-Sensitive Splitting and Selection Method for Medical Decision Support System

Konrad Jackowski, Bartosz Krawczyk, and Michał Woźniak

Department of Systems and Computer Networks,
Wroclaw University of Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{konrad.jackowski,bartosz.krawczyk,michal.wozniak}@pwr.wroc.pl

Abstract. The paper presents a cost-sensitive modification of the Adaptive Splitting and Selection (AdaSS) algorithm, which trains a combined classifier based on a feature space partitioning. In this study the algorithm considers constraints put on the cost of selected features, which are one of the key-problems in the clinical decision support systems. The modified version takes into consideration both the overall classification accuracy and the cost constraints, returning balanced solution for the problem at hand. Proposed method was evaluated on the basis of computer experiments run on cost-sensitive medical benchmark datasets.

Keywords: machine learning,multiple classifier system, cost-sensitive classification, clustering and selection, evolutionary algorithm, feature selection.

1 Introduction

The classification systems are widely used to solve practical problems coming from different areas of human activities. Nowadays for most of them we can choose an appropriate classification method to make a high-quality decision. However, its choice is usually limited by several constrains related to the classification cost. We can introduce this problem into training and exploitation of classifiers. First, and probably the most known, criterion is the misclassification cost which is widely discussed in the literature e.g., [2] and is the key feature of the Bayes decision theory. It proposes to define so-called loss function which informs about the misclassification costs between each pair of classes. According to this theory the optimal classifier makes decisions which minimize the expectation value of the mentioned above loss function. We can find several proposition on how to deal with it e.g., Peng et al. proposed how to create cost-sensitive ensemble for the medical decision support system [13]. Of course the misclassification cost is important but let us note that we can find another sources that may generate additional costs during training or exploitation of classifiers:

– cost of training which is usually high for classifiers building a structure used during classification e.g., decision trees [1];

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 850–857, 2012.

– cost of testing;
– cost of decision making which usually takes into consideration the cost of necessary feature acquisition.

The scope of this work focuses on the last type of presented cost sources. Here cost may be measured in a given currency as a price of medical tests or time, in case of measurements, which repetitions require significant time. Such a meaning of cost-sensitive classification arises frequently in many fields of human activities as an industrial production process [16], robotics [14], technological diagnosis [8] to enumerate only a few. But this problem is most clearly visible for the medical diagnosis [11]. As we mentioned above for the most of decision problems we have a necessary tools at our disposal to make a high quality decision, but from practical point of view we have to notice that situation that a physician has unlimited budget for diagnosis can be found in TV serials, as "House M.D.", only. Therefore in real cases physicians have to balance the costs of various tests with the expected benefits. Often doctors have to make the diagnosis fast on the basis of (low cost) features acquired form measurements that do not require much time to conduct - because therapeutic action has to be taken without any delay.

In this work we are focusing on the problem of how to design combined classifier which takes into consideration the cost of exploitation.

This work is the continuation of the previous works of authors on combined classifiers [4,6,17] and cost-sensitive classifiers where they considered decision tree training with limited budget of exploitation [12].

2 Classification Algorithm

Let us present shortly the model of the compound classifier.

We have a pool of n individual classifiers $\Psi^{(1)}, ..., \Psi^{(n)}$ which assign object described by its features $x = [x^{(1)}, ..., x^{(d)}]^T \in X$ to the class drawn from a predefined set of classes $\mathcal{M} = \{1, ..., M\}$. The l-th classifier makes the decision independently according to:

$$\Psi^{(l)}(x) = i \Leftrightarrow F^{(l)}(i, x) = \max_{k \in \mathcal{M}} F^{(l)}(k, x), \tag{1}$$

where $F^{(l)}(k, x)$ denotes the discriminant function which supports classification of the object x to the class k.

The pool is homogeneous i.e., all individual classifiers use the same classification and training algorithms. Therefore, to ensure diversity of the pool each classifier uses an unique subset of attributes drawn from X [7].

To effectively exploit the local competencies of individual classifiers, the feature space is partitioned into a set of disjoint subspaces (constituents) called competence areas:

$$X = \bigcup_{h=1}^{H} \hat{\mathcal{X}}_h, \quad \forall k, l \in \{1, ..., H\}, \quad k \neq l, \quad \hat{\mathcal{X}}_k \cap \hat{\mathcal{X}}_l = \varnothing. \tag{2}$$

Each area is represented by an associated centroid [5]:

$$\mathcal{C}_h = [c_h^{(1)}, c_h^{(2)}, ..., c_h^{(d)}]^T \in \mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_H\}. \tag{3}$$

$$x \in \hat{\mathcal{X}}_h \Leftrightarrow d(x, C_h) = \min_{k=1}^{H} d(x, C_k), \tag{4}$$

where $d(a, b)$ is an Euclidean distance metric.

Additionally let's define useful function *member* which returns the centroid label for a given x

$$member(\mathcal{C}, x) = h \Leftrightarrow x \in \hat{\mathcal{X}}_h, \tag{5}$$

Each competence area has assigned its own area classifier $\Psi_h(x)$ which is devoted to classifying all objects which belong to $\hat{X}_h$. It makes the collective decision based on the weighted fusion of the discriminant functions returned by elementary classifiers drawn from the pool. Let $\prod_h$ denotes the set of indices of classifiers in the pool which join the committee of $\Psi_h(x)$ area classifier:

$$\Psi_h(x) = i \Leftrightarrow F_h^{(l)}(i, x) = \max_{k \in \mathcal{M}} F_h^{(l)}(k, x), \tag{6}$$

where $F_h^{(l)}$ is the common discriminant function of the h-th area classifier given by:

$$F_h^{(l)}(k, x) = \sum_{l \in \prod_h} w_h^{(l)}(k) F^{(l)}(k, x), \tag{7}$$

where $w_h^{(l)}(k)$ is the weight assigned to l-th elementary classifier and class k in h-th area classifier. The details of the used fuser can be found in [17,18].

Aforementioned assumptions are satisfied by a compound classifier making decision according to the following formulae:

$$\Psi(x) = i \Leftrightarrow \Psi_{member(c,x)}(x) = i. \tag{8}$$

3 Training Algorithm

3.1 Training Criterions

Two main objectives are defined for the training algorithm:

1. maximization of the accuracy of the classification,
2. minimization of the cost of data acquisition required for classification.

The first one is based on the commonly known criterion of frequency of correct classifications:

$$Q(\Psi) = \frac{1}{K}\sum_{n=1}^{K}(\delta(\Psi_{member(C,x_n)}(x_n), j_n)), \tag{9}$$

where δ denotes Kronecker's delta. This is computed over a learning set:

$$\mathcal{LS} = \{(x_1, j_1), (x_2, j_2), ..., (x_K, j_K)\}, \tag{10}$$

where x_i denotes observations described in the i-th object and j_i denotes its correct class label [1].

Because we would like to assure the pool diversity by the individual classifier training on the basis of different subsets of available features, therefore let us propose the following representation of the classifier $\Psi_h(x)$:

$$\mathcal{A}_h = \begin{pmatrix} a_h^{(1)}(x^{(1)}) & \cdots & a_h^{(1)}(x^{(d)}) \\ \vdots & \ddots & \vdots \\ a_h^{(n)}(x^{(1)}) & \cdots & a_h^{(n)}(x^{(d)}) \end{pmatrix}, \tag{11}$$

where $a_h^{(p)}(x^{(q)}) = 1$ if the q-th feature is used by the p-th individual classifier used by $\Psi_h(x)$, otherwise $a_h^{(p)}(x^{(q)}) = 0$ means that mentioned above attribute is not used by it. It allows us to formulate classification cost of the compound classifier (8):

$$cost(\Psi) = \sum_{l=1}^{d}\left(sgn\left(\sum_{h=1}^{H}\sum_{k=1}^{n} a_h^{(k)}(x^{(l)})\right) cost(x^{(l)})\right), \tag{12}$$

where $cost(x^{(l)})$ denotes the acquisition cost of l-th feature. Both objectives are included in the proposed criterion, which must be subject to a maximization procedure:

$$\hat{Q}(\Psi) = \frac{Q(\Psi)}{(1 + cost(\Psi))^{\omega}}, \tag{13}$$

where ω is a parameter which control the weight of the cost criterion. It may take values in range [0,1]. The mentioned above proposition is similar to the split criterion used by Núñez in the cost-sensitive decision tree induction algorithm EG2 [10].

3.2 Optimization Algorithm

The process of searching for maximum value of criterion (13) was treated as a compound optimization problem solved by an evolutionary algorithm (EA) [9]. It

[1] In our research we assume using so-called 0-1 loss function [2] but let us note that instead of misclassification error estimator we can use the cost of misclassification what requires a different lost function definition.

relays on an iterative processing of a population of individuals which represents possible solutions in the form of a chromosome consisting of three constituents:

$$Chromosome = [\mathcal{A}, \mathcal{C}, \mathcal{W}], \tag{14}$$

where

$$\mathcal{A} = [\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_H]^T, \tag{15}$$

The second constituent $\mathcal{C} = [C_1, C_2, ..., C_H]^T$ represents centroids according to the Eq 3.

Third constituent $\mathcal{W} = [\mathcal{W}_1, \mathcal{W}_2, ..., \mathcal{W}_H]^T$ consists of weights assigned to each individual classifier and each class i.e.:

$$\mathcal{W}_h = \begin{pmatrix} w_h^{(1)}(1) & \cdots & w_h^{(1)}(M) \\ \vdots & \ddots & \vdots \\ w_h^{(n)}(1) & \cdots & w_h^{(n)}(M) \end{pmatrix}. \tag{16}$$

For the purpose of EA implementation $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{W}$ matrices are transformed into vectors. Nonetheless, having in mind that the three constituents of the chromosome have entirely different meaning and nature, we ensure that no data exchange can be done between them while processing the population.

Training procedure consists of following steps:

- Initialization - procedure involving random generation of the initial population and setting of parameters which controls the algorithm: N_c - upper limit of algorithm cycles, N_p - size of population, β - mutation probability, γ - crossover probability, Δ_m - mutation range factor and V - upper limit of algorithm iterations with falling quality ($V < N_c$).
- Selection and reproduction process of drawing the best individuals from the population according to the roulette selection with elitism for further operations.
- Mutation genetic operator which alter chromosomes of selected individuals by adding some random noise, generated according to Gaussian Normal distribution with mean value equal to 0 and standard deviation set to Δ_m. Each constituents of the chromosome is treated separately and can be affected with the following probabilities:

$$P_a(t) = 2\beta\frac{t}{N_c}, \quad P_c(t) = \beta\frac{t}{N_c}, \quad P_w(t) = \beta - P_c(t), \tag{17}$$

  where t is the iteration index of the algorithm, $P_a(t)$ is the mutation probability of the feature vector, $P_c(t)$ is the mutation probability of the centroid vector and $P_w(t)$ is the mutation probability of the weight vector.
- Crossover procedure for exchanging data between the two parent individuals according to the two-point rule.
- Protecting against overfitting procedure which cancels training process when the accuracy of classification (controlled at each generation over a validation data set) deteriorates.

4 Experimental Investigations

The main objective of the experiment was to examine the behavior of the proposed cost-sensitive modifications of the AdaSS algorithm and to investigate the procedure of the cost-driven feature selection.

4.1 Set-Up

In this study medical benchmark datasets from [3] with features described by cost values are used. Details of the datasets used in the experiments are given in the Table 1:

Table 1. Details of datasets used in the experimental investigation

No.	Name	Objects	Features	Classes
1	Heart disease	303	13	2
2	Hepatitis	155	19	2
3	Liver disorders	345	5	2
4	Pima Indians diabetes	768	8	2

All experiments were carried out in the R environment [15]. A pool of individual classifiers consisted of three Support Vector Machines with a polynomial kernel and slack variables, trained with the Sequential Minimal Optimization procedure. Their diversity was ensured by the feature selection process, embedded in the optimization procedure. Additionally to avoid a situation when optimization algorithm returns a homogeneous ensemble a setting in which more than one classifier is trained on identical feature subspace was banned.

For the training phase following parameters have been set: $H = 5$, $N_c = 200$, $N_p = 100$, $\beta = \{0.7;0.3\}$, $\gamma = \{\ 0.3;0.7\}$, $\Delta_m = 0.2$ and $V = 15$.

The ω parameter, responsible for weight of cost criterion, was tested with values $\omega = \{0.0;\ 0.3;\ 0.7;\ 1.0\}$.

4.2 Results and Discussion

The results of experiments are presented in the Figure 1. The plots show the correlation between the maximum cost threshold and overall ensemble error, with respect to chosen cost weight parameter ω. Experiments emphasized the effectiveness of the proposed cost-sensitive modification of the AdaSS algorithm. When comparing the received results with the ones presented in [12] one can see the significant improvement in overall accuracy with the respect to the cost criterion. In all four cases the best results were returned for ω equal to 0.3 or 0.7. But at the same time it is worth noticing that for the most cost thresholds the differences between the ω values were not so significant.

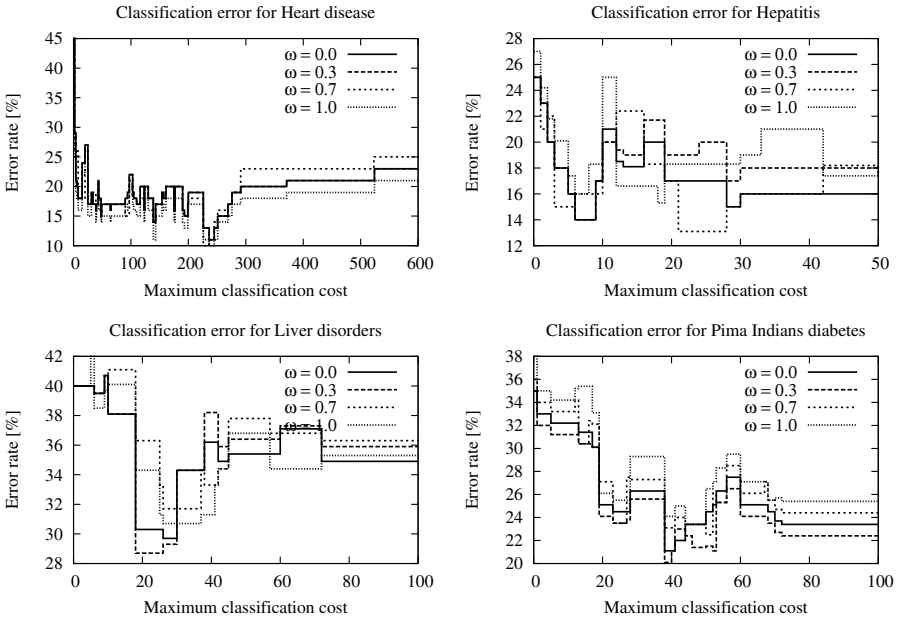


Fig. 1. Classification errors with the respect to the maximum classification cost for the tested datasets

5 Conclusions

The paper dealt with cost-sensitive combined classification algorithm based on a feature space partitioning. The cost in this paper was associated with features selected for each of classifiers in the pool, but the misclassification cost can be easy implemented in any form that was mentioned in the introduction of this paper. The cost-sensitive criterion was embedded in the training procedure via additional constituent in the optimization algorithm responsible for an evolutionary feature selection. The training procedure took into consideration both the overall classification accuracy and the cost constraints, returning a balanced solution for the problem at hand.

We believe that the proposed method may be very useful in real-life clinical decision support systems, in which physicians require at the same time a high accuracy and a lowest possible diagnosis cost. Our future studies will concentrate on increasing the diversity of the ensemble by adding a classifier model selection to the optimization procedure and expanding our approach with a classifier selection scheme for each of the clusters.

Acknowledgement. This work is supported by The Polish National Science Centre under the grant which is being realized in years 2010-2013.

References

1. Alpaydin, E.: Introduction to Machine Learning, 2nd edn. The MIT Press (2010)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2001)
3. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
4. Jackowski, K., Wozniak, M.: Algorithm of designing compound recognition system on the basis of combining classifiers with simultaneous splitting feature space into competence areas. Pattern Analysis and Applications 12(4), 415–425 (2009)
5. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. 31, 264–323 (1999)
6. Krawczyk, B., Woźniak, M.: Designing Cost-Sensitive Ensemble – Genetic Approach. In: Choraś, R.S. (ed.) Image Processing and Communications Challenges 3. AISC, vol. 102, pp. 227–234. Springer, Heidelberg (2011)
7. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
8. Lirov, Y., Yue, O.-C.: Automated network troubleshooting knowledge acquisition. Applied Intelligence 1, 121–132 (1991)
9. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs, 3rd edn. Springer, London (1996)
10. Núñez, M.: The use of background knowledge in decision tree induction. Mach. Learn. 6(3), 231–250 (1991)
11. Núñez, M.: Economic induction: A case study. In: EWSL, pp. 139–145 (1988)
12. Penar, W., Wozniak, M.: Cost-sensitive methods of constructing hierarchical classifiers. Expert Systems 27(3), 146–155 (2010)
13. Peng, Y.H., Huang, Q., Jiang, P., Jiang, J.: Cost-Sensitive Ensemble of Support Vector Machines for Effective Detection of Microcalcification in Breast Cancer Diagnosis. In: Wang, L., Jin, Y. (eds.) FSKD 2005, Part II. LNCS (LNAI), vol. 3614, pp. 483–493. Springer, Heidelberg (2005)
14. Tan, M., Schlimmer, J.C.: Cost-sensitive concept learning of sensor use in approach and recognition. In: Proceedings of the Sixth International Workshop on Machine Learning, pp. 392–395. Morgan Kaufmann Publishers Inc., San Francisco (1989)
15. R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008)
16. Verdenius, F.: A method for inductive cost optimization. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 179–191. Springer, Heidelberg (1991)
17. Wozniak, M., Krawczyk, B.: Combined classifier based on feature space partitioning. Journal of Applied Mathematics and Computer Science 22(4) (in press, to appear, 2012)
18. Wozniak, M., Zmyslony, M.: Combining classifiers using trained fuser - analytical and experimental results. Neural Network World 13(7), 925–934 (2010)

Author Index